

STEPPING INTO THE CLOUDS : ENABLING  
COMPANIES TO ADAPT THEIR CAPABILITIES TO  
CLOUD COMPUTING TO SUCCEED UNDER  
UNCERTAIN CONDITIONS

Marc Werfs

A Thesis Submitted for the Degree of PhD  
at the  
University of St Andrews



2016

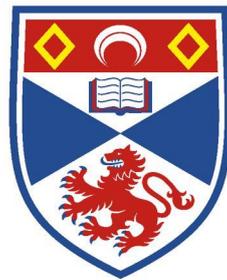
Full metadata for this item is available in  
St Andrews Research Repository  
at:  
<http://research-repository.st-andrews.ac.uk/>

Please use this identifier to cite or link to this item:  
<http://hdl.handle.net/10023/15651>

This item is protected by original copyright

Stepping into the clouds – enabling companies to  
adapt their capabilities to cloud computing to succeed  
under uncertain conditions

Marc Werfs



University of  
St Andrews

This thesis is submitted in partial fulfilment for the degree of PhD  
at the  
University of St Andrews

September 2015

## **Candidate's declarations**

I, Marc Werfs, hereby certify that this thesis, which is approximately 67000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in November, 2012 and as a candidate for the degree of PhD in November, 2013; the higher study for which this is a record was carried out in the University of St Andrews between 2012 and 2015.

Date..... signature of candidate .....

## **Supervisor's declaration**

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date..... signature of supervisor .....

## Permission for publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

### PRINTED COPY

- b) Embargo on all or part of print copy for a period of 2 years on the following ground:
- Publication would preclude future publication

### ELECTRONIC COPY

- b) Embargo on all or part of electronic copy for a period of 2 years on the following ground:
- Publication would preclude future publication

Date..... signature of candidate .....signature of supervisor .....

## List of publications

This thesis includes material from the following publications

1. Werfs, M. & Baxter, G., 2013. Towards resilient adaptive socio-technical systems. *Proceedings of the 31st European Conference on Cognitive Ergonomics - ECCE '13, Toulouse, France*.
2. Werfs, M. et al., 2013. Migrating Software Products to the Cloud: An Adaptive STS Perspective. *Journal of International Technology & Information Management*, 22, pp. 37–54.
3. Werfs, M., Baxter, G., 2014. A longitudinal study of the issues involved in migrating software products and services to the cloud. Services to the Cloud Project Technical Report.
4. Werfs, M. (2015). cFRAM - Using the Functional Resonance Analysis Method to plan organisational changes of software vendors that move to the cloud – Handbook. Downloadable at: <http://thecfram.wordpress.com>

## Abstract

Recent technologies have changed the way companies acquire and use computing resources. Companies have to adapt their *capabilities*, which combine business processes, skills, etc., to exploit the opportunities presented by these technologies whilst avoiding adverse effects. The latter part is, however, becoming increasingly difficult due to the uncertain long-term impact recent technologies have. This thesis argues that companies are required to adapt their capabilities in a way that increases the company's resilience so that they are robust yet flexible enough to succeed under uncertain conditions.

By focusing on cloud computing as one recent technology, this thesis first identifies the underlying processes of adapting capabilities to cloud computing by investigating how software vendors migrated their products into the cloud. The results allow the definition of viewpoints that influence the adaptation of capabilities to cloud computing.

Furthermore, the *Functional Resonance Analysis Method* (FRAM) is applied to one software vendor after the migration of their product into the cloud. FRAM enables the analysis of 'performance variabilities' that need to be dampened to increase the resilience of systems. The results show that FRAM appropriately informs steps to increase and measure resilience when migrating products into the cloud.

The final part develops cFRAM which extends FRAM through the viewpoints to enable the analysis of *capabilities* within FRAM. The goal of cFRAM is to enable companies to (1) identify existing capabilities, (2) investigate the impact of cloud computing on them, and (3) inform steps to adapt them to cloud computing whilst dampening performance variabilities. The results of the cFRAM evaluation study are unequivocal and show cFRAM is a novel method that achieves its goal of enabling companies to adapt their capabilities to cloud computing in a way that increases the company's resilience. cFRAM can be easily adapted to other technologies like smartphones by changing the viewpoints.

# Reading guide

This thesis combines and relates several different topic areas. The following paragraphs provide readers with guidance on how to find the information relevant to them faster. In addition, the reading guide can be used to identify and follow the major stories of this thesis that span across the different topic areas.

## **The effects of technological discontinuities like cloud computing**

Section 1.1 (properties of complex systems like cloud computing) -> Section 1.2 (properties of cloud computing) -> Section 2.3 (effects of cloud computing and similar technologies on users) -> Section 4.1 (effects of cloud computing on product development) -> Section 4.2 (internal impact of cloud computing) -> Section 5.2 (relationships in a cloud computing environment) -> Chapter 6 (effects of cloud computing on resilience) -> Section 8.2 (effects of cloud computing on capabilities) -> Section 5.3 (summary of areas that influence cloud computing decisions) -> Section 2.4.1 (the need to educate users about technologies like cloud computing) -> Appendix E (cFRAM: a method to plan the adoption of technological discontinuities)

## **The applicability of socio-technical systems (STSs) in today's environment**

Section 1.1 (properties of complex systems like cloud computing) -> Section 2.3 (effects of new technologies on STSs) -> Section 3.1 (introduction of adaptive STSs to take into account the effects of new technologies) -> Section 2.4.2 (a method to investigate STSs in practice)

## **Resilience Engineering (RE) for the IT industry**

Section 1.1 (properties of complex systems like cloud computing) -> 2.2 (definition of RE) -> section 2.3 (the need for RE in IT) -> section 2.4 (methods to achieve RE in IT) -> chapter 6 (application of methods in 2.4 in practice and cloud computing as example) -> Appendix E (cFRAM: a method for RE in IT)

## **Software vendors that plan to migrate their software products into the cloud**

Section 1.1 (properties of complex systems like cloud computing) -> chapter 4 (effects of cloud computing on product development and internal processes) -> chapter 5 (major adoption decisions for cloud computing and relationship with cloud provider and customers) -> Appendix E (cFRAM: a method to plan the migration into the cloud)

## **Understanding the actions of software vendors if you are a cloud provider or cloud end-customer**

Section 1.1 (properties of complex systems like cloud computing) -> chapter 4 (effects of cloud computing on product development and internal processes) -> chapter 5 (major adoption decisions for cloud computing and relationship with cloud provider and customers) -> Appendix E (cFRAM: a method to plan the migration into the cloud of future cloud providers and end-customers of cloud computing)

# Contents

Candidate's declarations.....	i
Permission for publication .....	ii
List of publications.....	iii
Abstract .....	iv
Reading guide .....	v
Contents.....	vi
<b>1 Introduction .....</b>	<b>1</b>
1.1 Academic merit .....	3
1.2 Problem space .....	4
1.3 Solution space and project aims.....	6
1.4 Novel contributions .....	7
1.5 Thesis structure .....	9
<b>2 Literature review.....</b>	<b>12</b>
2.1 Adapting capabilities to technological discontinuities .....	12
2.2 Resilience Engineering to succeed under varying conditions .....	16
2.3 The need for Resilience Engineering in IT .....	19
2.4 Succeeding under varying conditions in complex systems.....	29
2.5 Conclusions.....	36
<b>3 Methodology.....</b>	<b>37</b>
3.1 The need for socio-technical systems to become adaptable.....	37
3.2 A multi-stage study that adopts an adaptive STSs approach.....	39
3.3 Methodology of each stage .....	45
3.4 Conclusions.....	51
<b>4 Capabilities in the cloud.....</b>	<b>53</b>
4.1 Investigating the effects of cloud computing on product development .....	53
4.2 Investigating the effects of cloud computing on internal areas .....	62
4.3 Identifying the capabilities .....	68
4.4 Conclusions.....	75
<b>5 Emergence and relationships of cloud capabilities .....</b>	<b>77</b>

5.1	Major decisions during the adoption process of cloud computing .....	77
5.2	Influence of the cloud provider on customer satisfaction.....	90
5.3	The underlying processes of adapting capabilities .....	102
5.4	Conclusions.....	114
<b>6</b>	<b>Increasing resilience by migrating software products into the cloud.....</b>	<b>116</b>
6.1	Goals of the FRAM analysis.....	116
6.2	Elements and steps of a FRAM analysis .....	117
6.3	Methodology of the FRAM analysis with PP1 .....	122
6.4	Results.....	127
6.5	Discussion .....	135
6.6	Conclusions.....	142
<b>7</b>	<b>cFRAM – Planning the migration of software products into the cloud.....</b>	<b>144</b>
7.1	Motivation for relating the FRAM to capabilities .....	145
7.2	Intended research users and their requirements.....	146
7.3	Theoretical model and underlying assumptions.....	146
7.4	Steps of a cFRAM analysis .....	149
7.5	cFRAM handbook.....	164
7.6	Conclusions.....	165
<b>8</b>	<b>Evaluation of cFRAM.....</b>	<b>166</b>
8.1	Methodology .....	166
8.2	Results and discussion.....	170
8.3	Improving cFRAM.....	200
8.4	Conclusions.....	204
<b>9</b>	<b>Conclusions.....</b>	<b>206</b>
9.1	Lessons learned from the multi-stage study and FRAM analysis.....	207
9.2	Similarities of multi-stage study to packaged software development.....	208
9.3	Similarities of cloud computing to IT Outsourcing .....	210
9.4	Threats to validity .....	211
9.5	Future work.....	212
	<b>Bibliography .....</b>	<b>215</b>
	<b>Appendix A – List of multi-stage study first stage questions .....</b>	<b>227</b>
	<b>Appendix B – List of multi-stage study second stage questions .....</b>	<b>228</b>
	<b>Appendix C – Grouping of micro themes .....</b>	<b>229</b>

<b>Appendix D – Generic product lifecycle for FRAM analysis .....</b>	<b>230</b>
<b>Appendix E – Final cFRAM handbook.....</b>	<b>231</b>
<b>Appendix F – Ethics form .....</b>	<b>267</b>
<b>Appendix G – Glossary .....</b>	<b>268</b>

# 1 Introduction

Technologies like cloud computing or smartphones change the way companies acquire and use computing resources. Companies that want to adopt these technologies are required to change at various ends of their organisation in order to exploit the opportunities presented and to avoid adverse effects (Rogers 2003; Tushman & Anderson 1986; Woods & Dekker 2000). They are required to change at various ends of their organisation because cloud computing and smartphones are technological discontinuities (Tushman & Anderson 1986). Technological discontinuities create major technological shifts that affect business processes, skills of employees, knowledge, etc. These major technological shifts can be classified as capability-enhancing or capability-destroying. Capabilities, in very general terms, convert investments into assets (Ravichandran & Lertwongsatien 2005). In order to convert investments into assets it is necessary to combine resources of physical, human, and technological nature, in a structured way to achieve a specific goal (Grant 1991). The goals of IT department capabilities might be the acquisition, deployment, and leverage of IT (Bharadwaj 2000) to design IT architecture and deliver IT services that support the business operations (Feeney & Willcocks 1998). A concrete example of a capability for an IT department is provided by 'IT infrastructure management' to create and maintain dependable IT infrastructure (Feeney & Willcocks 1998). The capability achieves this by combining technology (e.g. servers, storage, and network), suppliers (e.g. of software, off-the-shelf or bespoke), and managerial skills (e.g. ensuring people acquire the right skills to operate servers and conflicts and problems with suppliers are handled). Other important IT capabilities can be IT project management (to ensure projects are completed within time, budget and quality), IT risk management (which could be a part of IT project management, as capabilities can be defined on different levels of abstraction), or IT support management (to ensure incidents and failures are dealt with in both a reactive and anticipative manner, Ravichandran & Lertwongsatien 2005).

When a technological discontinuity is capability-enhancing companies are able to exploit the technology by building on their existing capabilities. An example is provided by the transition from propellers to turbines on airplanes because airlines use their airplanes still in the same way to transport passengers or cargo (for aircraft

manufacturers this shift was capability-destroying, Tushman & Anderson 1986). When a technological discontinuity is capability-destroying companies are required to adapt their existing capabilities because the resources of capabilities have changed (e.g. business processes became obsolete or employees require new skills). An example is provided by the transition from break-bulk shipping to container shipping because container shipping enabled a faster loading and unloading of ships, required a change in business processes to enable standardisation and automation, and required fewer people per ship. Because cloud computing changes the way companies acquire and use computing resources, cloud computing is a capability-destroying technological discontinuity, as the following example illustrates.

Baxter et al. describe a case where a developer made use of cloud computing as a way to help the company produce an application more quickly (2012). In the short term everybody was happy with this. It was only much later, when the application needed to be modified, that problems arose. The developer had used his own machine and paid for the cloud services he had used with his own credit card. In the interim period he had moved on to another job. His previous company was denied access to the code for the application because they could not provide appropriate authentication details to allow them access.

The example illustrates that adverse effects can arise, particularly in the long term, when companies do not adapt their capabilities to cloud computing appropriately (Anderson & Felici 2012; Brynjolfsson & McAfee 2014; Hopkins & Jenkins 2008). Adapting capabilities to cloud computing appropriately is, however, challenging as the long-term technical and organisational effects of technological discontinuities are not always clear at the time of adoption (Brynjolfsson & McAfee 2014; Christensen 2013). This thesis argues that technological discontinuities require companies to adapt their capabilities in a way that increases the company's resilience. Resilience is defined as the ability to succeed under varying conditions (Hollnagel et al. 2006; Laprie 2008; Pearlson & Saunders 2010). Thus, capabilities that increase a company's resilience are robust yet flexible enough to address the long-term technical and organisational effects of technological discontinuities.

By adopting a Socio-Technical Systems (STSs) perspective this thesis seeks to investigate how companies can identify and address the long-term technical *and* organisational effects of technological discontinuities and why this is necessary to increase the company's resilience. STSs is a theoretical framework with a limited number of practical methods (Baxter & Sommerville 2011). This thesis relates the concept of capabilities, which is well known in industry, to STSs as capabilities capture technical and organisational resources similarly to STSs. This results in the following overarching research question of this thesis: *How can companies adapt their capabilities to technological discontinuities to increase the company's resilience?*

## **1.1 Academic merit**

Capabilities, in the context of technological discontinuities, are worthy of academic study as some of today's technological discontinuities, like cloud computing, present novel challenges. The original theory of technological discontinuities claims that technological progress is evolutionary with *rare* events of discontinuous change (Tushman & Anderson 1986). Discontinuous changes created a major technological shift, which could be classified as either *capability-enhancing* or *capability-destroying* as explained above.

Cloud computing, however, is a complex system that constantly changes (due to the unique characteristics of complex systems shown in Table 1). In other words, discontinuous change occurs *regularly*. Whereas the majority of literature on complex systems investigated their properties, this thesis seeks to investigate the impact of complex systems on its users. Users that rely on complex systems have limited control over when and how the system changes as many stakeholders control the system. Thus, as complex systems can change their properties and behaviour, companies adopting complex systems have difficulties anticipating their long-term effects. The emergence of increasingly complex systems, therefore, makes it necessary to adapt capabilities in a way that increases the company's resilience so that they are able to succeed when the properties and behaviour of complex systems change.

Table 1 - Characteristics of complex systems (according to Northrop et al. 2006)

No.	Explanation of characteristic
1.	Data storage, development, maintenance and operation are all decentralised
2.	Systems are developed and used by a variety of stakeholders which have conflicting, unknowable, diverse and changing requirements
3.	Systems continue to evolve and keep changing after they have been deployed. For example, new functionality is integrated while the system is being used
4.	Systems will contain heterogeneous, inconsistent and changing elements as they evolve over time
5.	The boundaries between stakeholders and systems erode. Stakeholders will not only be users of the systems. They will be a part of it. This affects the emergent behaviour of the system
6.	It will be the norm that software and hardware of the systems fail. Some part of the system will always be in a state of failure
7.	Acquisition and operation of systems will happen simultaneously. This requires new approaches for the development and governance of these systems

## 1.2 Problem space

The necessity of adapting capabilities to technological discontinuities has been concluded frequently (e.g. see Garrison et al. 2015; Tushman & Anderson 1986). What capabilities to develop has also been examined for some technological discontinuities (Bharadwaj 2000; Bharadwaj & Lal 2012; Lu & Ramamurthy 2011; Ross et al. 1996). The identification of the underlying processes of adapting capabilities to technological discontinuities, however, is under-researched, particularly for cloud computing (Alshamaila et al. 2013; El-Gazzar 2014; Khanagha et al. 2013). Identifying the underlying processes of adapting capabilities to technological discontinuities can inform future adoption of technological discontinuities like cloud computing.

The majority of studies on the adoption of cloud computing investigated general advantages and drawbacks of the technology (e.g. Gupta et al. 2013 for SMEs), took a user-centric, i.e. effects of cloud computing on the end-users (e.g. Liu & Orban 2008), or technology-centric perspective, i.e. technical challenges of adopting cloud computing (e.g. Rochwerger et al. 2009). Those studies that have focused on the

adoption of cloud computing from a STSs perspective only investigated factors that influenced the adoption decisions but not the adoption process itself (e.g. Alshamaila et al. 2013).

The importance of having to adopt a STSs perspective while adapting capabilities to cloud computing is best illustrated by SME software vendors that plan to migrate their software products into the cloud, for two reasons. First, the responsibilities of SME software vendors change after migrating their software products into the cloud because they will be positioned between the cloud provider and their own customers. Before the cloud, customers would install the software vendor's product in their own data centre and maintain it. In the cloud, the product is hosted with the cloud provider and managed by the software vendor, e.g. when updates need to be installed. At the same time, software vendors outsource tasks to the cloud provider over which they have only a limited amount of control (Bigdoli 2011; Khajeh-Hosseini et al. 2012; Zardari & Bahsoon 2011). In case the cloud provider changes the services they offer, the software vendor has to adapt accordingly. Hence, the kind and amount of change software vendors have to deal with (technical and organisational) is tightly coupled to the cloud provider and the software vendor's customers. Initial investigations in this area have been carried out by Afuah (2000) and Spedale (2003), although not for technological discontinuities like cloud computing. Afuah concluded that the capabilities of a company are affected by co-opetitors (suppliers, partners, complementors, etc.) and that capabilities reside in a network rather than a single organisation. Technologies like cloud computing, however, couple different organisations more tightly than before, as capabilities are not only affected in times of technological change, in the way Afuah concluded, but in everyday situations too (see Table 1).

Second, SME software vendors have limited resources (see Maglyas et al. 2012 for an extensive discussion on the differences in resources between SME and large enterprise software vendors). Many SME software vendors have a limited number of products from which they generate the majority share of their revenue (some have only one product). If that product should fail through a migration of it into the cloud, e.g. because customers do not want to adopt the cloud, these SMEs might experience

financial distress. It can also be the case that SME software vendors have to migrate their software products into the cloud as a defensive measure, e.g. when competitors choose to migrate theirs into the cloud, in order not to lose customers (Alshamaila et al. 2013; Porter 2008). Thus, SME software vendors need to investigate thoroughly which technologies to adopt and how to adopt them.

### **1.3 Solution space and project aims**

In the following, the overarching research question will be decomposed into two more explicit research questions that apply to SME software vendors and cloud computing. The answers to the two more explicit research questions will form the basis for answering the overarching research question proposed above.

Capabilities are appropriate when adopting a STSs perspective as they can combine organisational resources and technical resources on different levels of organisation. Rather than understanding *what* capabilities companies need to adapt to technological discontinuities, this thesis argues that it is more important to understand *how* companies can adapt capabilities to technological discontinuities. Focusing on the *how* can yield greater insights into the adaptation of capabilities to produce results that are transferable between different types of companies and technological discontinuities. In other words, it allows the identification of the underlying processes of adapting capabilities, an area that is under-researched (see section 1.2). The term ‘underlying processes’, however, is very broad. This thesis defines the term more specifically by meaning the following: identifying the technical *and* organisational challenges that influence the adaptation of capabilities, identifying the steps companies take to adapt capabilities and in what order they take these steps, and identifying the reasons for taking these steps. Research question 1 (RQ1), thus, is as follows: *For SME software vendors migrating their software products into the cloud, what are the underlying processes of adapting core capabilities to cloud computing?*

Core capabilities are those capabilities that are critical for a successful migration of software products into the cloud and from which software vendors derive competitive advantages (when this thesis refers to capabilities it always means core capabilities). Identifying the underlying processes of adapting capabilities promises to be

challenging, as one does not know beforehand how capabilities will adapt and what they will look like. To address this challenge, this thesis takes a high-level perspective. A high-level perspective is preferable over, for instance, a functional perspective such as Finances (similar to what Nuseibeh 2011 did for cloud computing) or a cross-functional perspective such as product development (similar to what Rimal et al. 2011 did for cloud computing). By taking a high-level perspective all kinds of issues can be captured in the early stages of research and gradually filtered to derive at the underlying processes of adapting capabilities to cloud computing.

Capabilities are appropriate for the concept of resilience and the aim to increase the resilience of companies because capabilities capture organisational routines. Technological discontinuities can affect these organisational routines (positively and negatively) and if companies do not investigate the effects of a technological discontinuity on their organisational routines they are neither able to exploit opportunities presented by the new technology nor are they able to avoid adverse effects (Anderson & Felici 2012a). When considering cloud computing, capabilities have to adapt, as previous organisational routines are unlikely to be appropriate due to the nature of cloud computing (it is constantly changing, contains a high number of dependencies and is vulnerable to cascading failure events, see Table 1). Thus, in order to increase the resilience of SME software vendors that plan to migrate their software products into the cloud, complementary organisational changes are required that need to influence and inform the adaptation of capabilities. Research question 2 (RQ2), thus, is as follows: *How can SME software vendors that plan to migrate their software products into the cloud increase their resilience with the adaptation of by capabilities affected by cloud computing and how can the increase be measured?*

#### **1.4 Novel contributions**

There are five novel major contributions and three novel minor contributions made by this thesis. The five novel major contributions are the following:

1. Development of the theory of **adaptive STSs** to provide a systemic approach that captures today's complex systems since, as evidenced by literature, the

original theory of STSs does not capture characteristics of current technological discontinuities. The theory of adaptive STSs is introduced in section 3.1 and also discussed in Werfs & Baxter (2013).

2. Investigation of the theory of technological discontinuities and capabilities with **five software vendors** that migrated their products into the cloud, where cloud computing is the example of a technological discontinuity. The methodology for the investigation is explained in sections 3.2 and 3.3. The results of the investigation are discussed in chapters 4 and 5 and in Werfs et al. (2013) and Werfs & Baxter (2014).
3. **Development of a framework** that identifies the areas that influence the adaptation of a company's core capabilities to cloud computing. Hence, the framework provides the answer to RQ1. It is described in section 5.3 and is based on the findings of chapters 4 and 5.
4. Application of **Functional Resonance Analysis Method (FRAM)**, a method designed to investigate the resilience of complex systems, to a software vendor after they migrated their software product into the cloud. The application shows that FRAM informs steps to increase and measure the resilience of software vendors that migrate their software products into the cloud. Hence, FRAM provides the answer to RQ2. The results of the FRAM application are described in chapter 6.
5. Extension of FRAM to enable the adaptation of capabilities. The extended FRAM will be called **cFRAM** (c for capabilities). cFRAM assists software vendors and other companies in planning the adoption of cloud computing, or similar technological discontinuities, by informing steps to adapt their capabilities in a way that increases the company's resilience. Hence, cFRAM provides the answer to the overarching research question by combining the answers to RQ1 and RQ2. cFRAM is explained in chapter 7 and evaluated in chapter 8. cFRAM is also explained in Werfs (2015).

The five novel minor contributions are the following:

- a) Sections 2.2 and 2.3 present a structured argument for the need of **Resilience Engineering in IT** to significantly increase the depth of discussion. Evidenced by literature, these sections identify and explain some of the challenges researchers need to overcome to inform approaches for resilience in IT.
- b) The **methodology** that structured the multi-stage study is a minor contribution as it can inform similar studies, possibly for other technological discontinuities, that would allow an extension of cFRAM and the comparison of the results between different technological discontinuities.
- c) The **cFRAM handbook** has informed the continued refinement of the handbook of the original FRAM (see Hollnagel et al. 2014 for the first edition of the handbook). More specifically, the cFRAM handbook is used to clarify the description of the FRAM method and examples that illustrate the steps of a FRAM analysis.

## 1.5 Thesis structure

This thesis is structured as follows:

Chapter 2 provides a literature review that describes the theory behind capabilities in more detail and explains drawbacks of existing frameworks for adapting capabilities. The second part of the literature review introduces Resilience Engineering and explains in detail why it is becoming relevant in the IT industry. To conclude this chapter, existing methods designed to achieve resilience in practice are discussed.

Chapter 3 defines the theory of adaptive STSs and explains the methodology of a multi-stage study for which an adaptive STSs approach was adopted. The multi-stage study follows five SME software vendors during the migration of their software products into the cloud. The overall goal of the study is the development of a framework that identifies the underlying processes of adapting capabilities to cloud computing. At the end of this chapter, the approach taken for each stage of the multi-stage study is explained in detail.

Chapter 4 describes the results of the first and second stage of the multi-stage study. With the results it is possible to identify the capabilities the five SME software vendors adapted to cloud computing. Identifying the capabilities is an important intermediate step towards identifying the underlying processes of adapting capabilities to technological discontinuities (i.e. to answer RQ1).

Chapter 5 describes the results of the third and fourth stage of the multi-stage study. With the results it is possible to identify and describe the underlying processes of adapting capabilities to cloud computing. The results are summarised in a framework that consists of four viewpoints (cultural, management, application, and governance). Hence, the framework provides the answer to RQ1.

Chapter 6 applies FRAM to a software vendor after they migrated their software products into the cloud. FRAM is used in chapter 6 to compare the functions of one of the software vendors from the multi-stage study before and after the migration of their software product into the cloud. The results explain how FRAM can be used to increase and measure the resilience of systems when adapting to technological discontinuities. Hence, FRAM provides the answer to RQ2.

Chapter 7 combines the findings from chapters 4-6 to introduce cFRAM. cFRAM extends FRAM through the framework with the four viewpoints (see above). cFRAM allows companies to plan the adoption of technological discontinuities, like cloud computing, by adapting their capabilities in a way that increases the company's resilience. Hence, cFRAM provides the answer to the overarching research question by combining the answers to RQ1 and RQ2.

Chapter 8 evaluates cFRAM with 14 companies and improves the method and its handbook based on the feedback collected from applying cFRAM with 14 companies. Overall, the evaluation of cFRAM shows that it is a useful and needed method to inform the adaptation of capabilities to cloud computing.

Chapter 9 concludes this thesis and identifies areas for future research.

Figure 1 summarises the structure of this thesis. A glossary of the most important terms used in this thesis is provided in Appendix G.

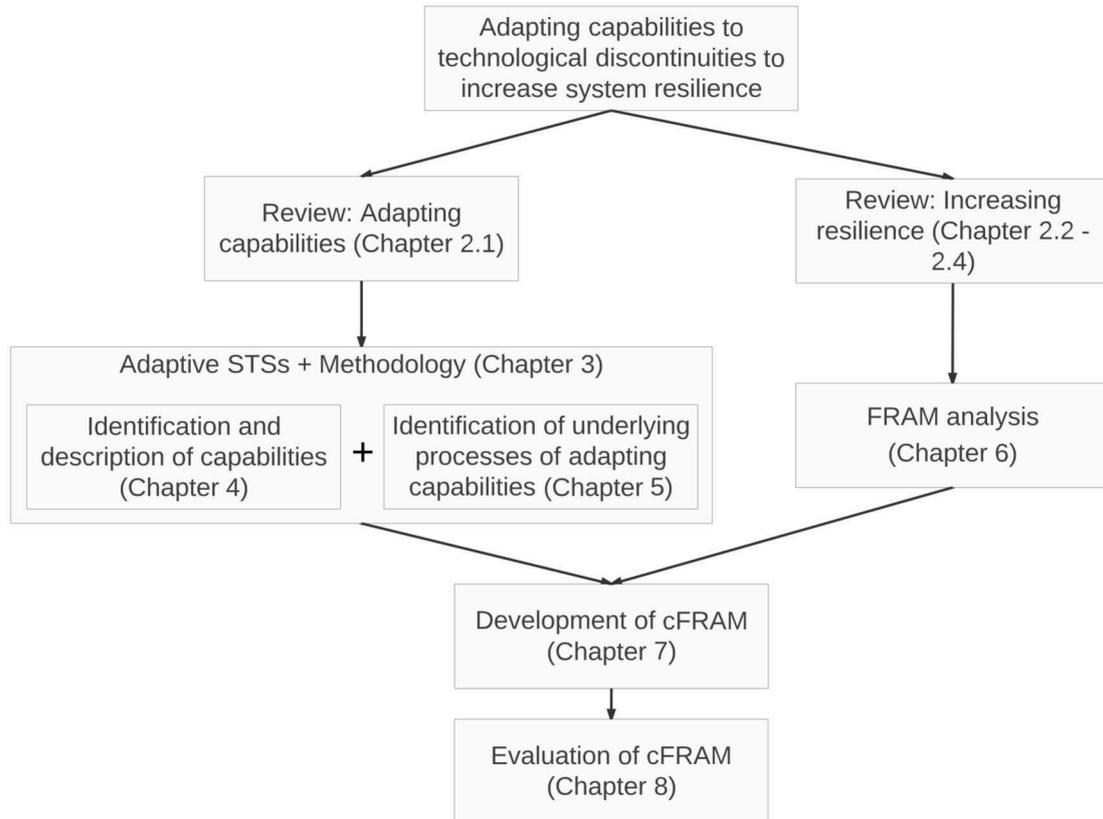


Figure 1 - Structure of thesis

## **2 Literature review**

The following literature review has two major objectives and one minor objective. The first major objective is the introduction of the theory behind capabilities and the identification of drawbacks of existing frameworks for adapting capabilities (see section 2.1). The second major objective is the introduction of Resilience Engineering (see sections 2.2) and an in-depth discussion of why it is becoming relevant in the IT industry (see section 2.3). The minor objective of this literature review is the discussion of methods that were designed to measure and increase resilience in practice (see section 2.4).

### **2.1 Adapting capabilities to technological discontinuities**

Capabilities are part of the resource-based theory of competitive advantage. The resource-based theory (or view, RBV) argues that a company's strategy is provided by its resources and capabilities (Barney 1991; Bharadwaj 2000; Grant 1996; Wernerfelt 1984). Resources in the RBV are the input for strategy development such as capital, skills, technologies, patents, etc. Capabilities combine various resources in a structured way to perform specific tasks. In theory, companies receive more sustainable competitive advantages from those capabilities that are "durable, difficult to identify and understand, imperfectly transferable, not easily replicated" and over which the company possesses clear control and ownership (Teece et al. 1997).

Capabilities are organisational routines of which SMEs are only able to perform a few in a highly efficient and near perfect manner through frequent repetition and organisational learning (Grant 1991; Leonard-Barton 1992; Lu & Ramamurthy 2011; Ravichandran & Lertwongsatien 2005). SMEs often find it difficult to adapt these routines to new circumstances. Furthermore, the development of capabilities is based on the company's existing organisational processes, its assets and its evolutionary path, all of which are not easily changed (Teece et al. 1997; Wade & Hulland 2004). SMEs need to be able to make informed decisions about where to deploy their resources and what capabilities to develop. Making informed decisions, however, is becoming more difficult, in part, due to the large number of new technologies (see section 2.3).

This thesis aims to investigate how software vendors can migrate (some of) their products into the cloud. Therefore, it is worth comparing the definition of ‘capability’ as used in this thesis with the definition in the Capability Maturity Model (CMM). The CMM aims to provide software vendors (and software engineering teams more generally) with guidance on how to improve their software development capabilities so as to develop better software (Paulk et al. 1993). The CMM, however, applies a slightly different definition of ‘capability’. It is important to highlight the differences to understand how the results of this thesis can, for example, inform the use of the CMM and *vice versa*.

The CMM defines capabilities as the range of expected results that can be achieved by the software development activities. In other words, capabilities in CMM primarily describe *what* results can be achieved. This thesis, in contrast, defines capabilities as describing *how* results can be achieved. Therefore, defining capabilities within the CMM can form the basis for the objectives this thesis aims to achieve. Similarly, the results of this thesis can form the basis for an application of the CMM.

To allow a more structured investigation into existing frameworks and methods for adapting capabilities to new situations and technologies, existing approaches are grouped into two categories: (1) structural change and (2) gradual change. Both are explained in turn.

Research in the category of *structural change* focuses on developing new capabilities and combining these with or replacing existing ones. The majority of approaches in this category propose one of three methods (Bower & Christensen 1995). First, create new organisational structures within the existing boundaries of the company. Second, create a spinout organisation to develop new capabilities. Third, acquire a company that already possesses the desired capabilities.

According to Bower & Christensen (1995), it is sufficient to alter the organisational structure to adapt capabilities. How a company can assure that the desired capabilities are developed is not explained. Moreover, how, once the capabilities are adapted, are

these migrated and spread through the entire company? Furthermore, if product development processes were required to change to adapt capabilities, how do these changes affect internal areas like sales & marketing or the learning of new skills? Pursuing structural change in order to adapt capabilities can lead to neglecting some of these questions, as the following two examples from industry illustrate.

Clark et al. (1997) describe how a telecommunications company changed their Information Systems (IS) organisation to react to changes from the external environment (i.e. creating new organisational structures). They defined change-readiness as the ability of the IS organisation to “deliver strategic IT applications within short development cycle times by utilising a highly skilled internal IS workforce”. They explain how the IS organisation created a centre of excellence to develop people with the right skills, and assemble and disassemble teams quickly. Although their research describes how the telecommunications company adapted to changes in the environment successfully, they fail to conclude whether the new capabilities made the company ready for change that occurred after the initial development of the change-ready capabilities. The authors conclude that the centre of excellence has worked well for the telecommunications company Bell Atlantic (today Verizon), it is questionable, however, if SMEs have the appropriate resources to build and sustain a centre of excellence and if they face as much change as a large organisation like Verizon, who have more employees and customers.

Galunic & Eisenhardt (Galunic & Eisenhardt 2001) propose “charter wars” between departments for technological discontinuities. A charter is a statement of purpose and includes the task, the market, customer characteristics, etc. A charter war tries to find the most suitable department for the charter, with the goal to develop a *dynamic community* of departments. In order for it to be successful, however, a company needs to possess different departments with similar skills that can ‘fight’ for a charter. For large, international conglomerates this is realistic (the company in their study was a Fortune 100 high-technology company). For SMEs, however, it is unrealistic. Although it helped the Fortune 100 company to recombine their resources and develop new capabilities, their study also showed that often the same departments win a charter and further extend their capabilities, whereas other, smaller departments, are

left behind. They concluded that this type of management puts a lot of pressure on employees.

The second category, *gradual change*, focuses on how to adapt, reconfigure, and deploy existing capabilities to address new situations. Gradual change approaches are captured in the definition of dynamic capabilities (a term coined by Teece et al. 1997). Dynamic capabilities describe an organisation's evolution over time and their ability to react to and anticipate changes in the environment (Eisenhardt & Martin 2000; Winter 2003). Research on dynamic capabilities is, however, very rudimentary and anecdotal (only one empirical study could be found; Ludwig & Pemberton 2011). The literature focuses on describing dynamic capabilities rather than explaining how they can be developed (Galunic & Eisenhardt 2001). Winter argues, for example, that the dynamic capabilities companies need to develop depend on the market. In so-called high-velocity markets (with high uncertainty and unpredictability), capabilities consist of a few simple rules that provide boundaries for action (Winter 2003). They miss to provide, however, examples of specific capabilities and explanations of how these were developed. Furthermore, the literature concentrates on change triggered by market situations (as they emerge, collide, split, etc., Eisenhardt & Martin 2000) and not on change triggered by new technologies.

An approach that builds on dynamic capabilities was advanced by Pan et al. (2007). They embraced the concept of modularisation to develop new capabilities. Modularisation can be defined as the intentional loose coupling of modules by standardising the interfaces between modules so that complementary ones can be combined more easily (Teece 2011). How this concept works in practice is clear for physical objects. It is unclear, however, how it would work with different types of resources that are being used by people (e.g. how can standard interfaces be created between teams of people?). Besides gaps in practicality, there is another drawback of pursuing modularisation. Modularisation requires companies to decompose their routines, reconfigure them individually, and put them back together. In other words, it follows a reductionist view and not a systemic view. When following a reductionist view it is argued that the system can be understood only by looking at its parts and that the whole is simply the sum of the parts.

For adapting capabilities to a new technology, however, a systemic view is advantageous as it is difficult to anticipate the effects of technological change (Woods & Dekker 2000). It is thus difficult to know beforehand which parts of the company will be affected by technological change and how they will affect each other (e.g. if the Sales & Marketing department changes, does Support need to change in a similar way to provide customers with a coherent experience?; Teece et al. 1997). Unknown interdependencies are particularly present in a cloud computing environment, due to the nature of complex systems (see Table 1 in section 1.1). In this case, capabilities are not only affected during times of change, e.g. during the adoption process, but also in everyday activities. Thus, new approaches for adapting capabilities are required that allow companies to investigate the interdependencies between different parts of the company and their environment. Furthermore, the new approaches need to allow companies to continually adapt their capabilities as the internal and external environment changes (see Table 1 in section 1.1).

## **2.2 Resilience Engineering to succeed under varying conditions**

Resilience Engineering, in contrast to more traditional approaches to disruptions, acknowledges that people need to adapt their behaviour to succeed under varying conditions, e.g. when adapting capabilities to a new technology. Adaptations to the behaviour are responsible for both failures *and* successes. Instead of trying to structure and constrain the tasks of people as much as possible to eliminate failures, Resilience Engineering aims to create processes that are robust yet flexible enough to succeed under varying conditions (Hollnagel et al. 2011). Voß et al. 2006, for example, carried out initial investigations for the necessity of making adjustments to actions, that can deviate from protocols in place, to maintain dependability of IT systems. Resilience Engineering acknowledges the need to use resources proactively to avoid adverse effects and to exploit opportunities that appear on the horizon (unlike lean management where resources are used as efficiently as possible). To understand how Resilience Engineering enhances more traditional approaches to disruptions, it is necessary to understand how Resilience Engineering emerged.

Over the last century, approaches to safety and risk management have changed significantly. They responded to the development of bigger and more complex systems (and systems of systems). Approaches to safety and risk management moved from linear cause-effect models (e.g. Fault Tree) towards multiple cause-effect models (e.g. Swiss cheese model) and finally towards systemic approaches (e.g. FRAM, Hollnagel 2012b, or Theoretic Accident Model and Process (STAMP), Leveson 2012). Systemic approaches emerged because accidents are presumed to result from the unexpected combination of multiple events. Systemic approaches also reflect the behaviour of people (as their normal performance varies while performing a task, in contrast to machines) and the possibility of emergent behaviour, that is necessary to consider when investigating intractable systems like STSs (Checkland 1999; Hollnagel 2012b; Rasmussen 1997).

The concept of resilience assumes that failures *and* successes both stem from performance variability. Traditional approaches to safety, like the Swiss Cheese Model, assume cause-effect relationships with linear consequences where failures could be clearly attributed to a simple combination of causal events, e.g. a malfunction (Reason et al. 2006). In complex systems, however, failures can also occur as a sequence of coincidences through emergent behaviour. Behaviour is classified as emergent if one cannot predict the behaviour of a system simply by looking at its parts or by decomposing it into its parts (Checkland 1999). In that case, no individual function of a system fails but the variability in performance of several functions (positive and negative variability) reinforce each other resulting in one function to exceed its limits of performance variability. The consequences are often disproportionate and unpredictable (Hollnagel 2012b).

Resilience has to be actively maintained over time by adapting: both reacting to change (through feedback loops) as well as anticipating change (through feedforward loops, Hollnagel et al. 2006). A failure can be evaded, for example, when people, systems or organisations are able to use the information, resources and time that is available to anticipate potential risks and make approximate adjustments to their behaviour (Hollnagel 2009). It can, therefore, best be understood as something a person, system or organisation does rather than something it has (Madni & Jackson

2009). In today's complex and dynamic environment the conditions of work (i.e. how a system operates) never completely match the way they were designed because it takes several years to implement a system; a time in which the environment and conditions of work continue to change (Hollnagel & Woods 2005). Resilience Engineering acknowledges the fact that in today's environment it is not possible to describe systems in detail because parts of it are intractable, i.e. they are characterised by emergent behaviour. People, therefore, play a vital role in maintaining resilience, because they are the ones who are flexible and adaptable by adjusting their behaviour to new information, resources or time constraints (properties that are often lacking in technological systems, Ignatiadis & Nandhakumar 2007).

The above introduction of Resilience Engineering makes clear that it is based on a systemic understanding of risks. It has been developed to take into account the fact that systems today are operating in an environment that is constantly changing as organizations react to both internal and external events (see Holling 1973 who originally suggested the term resilience). By combining ideas from Normal Accident Theory (NAT, Perrow 1984), High Reliability Organisations (HRO, Roberts 1990) and dependability (Laprie 2008) it is possible to define what resilience means for Information Technology (IT). NAT focuses on the dimensions of interactions (which range from linear to complex) and couplings (which range from loose to tight). In general, linear interactions and loosely coupled systems are regarded as safer and more reliable. There are, however, many examples of organizations that have relatively high numbers of complex interactions, and tightly coupled functions, yet have lower than expected numbers of accidents. These HROs achieve higher levels of reliability by making it an inherent part of everybody's job and embedding it into the organisational culture. Dependability, an emergent system property, describes the ability of a system to avoid failures that are more frequent or more severe, and outage durations that are longer than is acceptable to the system's users. Based on the concepts of NAT, HRO and dependability, resilience for IT can be defined as the adjustment of a systems functioning to maintain its dependability during changing conditions (this definition is consistent with ideas from Hollnagel et al. 2006; Laprie 2008; Tolerance 1992).

### 2.3 The need for Resilience Engineering in IT

Resilience Engineering is still a relatively young discipline, particularly within the IT industry (Nemeth & Herrera 2015). This section argues that Resilience Engineering is becoming more important for IT. In an IT environment, when processes fail, the consequences are not necessarily a loss of life (like in other industries where Resilience Engineering is more widespread, e.g. health care or nuclear power plants). The consequences are, however, often the loss of money and damage to the reputation of the company, as the majority of companies today rely on some form of IT or software. To understand the need for Resilience Engineering in IT in more detail, it is necessary to understand the underlying reasons for the use of IT in companies first.

IT is used to generate and use accurate, reliable and secure information (Dobbs et al. 2014; Van Grembergen et al. 2004). Information also needs to be provided to the right person, at the right time, at the right place via the right technology. This *equation*, however, is becoming increasingly dynamic as factors such as right time or right technology change more often (e.g. through the development of new technologies or the increasing deregulation of systems, see Table 1 in section 1.1, Bradley & Matson 2011; McDonald & Aron 2013; or see Burton & Willis 2014 for the Gartner Hype Cycle which is an oversimplified presentation of innovation but is useful to show the amount of emerging technologies, 45 in total). As some of the factors of the equation change more often, companies increasingly struggle to make informed decisions on what technologies to adopt and how to adapt their capabilities (see, for example, Craig et al. 2007 who argue that it is necessary to take a midrange view of technology in order to develop competitive advantages).

The underlying challenges of the above equation, i.e. not being able to anticipate the effects of technological discontinuities, are also stated in the “envisioned world problem” (Woods & Dekker 2000). The envisioned world problem can be decomposed into two categories to inform investigations into Resilience Engineering for IT. The first category will be called *use uncertainty*. Users increasingly expect to use the latest technologies because they also use them in other areas (see, for example, Bughin 2012 who argue that the boundaries between employees, vendors and customers will blur). These people are often described as technically savvy

workers that emerge from the Millennials generation (Brown 2013). Furthermore, some of these technologies can be used without the IT department knowing about them (e.g. smartphones or cloud computing, Baxter et al. 2012, Manyika et al. 2013). The second category will be called *technology uncertainty*. Recent technologies work differently by taking away control from the IT department and giving it to a third party, like cloud computing which gives control to the cloud provider (Cox & Alm 2008). IT departments also rely more on bigger and more connected systems (or systems of systems) that are vulnerable to unforeseeable and cascading failure events (Northrop et al. 2006).

Use and technology uncertainty are interdependent and can develop emergent behaviour (reflecting the need for carrying out organisational and technical changes together in a systemic manner). Through emergence the level of uncertainty increases exponentially because companies do not know which technologies people use and how they use them. Thus, the IT department has no way of assessing the impact of the technology on the company as a whole. The impact on the company as a whole can be particularly severe if complex systems like cloud computing are being used (for the reasons laid out in Table 1, section 1.1) In other words, the level of uncertainty can rise to a point where companies do not know what they do not know (called unknown unknowns, a term first coined by then US Defense Secretary Donald Rumsfeld in 2002, see Girard & Girard 2009). Unknown unknowns are likely to have a negative impact on any long term planning activities of companies for three reasons. First, it becomes more difficult to acquire and maintain the appropriate infrastructure if expectations of employees and customers change due to the development of new technologies. Second, it becomes more difficult to align IT with the organisation to develop competitive advantages when new technologies and changing customer requirements require a change in the business plan. Third, it becomes more difficult for companies to operate efficiently as the external and internal environment change more often and rapidly and companies are forced to adapt constantly.

Any efforts invested into developing a new approach that can assist companies in adapting their capabilities to technological discontinuities in a way that increases the company's resilience needs to address the challenges imposed by use and technology

uncertainty. The following three sections will explain how use and technology uncertainty affect the adaptation of capabilities.

### **2.3.1 Use and technology uncertainty affect the adaptation of capabilities**

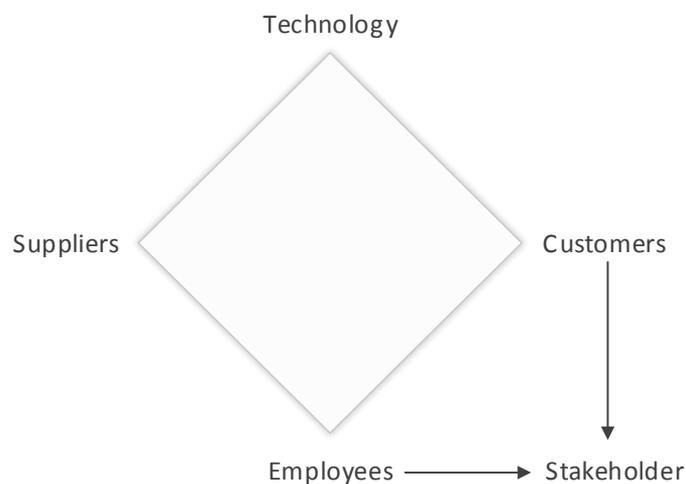
Figure 2 shows essential building blocks of the average company. A company comprises suppliers that supply raw materials, that are transformed through technology and employees to an end product that is distributed to customers (Brandenburger & Nalebuff 1995; Koller et al. 2010; Trist 1981).

Companies increasingly struggle to judge what technologies employees and customers (i.e. stakeholders) want and how they might react to new technologies. The main reason for this is that their expectations are transforming (defined as *use uncertainty*). Employees can use new technologies, like smartphones or cloud computing, without the company knowing about it. Using technologies without the consent of the company or IT department can have implications on procedures, e.g. making them opaque if different employees use different technologies to accomplish the same goal (procedures describe work as performed, processes on the other hand describe work as imagined, see, for example, ISACA 2014 for how cloud computing can change governance procedures). This affects the adaptation of capabilities as they, per definition, combine different resources in a structured way to achieve a specific task. When employees use new technologies (which are resources) without the consent of the IT department they cannot be combined in a structured way. Companies struggle to inform the adaptation of capabilities because they do not know what resources to combine.

Figure 2 also shows that technology itself has a significant influence on the way companies operate, i.e. what processes look like, the kinds of products they sell and the way they sell products (Rosenbloom & Christensen 1994). Companies need to decide carefully which technologies to adopt and which not, i.e. which technologies can help achieve the business strategy and satisfy stakeholder demands. The task of deciding which technologies to adopt, however, is becoming increasingly difficult, as companies do not know which technologies are here to stay (defined as *technology uncertainty*). This affects the adaptation of capabilities as they might be required to

adapt more often, or even constantly. The need for constant adaptation is present in complex systems (Table 1 in section 1.1). As complex systems evolve over time, e.g. components of the system or stakeholder demands change, capabilities have to incorporate and reflect these changes. The more often and rapidly complex systems change, the more often and rapid capabilities need to adapt. Otherwise, companies risk decreasing their resilience.

The following two sections will describe use and technology uncertainty in turn and explain how use uncertainty has an effect on informing the adaptation of capabilities and how technology uncertainty has an effect on the identification of IT risks.



*Figure 2 - Abstract representation of essential parts of a company (based on Brandenburger & Nalebuff 1995; Koller et al. 2010; Trist 1981)*

### **2.3.2 Use uncertainty makes the adaptation of capabilities more difficult**

Use uncertainty means that companies do not know if and how employees and customers will adopt new technologies. Recently the expectations and behaviours of employees and customers started to change more rapidly. It makes it more difficult for companies to continually assess them, thus, making it more difficult to adapt their capabilities with the existing methods introduced in section 2.1.

Competition among companies is becoming faster, more volatile and increasingly global (Crowston & Myers 2004; Roberts 2013). All three factors require companies to stay flexible, find new ways to innovate and create value (Peterson 2004). The

change in the business environment affects the expectations of stakeholders. No matter if they are customers or employees, stakeholders increasingly want information at the right time, at the right place on the right device without having to search for it (Roberts 2013). If companies are not able to respond to these challenges, stakeholders will take their own actions. Roberts (2013), for example, describes that younger people have no patience waiting for the IT department and a survey by Skok (2013) showed that many companies and employees use the cloud regardless of the IT department's opinion.

The issues around 'use uncertainty' have existed for many years and employees and customers were able to, for example, implement workarounds to circumvent the use of official technologies for many decades. These workarounds have been investigated frequently before, especially in the field of Computer Supported Cooperative Work (CSCW, see Borghoff & Schlichter 2000 for an introduction). Workarounds that have been investigated in more detail range from not using official software applications (see, for example, Timmons 2003) to the development of entire databases to support the use of official software applications (see, for example, Handel & Poltrock 2011). The advent of technologies such as cloud computing and smartphones, however, have changed the nature of user behaviour which affects workarounds in different ways. The changes in user behaviour can be explained in more detail through three aspects (Benson et al. 2014; Brynjolfsson & Saunders 2013; ISACA 2012):

First, the environment sees an acceleration of innovation (Brynjolfsson & McAfee 2014). More technologies are being developed and introduced into the market. Some of them disappear as fast as they were introduced. Others, however, are here to stay. Benson et al. see technology itself as the main factor for creating turbulence in the environment. They argue that every year new technologies promise to offer "bigger, faster, cheaper versions of essentially the same stuff" (2014). Yet, stakeholders expect to use these technologies as they increasingly use them in their private live for two reasons. First, new technologies are largely general purpose that can be tailored to specific tasks through software, compared to previous technologies that were designed for just one task e.g. increasing mechanical precision (see Hollnagel & Woods 2005 for an extensive discussion). Second, it is easier to buy new

technologies. A few decades ago there was a limited supply of technologies and places to buy them. They were also more expensive. Today it is relatively easy and cheap to buy new technologies over the Internet.

Second, some of the new technologies, like cloud computing, work differently compared to previous technologies, making the boundaries of systems less well-defined (systems in a general sense, like a department or a company, ISACA 2012; Badham et al. 2000; Hollnagel & Woods 2005; Northrop et al. 2006). With earlier technologies the way connections to the external environment could happen was constrained by the design of the system. The exchange of information with the external environment of systems took place in a regulated and structured manner (Werfs & Baxter 2013). Technologies like cloud computing or smartphones and current management practices—outsourcing, offshoring, value nets, value ecosystems, peer production, and so on (Porter 2004)—change that and make it easier for information to leave the physical boundaries of the company, thus circumventing any protocols for the exchange of information that are in place (the idea of *open systems*, systems that change their behaviour depending on what is happening in the environment, was first formulated in 1940 in the field of biology by Bertalanffy 1969).

Third, companies nowadays do not necessarily know which technologies stakeholders use, because some of the new technologies work differently. Until a few years ago the majority of companies were relying on computers and servers to do their information processing tasks. In this kind of environment the IT department was able to limit the rights users had on computers. Limiting the rights was straightforward as the IT department was responsible for buying and configuring the devices and all devices were part of the company's data network. More recent technologies, like smartphones or cloud computing, do not require access to the company's data network. Additionally, it is difficult for IT departments to block access to these services. Most cloud computing applications, for example, operate in an Internet browser and do not require access rights on the computer they are being executed on. Smartphones even do not require access to the company's data network. They are able to access the Internet via a carrier network. IT departments have no reasonable way of blocking

access to these technologies. Furthermore, they have difficulties finding out that they are being used. Both factors are a potential threat to the security, integrity and reliability of company information (see, for example, HP 2013 for how Hewlett-Packard defines the problem, as a large provider of IT services). It can also lead to a mixture of home and work devices, which usually have different security configurations, with the home devices being much less secure. ISACA (Information Systems Audit and Control Association) explain in more detail how the flexibility of cloud computing can enable people to circumvent authorisation, change processes, information security protocols or oversight processes (ISACA 2014).

Recent examples have shown that attempts by companies to ignore new technologies or forbid employees to use them failed. Baldwin describes an example where an employee bought a Laptop with the desire to use it at work (2013). The company, however, was not providing Wi-Fi so the employee decided to install a Wi-Fi router himself. After a while the Internet connection of the entire company was slower than usual. Executive management brought in an expert to investigate the situation. Eventually they found the unapproved Wi-Fi router. The Wi-Fi router had not been properly secured and someone from the outside gained access to sensitive company data such as passwords and usernames.

The above examples show that companies do not necessarily know what technological discontinuities are being used and how they are being used. Thus, the effects of use uncertainty require companies to adapt their capabilities in a way that increases the company's resilience so that they are robust yet flexible enough to succeed under varying conditions. The majority of frameworks to adapt capabilities discussed in section 2.1 (and more general change management methods, like the Balanced Scorecard) are several years old (some are even more than two decades old; the Balanced Scorecard was introduced in 1992, by Kaplan & Norton). It is questionable if they are still appropriate for technological discontinuities like cloud computing due to the reasons explained above.

### **2.3.3 Technology uncertainty makes the identification of risks more difficult**

Technology uncertainty means that the risks companies have to deal with start to change more rapidly (e.g. the spectrum of natural, human and environmental threats to IT, see for example Murray-Webster 2010). For two reasons the identification of risks is becoming more difficult.

First, some of the recently introduced technologies take away control from companies, forcing them to rely on a third party over which they have a limited amount of influence. In a cloud computing environment, for example, customers rent computing resources from the cloud provider (Mell & Grance 2009). Computing resources can be virtual machines, databases, virtual networks etc. (so called Infrastructure as a Service). It is also possible to rent entire computing environments where databases, for example, are already configured for a particular purpose (so called Platform as a Service). A third way of renting computing resources is the ability to rent ready-to-use applications such as customer relationship management systems or Office solutions (so called Software as a Service). In all three operating modes, the customer only rents the resources but has no physical access to them. The customer has to rely on the cloud provider to deliver what was promised. In case, however, a data centre from the cloud provider burns down, for example, the customer can only wait until the cloud provider has restored their services.

Companies like Microsoft propose a change in the mind-set for renting computing resources in the cloud (Mercuri et al. 2014). Previously, where the IT department was responsible for buying, configuring, and maintaining computing resources, the general wisdom was to avoid change as it could trigger failures. The operating efficiency of IT departments was often measured in the *Mean Time Between Failures* (see, for example, Engelhardt & Bain 1986, for an extensive discussion of the term). In other words, how long the IT department could operate without a failure in the network, servers, databases, etc. In a cloud computing environment, the IT department has no influence over the physical computing resources and how often they are reconfigured or updated. A new way of thinking was established where, instead of measuring the mean time between failures, IT departments measure the *Mean Time To Recovery*. The efficiency of IT departments is, therefore, measured in

terms of how fast they are able to recover from a failure and can restore operations back to normal.

The second reason why the identification of risks is becoming more difficult has to do with the fact that the IT industry is moving towards bigger and more connected systems, hence more complex systems, which are more vulnerable to unforeseeable and cascading failure events (Hopkins & Jenkins 2008). Within the last two decades they started to be not only computer-based (Clegg et al. 1996), but also sometimes computer-controlled (Åström & Wittenmark 2011). These increasingly complex systems (and systems of systems) have been characterised by the Software Engineering Institute as Ultra-Large-Scale Systems or ULS, which characteristics are shown in Table 1 in section 1.1 (Northrop et al. 2006, a similar idea was developed by the LSCITS project which stands for Large Scale Complex IT Systems, see Calinescu et al. 2010 for an introduction).

The use of increasingly complex systems has three important implications for IT risk management. First, as many technologies today are general-purpose technologies with no specific users in mind, they need to fulfil more requirements and have more functionality. A logical consequence is that more people are required to develop such systems, because each part of the system requires specialised developers (e.g. web developer and database developer). As more developers work on a system, they all understand their part of the system but find it increasingly difficult to understand the system as a whole. In fact, developers *and* users struggle to understand the whole system because they are becoming too big and complex (see Herritt 2014 who argues that we are reaching a point where no single person fully understands how complex technologies work).

Second, when different parts from various developers are combined to form a system, and people start using the system, emergent behaviour appears. Emergent behaviour is often the result of various people working together and means that the whole is more than the sum of the parts (Checkland 1999). Companies often desire emergent behaviour as it can lead to increased productivity and creative work results, but it also makes the adoption of new technologies more difficult. Emergent behaviour is not

predictable. While the system is being developed, the developers do not know beforehand how the technology will be used and thus what kinds of behaviour will emerge. Yet, system design is important so that one can be reasonably sure that the system being built only does what it is supposed to do and that it functions reliably (Hollnagel 2012b).

The third implication for the identification of risks is related to the reliability of increasingly complex systems. Most systems today are not developed in isolation. One system, for example, might rely on a different system for some form of functionality. If one system is not working properly, other systems might experience the effects and not work properly too. The results can lead to cascading and unforeseeable events. Cloud computing provides a good example. Many websites rely on Amazon's cloud services for some form of their functionality. When Amazon is having an outage, as happened on several occasions in the last few years, many websites are unreachable (Sultan 2011). Furthermore, although the majority of cloud providers state on their websites a reliability of 99.95% (or even higher), these numbers ought to be considered a marketing slogan rather than an accurate indicator for reliability.<sup>1</sup> The 99.95% reliability is based on past outages and only considers linear failures (i.e. one component fails and the failure has no immediate impact on other components). It would be more realistic, however, to acknowledge that the absence of outages in the past is no indicator for the absence of outages in the future and that dependencies between different components and systems can have a negative impact on reliability (i.e. a failure of one component can affect another component which can have effects on the data centre as a whole).

Another example is provided by mobile Internet. A recent report by the European Union Agency for Network and Information Security stated that 61% of mobile Internet outages were caused by system failures (i.e. software bugs, hardware failures, and system misconfigurations) and affected on average 1.4 million user connections (ENISA 2014). One of the underlying reasons for the increased number of outages could be that the telecom sector tries to respond to the increasingly dynamic and complex environment by launching new software faster, which is not as well tested as

---

<sup>1</sup> See, for example, the SLA of Amazon EC2: <http://aws.amazon.com/ec2/sla/>

in the past. Twitter, for example, allows its developers to experiment with any part of the Twitter system, as long as it only affects 1% of the user base (Amazon and Facebook operate in similar ways, Downes & Nunes 2014).

The effects of technology uncertainty on the identification of risks illustrated above show that if risks change capabilities need to change too so that major risks are being avoided and the resilience of the company does not decrease. If, however, risks change more frequently and companies struggle to identify risks in the first place, new methods for identifying and dealing with risks are necessary. Furthermore, the goal of this thesis is to develop a method that can help companies increase their resilience when adapting capabilities to technological discontinuities. The next section proposes two approaches for dealing with use and technology uncertainty.

## **2.4 Succeeding under varying conditions in complex systems**

Use and technology uncertainty require companies to succeed under varying conditions. In other words, companies need to develop capabilities that make them overall more resilient (Hollnagel et al. 2006). Capabilities to increase system resilience will be advantageous not only in times of technological change, like the adoption of a new technology, but in everyday activities too. Developing capabilities that make companies more resilient will, however, be challenging, as this area is under-researched. Righi et al. (2015) identified in their analysis of 237 studies in Resilience Engineering between 2006 and 2014 that 52% focused on the theoretical foundations, e.g. defining the term. The discipline thus lacks methods and frameworks to achieve resilience in practice (Righi et al. 2015 refer to 11 out of 237 studies that are concerned with achieving resilience in practice, e.g. case studies). This section will explore a theoretical model, Rasmussen's definition of skill-, rule-, knowledge-based behaviour, and a practical method, Hollnagel's FRAM, that combined could provide a way forward in adapting capabilities to technological discontinuities in a way that increases system resilience.

### **2.4.1 Companies need to have complementary organisational changes**

Before the emergence of use and technology uncertainty a closed system perspective, also referred to as *technological optimism*, was an appropriate viewpoint on technologies. When taking a closed system perspective things are assumed to go right, because systems are well designed and maintained, procedures are complete and correct, people behave as one expects and as one taught them, and designers can foresee and anticipate every contingency. Overall, people are seen as a liability and threat to the system and therefore their flexibility is minimised to achieve efficiency (Hollnagel et al. 2011). Use and technology uncertainty and the concept of technological discontinuities made clear, however, that organisational changes triggered by new technologies are not easy to foresee. Technologies (machines and automation in particular) are very good to tackle problems in predictable environments. They are suitable for predictable environments because risks can be clearly identified, assessed, and controlled. One does not need to be highly flexible. For uncertain environments and the use of complex systems, however, one does not know what the risks are. Flexibility and adaptability are needed—skills normally associated with people but less with technology. The argument that technology is too brittle was made by Dreyfus several decades ago but still seems to apply today (Dreyfus 1987; Dreyfus 1992). It is necessary to adopt an open system perspective, also referred to as *technological realism*. In an open system perspective, things are assumed to go right, because people learn to overcome design flaws, adapt their performance to meet demands, interpret, and apply procedures to match conditions, and people can detect and correct things that go wrong. Overall, people are seen as an asset that enable systems to function properly (Hollnagel et al. 2011).

People and organisational processes too can become *brittle*, however. Rasmussen developed the idea that people go through three stages of skilfulness (knowledge-, rule-, skill-based, 1983). At the beginning, people perform their jobs on a knowledge basis. They face unfamiliar situations and need to analyse the environment, develop plans, and test them. Testing can be done by trial and error or conceptually by predicting how the plan affects the environment. Once people get more familiar with situations, they move to rule-based behaviour. People have developed procedures through experience or adopted them from colleagues. They develop expertise about their job and are able to describe explicitly what they are doing. People are not able to

describe what they are doing when they move to skill-based behaviour. At this stage they are able to perform their jobs without conscious attention by relying on their procedures (Rasmussen 1983).

In less uncertain environments, rule- and skill-based behaviour are desirable. As risks can be clearly identified companies aim to address and control these as efficiently as possible e.g. by developing procedures. In uncertain environments, e.g. complex systems, rule- and skill-based behaviour are undesirable. In complex systems companies need to aim for knowledge-based behaviour, as people are required to constantly analyse the environment and adjust their behaviour according to new information. Problem solving skills like trial and error become more important so that companies are able to react to and anticipate new circumstances quickly and head off problems that appear on the horizon (e.g. when the cloud provider changes the services they offer or customers of software vendors demand a new product feature).

A contrasting argument is put forward by Suchman (1983). Suchman argues that the application of rules always also needs to include the application of knowledge and skills. The underlying reason is that system designers cannot anticipate all future states that employees might encounter. Thus, employees see it as their task to adapt any system so that the “smooth flow of office procedures” is ensured (Suchman 1983). System designers should, therefore, design systems in a way that enables employees, at all times, to analyse the environment and adapt the system to the current needs (in contrast to Rasmussen, where an analysis of the environment is only necessary when the tasks of employees or the environment change). For the aims of this thesis, however, Rasmussen’s and Suchman’s argument are compatible. Both state that in times of change, e.g. migrating products into the cloud, rules need to be redefined and skills adapted. The only difference is that Suchman suggests that the redefinition of rules and adaptation of skills is normal rather than exceptional.

As capabilities capture organisational routines (see section 2.1), and thus procedures, the above section explains why capabilities have to adapt to increase system resilience when companies want to migrate products into the cloud. The organisational routines, or procedures, are optimised for the technologies that are currently being used within

the company. Using new technologies, like cloud computing, exposes companies to a new environment that requires the analysis of new information, which can lead to an adjustment of organisational routines. Thus, to adapt capabilities to cloud computing appropriately companies are required to adopt knowledge-based behaviour to analyse new information and not rely on existing organisational routines. Over time it is then possible to inform the development of new organisational routines that form the basis for adapted capabilities. In that sense, capabilities can be used as a communication tool among employees. If employees are informed about the need to adapt capabilities they know that they have to adopt knowledge-based behaviour and question existing organisational routines. This can be particularly helpful for technically focused employees, e.g. software developers, that tend to neglect complimentary organisational changes new technologies require (see chapter 1).

#### **2.4.2 FRAM - A potential framework to investigate the resilience of systems**

The following introduction of the Function Resonance Analysis Method (FRAM) will be kept brief. The focus is on introducing the underlying principles of FRAM to explain the FRAMs advantages over other methods. Section 6.2 will focus more extensively on the practical application of FRAM, e.g. what steps are necessary for a FRAM analysis, as the sections after 6.2 will apply FRAM to a software vendor that migrates their software products into the cloud.

The concept of resilience and the nature of complex systems (see Table 1 in section 1.1) make it necessary to develop new methods and models to investigate complex systems (Hollnagel & Speziali 2008). Previously developed methods and models, like the Swiss Cheese Model, are inappropriate for resilience because they underlie different theoretical models and assumptions. Resilience requires organisations to be, at all times, responsive, attentive, anticipatory, and able to learn from past experience (Lundberg et al. 2009). The majority of older methods like the Swiss Cheese Model aim to constrain some or all of the requirements of resilience by building barriers that prevent people from doing something wrong. A drawback of building barriers is, however, that the adjustment of peoples' behaviour for valid reasons is blocked as well (see Voß et al. 2006 for a case study, or see Rasmussen 1997 for a comparison of the theoretical models and assumptions of methods for investigating the role of

organisational behaviour in accidents). The majority of models for investigating incidents take a bottom-up approach where a specific incident starts an investigation. Resilience, and this thesis, take a top-down approach where the behaviour of complex systems is investigated and analysed in different situations (Hollnagel 2012a).

FRAM builds on the concept of resilience and STSs. FRAM appears to be more appropriate than STAMP (Systems-Theoretic Accident Model and Processes, a concept with similar assumptions to FRAM, Leveson 2004; Leveson 2012) for complex systems, such as software vendors that migrate their software products into the cloud, because it allows the investigation of approximate adjustments that people have to make in complex systems (STAMP focuses on the development of feedback loops for information and control to impose adequate constraints that limit behaviour, Frost & Mo 2014; Herrera & Woltjer 2010; Hollnagel & Speziali 2008; Leveson 2004). Indeed, approximate adjustments is one of the four core principles of FRAM (Hollnagel 2012b):

- Approximate adjustments: FRAM acknowledges that the performance of STSs varies to adapt to current conditions in the internal and external environment.
- Equivalence of successes and failures: in comparison to older methods, FRAM acknowledges that successes and failures often have the same origin; or actions go right or wrong for the same reasons.
- Emergence: FRAM focuses on the investigation of emergent behaviour as many actions that result in success or failures should be investigated in light of the conditions that were present during the action. In other words, would an action have occurred at a different point in time, the result could be different.
- Resonance: Building on emergence, FRAM acknowledges that relationships and dependencies in complex systems constantly change. Thus, they should be analysed for a specific point in time or action and not considered as fixed cause-effect links.

Systems in FRAM are analysed by identifying functions that are necessary for everyday performance, i.e. top-down approach. Functions are abstractions that capture work routines. Once functions are identified, they are connected with each other

through one of six aspects, namely Input, Output, Time, Control, Precondition, and Resource (see Figure 3 for a summary of the aspects). The Outcome of one function could be the Input of another function (see Figure 4 for an example, section 6.2 describes the same example in more detail). It is not necessary to describe and connect all aspects of every function. In fact, it is recommended not to do so as the focus should be on the most pressing issues. Describing and connecting all aspects of every function can quickly make a FRAM analysis too complex.

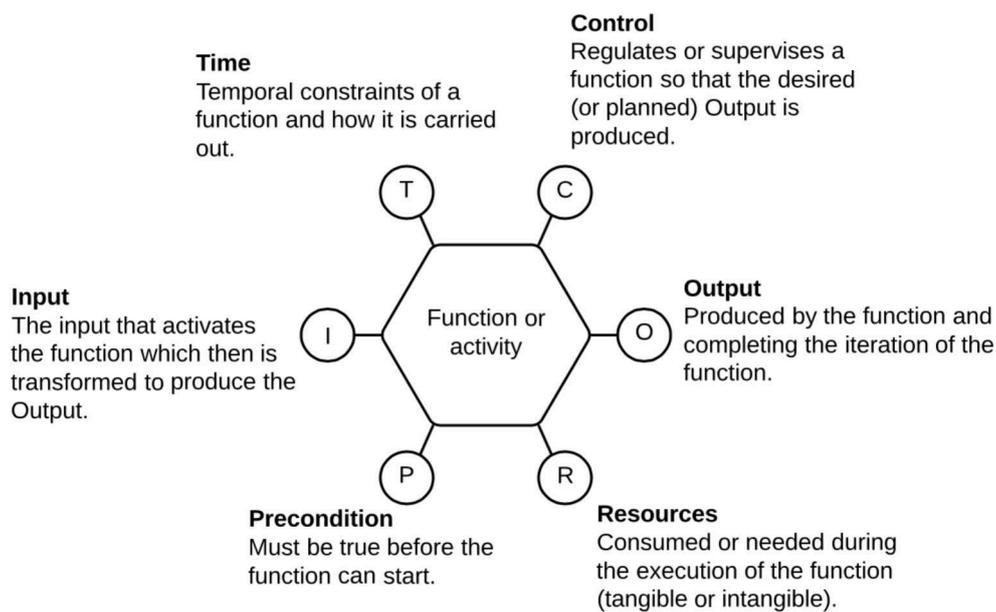


Figure 3 - The FRAM hexagon showing the six aspects and their descriptions (adapted from Hollnagel et al. 2014)

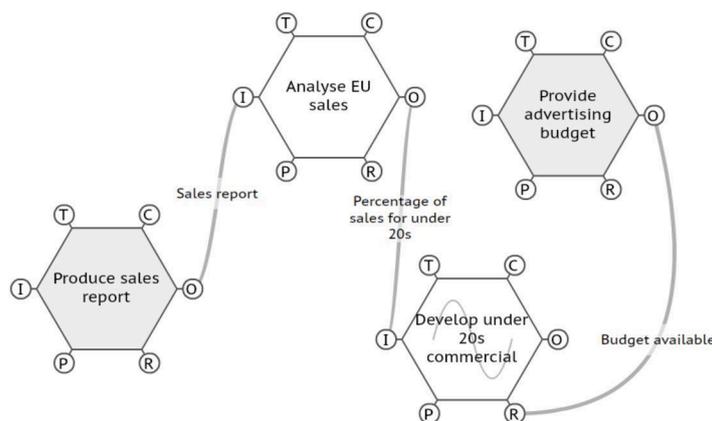


Figure 4 - Example of a FRAM model (grey functions are background functions)

FRAM focuses on the analysis of “functional resonance”, hence the name Functional Resonance Analysis Method (Hollnagel 2012b). Failures in today’s systems emerge

because the performance of functions vary and sometimes the variabilities reinforce each other causing the variability of one function to exceed its limits. With FRAM, it is possible to investigate the variability in performance of certain functions and analyse how the variabilities influence the behaviour of the system as a whole in different situations (see function <Develop under 20s commercial> in Figure 4 that experiences a performance variability when insufficient budget is available from <Provide advertising budget>, shown by the wave symbol in the hexagon). The aim is to dampen the unwanted variability (e.g. through changing how functions work, their couplings or by introducing new functions). The identification and dampening of performance variabilities aim to provide a way of measuring the resilience of companies.

FRAM allows the analysis of systems on different levels of organisation. It can, therefore, be used to increase the resilience of systems when moving to the cloud by structuring organisational changes that are necessary for the migration of software products into the cloud. For example, it is possible to build a FRAM model with known functions and use it to structure the investigation into understanding how other, possibly hidden, functions could influence the migration of software products into the cloud. The principles of FRAM, particularly emergence and resonance, support the application of FRAM on different levels of organisation. The results of this analysis can then be used to inform the adaptation of functions to suit cloud computing.

In its current form, FRAM should not be used over and above the analysis of resilience. Once performance variabilities have been identified with FRAM, it is difficult to use FRAM to keep track of organisational changes. When systems change, the FRAM model needs to be changed accordingly. Keeping the FRAM model up to date can become a resource intensive activity, especially if several people or departments were part of constructing the initial FRAM model. An unmaintained FRAM model can project a false sense of security. If, for example, all performance variabilities of the initial FRAM model are addressed but the FRAM model is not updated, a company might decrease their resilience instead of increasing it (because changing functions to address performance variabilities can change how the system as

a whole works thus creating new performance variabilities). Particularly in today's complex systems, the rate in which performance variabilities change has accelerated. Furthermore, FRAM is not useful to create a general set of principles that companies could follow to mitigate performance variabilities. Based on the findings of this chapter it can be concluded that there are no such principles. Complex systems change too often and fast and are too diverse as that there could be a general set of principles companies or systems could follow to adapt their capabilities to a technological discontinuity like cloud computing to increase their company's resilience (Rosenzweig 2007).

## **2.5 Conclusions**

This chapter provides two major conclusions and one minor conclusion. This chapter introduced the theory behind capabilities and identified drawbacks of existing frameworks for adapting capabilities. The first major conclusion of this chapter is that none of the identified frameworks describes the underlying processes of adapting capabilities. In addition, most are unfeasible for SMEs due to their limited resources.

This chapter also introduced Resilience Engineering and explained why it is becoming relevant in the IT industry. The goal of Resilience Engineering is to develop processes that are able to succeed under varying conditions. The second major conclusion of this chapter is that Resilience Engineering in the IT industry needs to overcome the challenges imposed by use and technology uncertainty. Use and technology uncertainty describe that it is difficult to anticipate the effects of technological change.

The last part of this chapter introduced FRAM. The minor conclusion is that FRAM is a promising candidate to enable the investigation of resilience in practice.

### 3 Methodology

This chapter has two major objectives and one minor objective. The first major objective is the definition of adaptive Socio-Technical Systems (STSs, see section 3.1). Adaptive STSs build on the original theory of STSs (Trist 1981) and extend it to include Resilience Engineering while acknowledging the challenges imposed by use and technology uncertainty (see chapter 2). The second major objective is the explanation of the methodology of a multi-stage study that follows five SME software vendors during the migration of their software products into the cloud (see section 3.2). The multi-stage study has four rounds (see section 3.3) and adopts an adaptive STSs approach. The overall goal of the multi-stage study is the identification of the underlying processes of adapting capabilities to cloud computing. The minor objective of this chapter is to make the reader aware of the fact that in order to increase the resilience of systems it is necessary to understand them first. Hence, the multi-stage study is a necessary intermediate step towards answering the overall research question of this thesis: *how can companies adapt their capabilities to technological discontinuities to increase the company's resilience?*

#### 3.1 The need for socio-technical systems to become adaptable

The term socio-technical emphasizes the importance of the interdependencies and interactions between the social and technical elements of systems, which can lead to emergent behaviours. STSs involve a complex interaction between different levels of organisation, in particular, between people, technology, and the environment in which the systems are deployed. Before STSs, engineers focused on the technical aspects, ignoring emergence and simply designing whatever the organization needed without changing the structure of jobs, i.e. neglecting organisational change (Trist 1981). The idea behind the socio-technical systems theory was to help design work on different levels of organisation, which could improve work functions while also improving technical performance (Walker et al. 2008).

With the original theory of STSs the behaviour of systems was influenced by the external environment only to a minimal extent (e.g. introduction of faster machines). The organisation of the systems, however, remained unaltered. Communication and

control, for example, were centralised and individual systems were controlled by people sitting in the same room. Networking technologies and distributed systems have changed the possibilities for communication and control of systems and made it possible to control systems remotely. The theory of Open STSs was developed to take into account that STSs could now operate and communicate with each other and with the environment (Badham et al. 2000).

The way that systems could communicate with the external environment, however, was controlled and could be constrained by the design of the STSs. In other words, there were protocols for how STSs communicated with their external environment and the boundaries of the STSs were clear. Hollnagel (2007) explains the situation of boundaries for the aviation industry. In the aviation industry a system can be defined on various levels, for example, the pilot, co-pilot and the cockpit form a system. The airplane, flight crew and ground personnel working on the aircraft form a larger system (that includes the previous system). This line of thinking can be extended further, for example, airports and air traffic control authorities form an even larger system. All the systems from the above example have different properties and therefore develop different emergent behaviour, thus, require different methods for communication and control (Checkland 1999).

Clearly defining the boundaries of a system and identifying the different perspectives are becoming more difficult in contemporary conditions (Anderson & Felici 2012b; Rasmussen 1997). Section 2.3.2 suggested that this is nothing new and that employees applied workarounds, e.g. to circumvent system boundaries, for many years (as part of Computer Supported Cooperative Work). The advent of use and technology uncertainty, through the continued growth in availability of new, cheap and free technologies (e.g. cloud computing and smartphones) let the boundaries of systems appear fuzzier than before as, for example, information can more easily leave the physical boundaries of a company. Thus, employees have access to forms of workarounds that affect both the behaviour *and* structure of socio-technical systems. It becomes clear that the changes in user behaviour triggered by these new technologies are not adequately captured by the current theory of open STSs.

In order to be able to deal with the problems from use and technology uncertainty that can arise with the introduction of cloud computing and other technological discontinuities, organisations have to be able to continually adapt to the internal and external environment. In other words, they need to function as adaptive socio-technical systems (Werfs & Baxter 2013). Doing so will allow them to react to events as they occur, which provides a mechanism for responding to failures and degradations in performance (Dalpiaz et al. 2013; Rasmussen 1997). In addition, organisations will be able to adapt in ways that are anticipative too, for positive reasons, such as exploiting expected opportunities in the market, as well as heading off problems that appear on the horizon (Hollnagel 2009). Hence, adaptive STSs enable companies to increase their resilience by having the intrinsic ability to change locally, both from a behavioural and structural perspective that allows them to be responsive, attentive, anticipatory and able to learn from past experience—necessary elements to increase system resilience (see section 2.3.3). Eason (2007), for example, describes how local adaptations helped to exploit technical capabilities while reducing costs and risks within a health care setting. Local adaptations can also adversely affect the wider STS, however, unless care is exercised to make sure that these adaptations are coherent and consistent with the entire organisation.

If adaptation is done in an ad hoc way, e.g. by the users of a technology, it can make processes and tasks become opaque (see chapter 2 for examples). Many companies currently proscribe the general use of technological discontinuities because they have not yet worked out a way to integrate them into their existing structures. An adaptive STSs approach would allow these technologies to be deployed in a more careful and controlled way, using ideas from experimental design (see Brynjolfsson & McAfee 2014 for a similar approach). In that way it should be possible to locally contain any adverse effects of using technological discontinuities, whilst at the same time providing a way to measure the potential benefits, and consider issues of generalisation before the adaptation is rolled out to other parts of the STS. In other words, adaptive STSs focus on gradual rather than structural change (see section 2.1).

### **3.2 A multi-stage study that adopts an adaptive STSs approach**

A multi-stage study allows the systematic investigation of RQ1: *For SME software vendors migrating their software products into the cloud, what are the underlying processes of adapting core capabilities to cloud computing?* The participating companies were five SMEs mostly working in the Oil & Gas industry (in the following called Project Partners or PPs). The Oil & Gas industry was chosen as it represents a large industry sector in Scotland, which provides a large pool of potential companies to interview. The five SMEs were contacted to participate in the study as all of them develop high-value software products without being direct competitors. Their products serve different needs of the market. Thus, the five PPs are more likely to speak openly in interviews and share their knowledge, as they do not have to fear a direct disadvantage when results from the study are shared. Furthermore, the author of this thesis worked with three of the five companies before (see Werfs 2012) and was able to establish a certain level of trust that made it more likely for the companies to be open during the interviews. Choosing companies that develop and sell high-value products provided a good sample because these companies face higher risks in case of adverse effects during the migration to the cloud. Thus, these companies are required to evaluate their steps more carefully, which helps the multi-stage study to identify the underlying processes of adapting capabilities to the cloud. Table 2 shows a short description of the five PPs.

*Table 2 - Short description of project partners of the multi-stage study*

<b>Project Partner (PP)</b>	<b>Type of software product that was the subject of the multi-stage study</b>	<b>Does the PP develop and distribute multiple software products?</b>	<b>Does PP offer their customers more than software products?</b>	<b>Main objective for a product migration into the cloud</b>
PP1	A project management software	No	Yes (business process consulting)	Expand customer base and market
PP2	A software tool to manage critical decisions	Yes (not all are suitable for cloud computing)	Yes (bespoke software development)	Be in control of product provision to increase resilience
PP3	A business process management software	No	No	Expand customer base
PP4	A software tool to manage risk management and safety assessments during drilling and well	Yes (parts of each product are combined to one cloud-based	Yes (business process consulting)	Expand customer base and control product usage

	engineering	service)		
PP5	A software tool to manage safety inspections on offshore Oil rigs	No	No	Expand customer base and market

The interviewees were either the Managing Director or leaders of Product Development. The multi-stage study was carried out over a 12-month time period (May 2013 to May 2014) with 18 interviews over four stages, each interview lasting around an hour (not all PPs were available for all the interviews due to more pressing issues they had to address). Interviews are superior over surveys or ethnographic studies for this type of research (Lazar et al. 2010). Surveys are suitable for reaching a large group of participants but they require a clear understanding of the phenomena that are being investigated in order to formulate well-defined questions. The goal of the multi-stage is, however, to *identify* the phenomena and then understand how the software vendors address them. Ethnographic studies are very time intensive. They require the researcher to embed with each software vendor and become a part of the team. They would allow the researcher to collect more in-depth data, compared to interviews, but are unsuitable for the study at hand. The aim of the study at hand is to compare how five software vendors migrate their software products into the cloud. Ethnographic research, however, is difficult to carry out when investigating several software vendors in times of change as one does not know beforehand when a software vendor changes (i.e. how would the researcher find out that software vendor B changes if he is currently embedded with software vendor A?).

As all PPs create and sell high-value software products, e.g. project management or time tracking software in the range of £100.000 per customer license, the goal was to investigate the impact of cloud computing on the PPs, and how they transform their existing software products into cloud-based services (i.e. Software as a Service, SaaS), or develop new cloud-specific products. At the time of the start of the multi-stage study, the PPs were at different stages of cloud adoption: some were already offering their products as cloud services (i.e. SaaS); others were currently migrating their software products into the cloud; and the rest were evaluating a migration into the cloud. By interviewing the PPs four times during the 12-month study period it was possible to follow them through the different stages of cloud computing adoption and

investigate them on different levels (Table 3 shows an overview of the stages which are explained in more detail below).

*Table 3 - Showing the research question and research method of the four stages of the study*

<b>Stage</b>	<b>Main research question and method used</b>
First stage	How did product development change in the cloud?
	Product development lifecycle
Second stage	How did internal processes change in the cloud?
	Balanced Scorecard
Third stage	What decisions did you make during the adoption process?
	Critical Decision Method
Fourth stage	How did you choose your cloud provider?
	Cloud resources process model

Interviewing PPs individually with the same set of questions and goals, regardless of their stage in the adoption, has advantages and disadvantages but promised to be more feasible than, for example, asking questions depending on their stage in the adoption. When asking questions depending on their stage in the adoption it is difficult to project when they will proceed to the next stage of adoption. For example, for some PPs it took 9 months to migrate their software products into the cloud, for others it already takes 3 years. In addition, it allowed for a more accurate comparison between the approaches taken by the various PPs as, for example, the more advanced PPs could comment on the plans of PPs in earlier stages and to what extent they were able to follow these in practice.

This thesis pursues a high-level perspective. A common framework for studies that aim to take a high-level perspective is the TOE framework (Technology-Organization-Environment). According to El-Gazzar (2014) the TOE framework has been used in the majority of cloud computing adoption studies. Since the introduction of the TOE framework in 1990 little development has taken place to further enhance it or make it more suitable to current challenges imposed by the environment and emerging technologies (see previous chapter and Baker 2012). Hence, the TOE framework will not be adopted by this thesis. Instead a socio-technical perspective was adopted and the particular questions asked were informed by taking an adaptive STSs perspective. The methods and objectives for each stage of the study were conceptualised and adjusted based on findings from previous stages.

In addition to following an adaptive STSs approach, the study was informed by a grounded theory (Glaser & Strauss 2009) and a case-study approach (Eisenhardt 1989). A grounded theory approach is appropriate for this kind of research as it can be adapted to the scope of research and works well in conjunction with case-study research (Pan & Tan 2011). Furthermore, grounded theory is well suited for the analysis of socio-technical systems in times of change (see, for example, Orlikowski 1993 who applied grounded theory to study the adoption of computer-aided software engineering tools). Grounded theory is also preferable over, for instance, hermeneutics (Myers 2004) or action research (Baskerville & Wood-Harper 1998; Susman & Evered 1978), as its aim is to create theory, which can be a step towards answering RQ1, as there is currently no theory that explains the underlying processes of adapting capabilities to technological discontinuities (Strauss & Corbin 1994).

The grounded theory is developed in two steps. The multi-stage study, of which the methodology is described in this chapter, forms the *framing cycle* of theory development according to Pan & Tan (2011). This means the theory will be constructed (see chapters 4 & 5 for the results). The latter parts of this thesis form the *augmenting cycle* of theory development (Pan & Tan 2011). This means that the constructed theory will be confirmed and validated through another study and the collection of additional data (see chapters 7 & 8 for the results).

Applying a case-study approach allows the analysis of cloud adoption and the adaptation of capabilities on multiple levels of analysis (Eisenhardt 1989, Pan & Tan 2011). Data was collected through an outside researcher, with no direct involvement in the PPs, who carried out semi-structured interviews. In other words, the researcher was a neutral person with the sole purpose of collecting data while not being aligned with the PPs or individuals within (Walsham 2006). All interviews were tape-recorded in addition to notes taken during the interviews (except for two due to the wish of the interviewee not to be tape-recorded). Recordings, although not required for grounded theory studies (Glaser & Holton 2004), have the advantage of allowing the interviewer to concentrate on the questions and interactions with the interviewee. In addition, they allow the re-analysis at a later time (as was done for all stages of the

study). On occasion the recordings and notes taken were supplemented by secondary data, e.g. presentations or reports provided by the PPs.

The collected data was analysed in two ways. First, a within-case analysis was carried out which involved writing a small summary of the findings after every interview (Eisenhardt 1989). Second, a between-case analysis in line with grounded theory was carried out by coding the notes and recordings to identify themes (Eisenhardt 1989; Miles & Huberman 1994).

Combining data analysis with data collection allows taking full advantage of the grounded theory approach taken for this multi-stage study. Based on the findings, the methods and data collection process can be adjusted (Eisenhardt 1989, Pettigrew 1990 in Pan & Tan 2011). Indeed, throughout the study the methodology for each round was adjusted to consider the results from previous rounds.

The main disadvantage of the grounded theory approach is the threat of the researcher injecting bias into the data. Several steps were taken to minimise the threat and to separate the signal from the noise. The results of each stage were discussed with three researchers from different backgrounds. Furthermore, the results of each stage were summarised in a report and sent to the PPs with a request for comments and feedback. However, the possibility of biases in the analysis of the results cannot be completely ruled out.

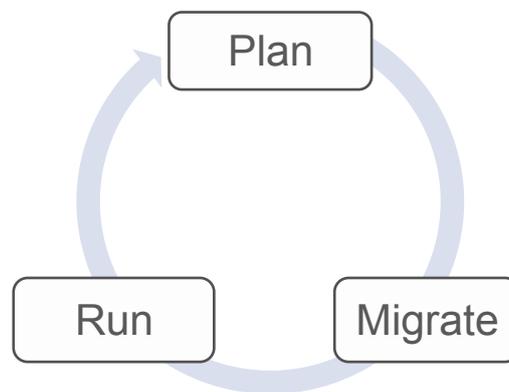
In summary, a grounded theory approach combined with a case-study approach can give a better understanding of the underlying processes of cloud adoption. A qualitative study is appropriate as it enables a deeper analysis of factors influencing cloud adoption by SMEs. Semi-structured interviews provide the opportunity to understand complex situations and explore all the factors while enhancing flexibility (i.e. answering how and why capabilities were adapted, Klein & Myers 1999; Leedy & Ormrod 2005). In the following, the methodology of each stage will be explained in detail.

### 3.3 Methodology of each stage

The first stage of this study investigated the effects of cloud computing on product development of the PPs (see section 4.1 for the analysis). The second stage investigated the effects of cloud computing on internal areas of the PPs (see section 4.2 for the analysis). The third stage identified major decisions made by the PPs during the adoption process of cloud computing (see section 5.1 for the analysis). The fourth stage investigated the impact of the relationship between the PPs and their cloud provider(s) on customer satisfaction (see section 5.2 for the analysis).

#### 3.3.1 Investigating the impact of cloud computing on product development (first stage)

The development of the questions for the first stage of the study was informed by a generic product development lifecycle (see Figure 5). By using a lifecycle it should be possible to adopt a long-term view, which is necessary as some actions might make sense in the short term but could have adverse effects in the long-term. In the following, the three stages of the lifecycle are explained.



*Figure 5 - Generic product development lifecycle that was used to structure the interviews*

The plan phase concentrates on the development of a strategy for the product, which should align with the overall organisational objectives. In this phase the company needs to decide why and what to use cloud computing for, as well as the resources that are required. The company also needs to review the impact of migrating their software products into the cloud on the entire company. The basic question that arises

here is, what are the issues influencing strategic decisions for the use of cloud computing?

The migration phase focuses on designing and implementing the product. In this phase the company needs to make decisions about the skills and methods that need to be employed. In addition, they have to consider the needs of their users and decide how to realise the product in the cloud. Two basic questions that arise here are, which areas of a company are affected in what way through cloud computing and how is cloud computing affecting the software vendor as a whole?

The run phase focuses on providing the product at the right time to the relevant customers. In addition, the company needs to monitor customer experiences of using the product and identify and prepare appropriate modifications to the software product, based on customer requests or incidents. The basic question that arises here is, how is cloud computing affecting product or service development and distribution?

Figure 6 shows the high-level questions the PPs were asked (Appendix A shows a full list of questions). The questions were validated beforehand during an interview with the leader of Product Development who had been closely involved in the successful adoption of the cloud in a company *outside* of this project.

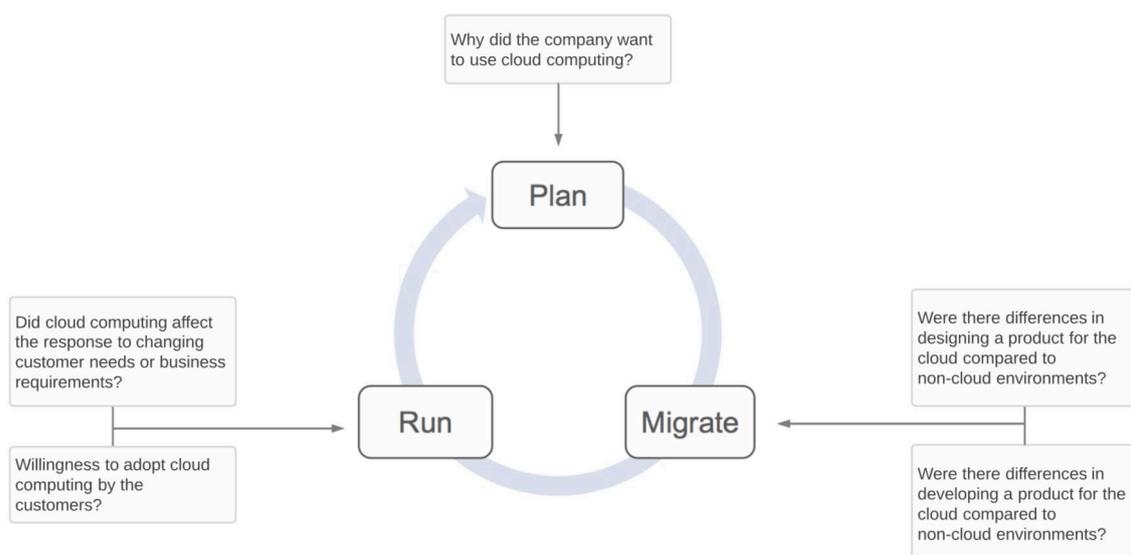


Figure 6 - Generic product development lifecycle that includes the high level questions that were asked for each phase

### **3.3.2 Investigating the impact of cloud computing on internal areas (second stage)**

The development of the questions for the second stage of the study was informed by the results from the first stage. As the goal of the second stage is the investigation of the impact of cloud computing on internal areas, the questions have been organised by adopting a Balanced Scorecard approach. Although the Balanced Scorecard model is decades old and thus might be inappropriate (it was introduced in 1992 by Kaplan & Norton, in fact the majority of change management methods have been introduced in the 90's and only sporadically updated since), it is one of a few change management methods that take a holistic perspective. By adopting a Balanced Scorecard approach the interviews focused on four main areas: (1) Customers, (2) Internal business processes, (3) Learning & growth, and (4) Financials.

As cloud computing offers access to reliable and scalable infrastructure without large upfront investments in software or hardware it offers SME software vendors new opportunities to enhance their software products and services. Furthermore, cloud computing is often referred to as a technology to increase the flexibility, agility and efficiency of Information Technology (IT). Cloud computing could, therefore, help organisations to increase their productivity by focusing on core tasks, e.g. developing software product or services, while outsourcing secondary tasks, e.g. maintaining infrastructure. Yet, few take into account that cloud computing can have a broader impact on the organisation as a whole and is not only a technological solution to a problem. This means cloud computing could enable companies to rethink their existing way of doing business which would affect all areas of a company.

From this line of thinking the following high level questions for each area of the Balanced Scorecard emerged:

- Customers: How is cloud computing affecting the relationship with your customers?
- Internal business processes: What is the impact of cloud computing on internal business processes (both IT and non-IT related)?
- Learning & growth: How is cloud computing helping you to grow and what effect does it have on your learning culture?

- Financials: How is cloud computing affecting your financial planning?

Appendix B shows the full list of questions that were asked in the second stage.

### **3.3.3 Investigating the decisions during cloud adoption (third stage)**

The first and second stage of the study developed an understanding of how the PPs migrated their software products into the cloud and what they did during the migration. The third stage was designed to investigate why the PPs did what they did. By focusing on *why* the decisions made by the PPs during the adoption are described together with the factors that influenced the decisions and what the outcomes of the decisions were.

The investigation of the decision making process borrows from the Critical Decision Method (Klein et al. 1989), a method used to analyse decisions that are made in complex, dynamic, high pressure situations. The method consists of several steps that were applied to every interview. First, an incident where critical cloud computing decisions had to be made is identified. For the purpose of this thesis the focus was on those decisions the companies have to make from the time when they decided to move into the cloud to the time when they distributed their first product through the cloud. The focus was on this period because this is when most of the decisions about cloud computing needed to be made for the first time. Those PPs that are not yet distributing their products through the cloud were asked what they thought they needed to do next to enable them to deliver their product through the cloud. Second, the interviewees were invited to describe what happened during the incident using the following question:

*“What happened in your company from the point where you reached an agreement on using cloud computing to the point where you distributed your first service or product through the cloud?”*

While the interviewee explains what happened, the interviewer draws a timeline of the events. The interviewer reads back the timeline of events to the interviewee before identifying the decision points together. Afterwards, the interviewees were asked

more detailed questions about the individual decisions. The following questions for the three decisions each interviewee identified as most important were asked:

- What was the type of decision, i.e. who was involved, who was consulted, and what kind of information was necessary?
- Were there any dependencies, e.g. to processes or other decisions and actions?

### **3.3.4 Investigating the influence of the cloud provider on customer satisfaction (fourth stage)**

Choosing a cloud provider is one of the most important decisions companies have to make while migrating their software products into the cloud. The PPs, as software vendors, are positioned between the cloud provider and their own customers (see chapter 1). The software vendor has to simultaneously look in two directions: to the cloud provider, because what they can offer their own customers is determined by what the cloud provider is giving them; and to the customers, because they need to ensure that the cloud provider can fulfil any customer requirements that the software vendors cannot directly satisfy.

In the first stage of this study the effects of cloud computing on the relationship between the PPs and their customers were investigated. In this final round of the multi-stage study the other side of the relationship was investigated: how the PPs chose their cloud provider and what the effects on the PPs' customers were. The focus is on how the PPs initially chose a cloud provider, how their approach changed as they gained experience from migrating their software products into the cloud and how they decided whether or not to use more than one provider.

A generic process with four steps to structure the interviews was developed, as shown in Figure 7. The steps were designed to highlight the four main issues that companies have to consider when deciding which cloud provider to use: choose a cloud provider; subscribe to cloud resources; use cloud resources; and release cloud resources. The generic process provides a holistic perspective for the investigation and allows the exploration of the initial choice of cloud provider as well as the long term effects of

using the cloud with a particular provider on the PPs and their customers. In the following the four steps will be explained in turn.

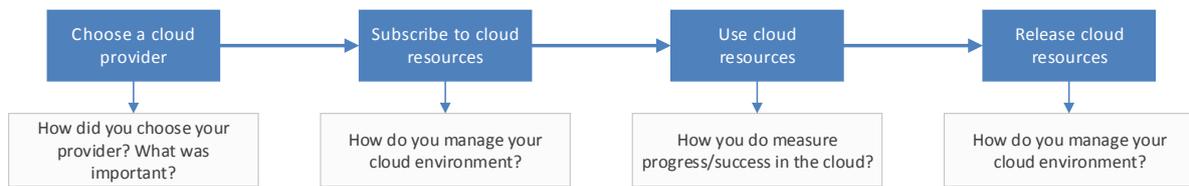


Figure 7 – Generic process to structure the fourth stage interviews

The first step is *Choose a cloud provider*. The main question asked in this part of the interview was, how did you choose your provider? A particular focus was on the decision making process and what kinds of requirements potential cloud providers needed to fulfil. Additionally, the focus was on whether the PPs had to make any trade-offs.

The second step is *Subscribe to cloud resources*. For this step two main questions were asked. First, how would you describe the relationship with your provider? The aim was to find out if the PPs have a close relationship with their provider, e.g. if they have a single point of contact, or if they are one among many customers. Additionally, the aim was on whether the cloud providers had influence on the PP's products, e.g. if they enabled or stifled products or features. The second main question asked was, how do you manage your cloud environment? The aim was to understand the internal processes for subscribing to and releasing resources. For example, who manages the environment and how is the use of resources measured?

The third step is *Use cloud resources*. The investigation for this step centred on how the migration into the cloud affected the overall performance of the PPs. The main question the PPs were asked was, how do you measure progress/success in the cloud?

The final step of the generic process is *Release cloud resources* after which companies could potentially move to another cloud provider. The questions the PPs were asked here were similar to the ones asked for *Subscribe to cloud resources*.

### **3.4 Conclusions**

This chapter introduced the theory of adaptive STSs. The term ‘adaptive STSs’ has been used loosely by previous works such as Dalpiaz et al. (2013) and Rasmussen (1997). This chapter, however, has provided the first clear definition of the term. Adaptive STSs provide a systemic viewpoint that allows the analysis of organisations that adopt complex systems while paying attention to the possible effects on system resilience.

Investigating the effects of cloud computing on the entire company and everyday work and development processes to identify the underlying processes of adapting capabilities will be done by a multi-stage study with five SME software vendors that plan to migrate their software products into the cloud. This chapter explained the methodology of the multi-stage study and described the goals and approaches taken for each stage. Figure 8 summarises this information before the next two chapters present and discuss the results of the multi-stage study.

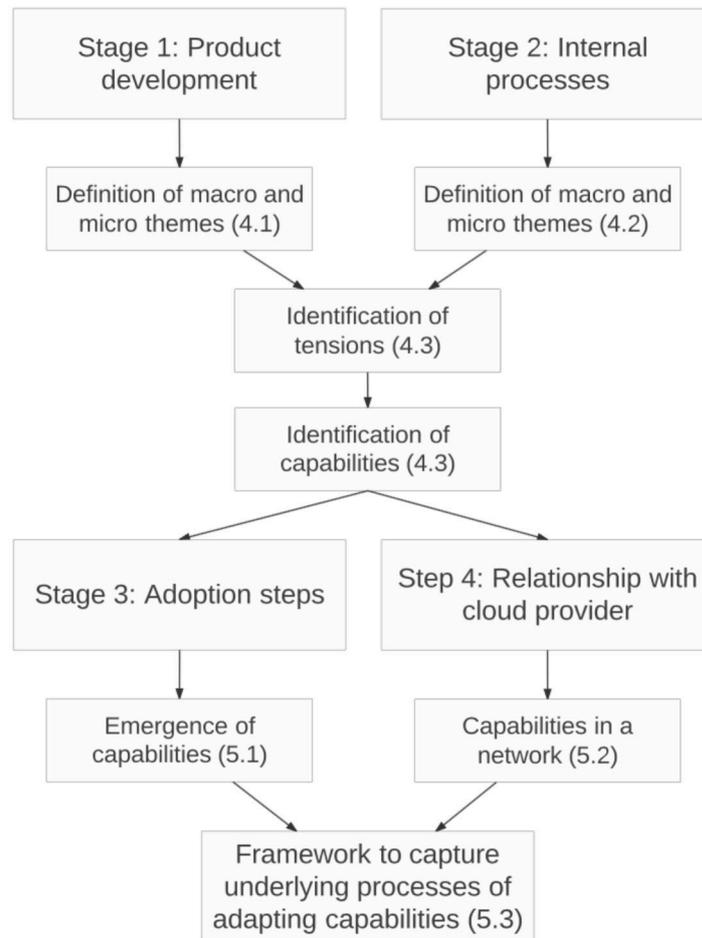


Figure 8 - Graphical presentation of the multi-stage study (the numbers in brackets refer to the sections in this thesis)

## 4 Capabilities in the cloud

This chapter has two main objectives. The first objective is the presentation and discussion of the first and second stage of the multi-stage that follows five SME software vendors (in the following referred to as PPs) during the migration of their software products into the cloud. The first stage investigated the effects of cloud computing on product development processes of the PPs (see section 4.1). The second stage investigated the impact of cloud computing on internal processes (see section 4.2). Based on the results of the first and second stage, the second objective is the identification of the capabilities the PPs adapted to cloud computing, (see section 4.3). Identifying the capabilities is a necessary intermediate step before the underlying processes of their adaptation can be identified and described.

### 4.1 Investigating the effects of cloud computing on product development

The notes and recordings from the interviews were analysed and compared for terms and expressions used by the interviewees. The terms and expressions were then categorised and grouped into macro and micro themes, in line with the grounded theory approach taken for this study (see Figure 9, Glaser & Strauss 2009). In the following the macro and micro themes will be described in detail.

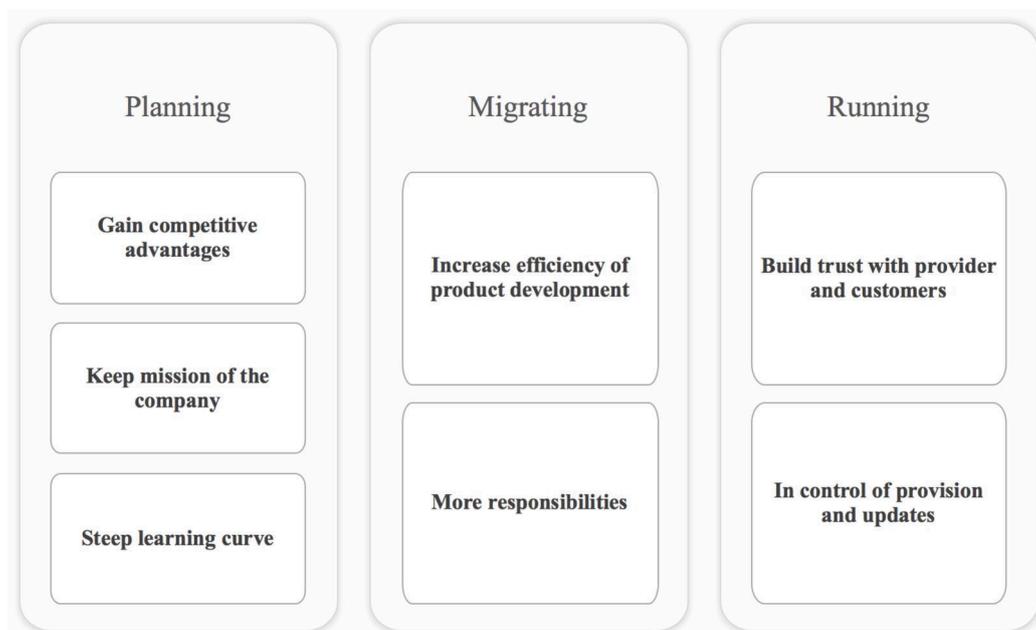


Figure 9 - Macro and micro themes identified from the notes and recordings taken during the interviews

#### 4.1.1 Planning a product migration into the cloud

Table 2 in section 3.2 briefly introduced the main objective for each PP to migrate their software product into the cloud. The following section will go into more detail to illustrate how the PPs achieved their objectives. Although the PPs had slightly differing objectives for the migration of their software products into the cloud, three general commonalities emerged that appeared to be necessary to achieve the objectives. They can be considered as part of the plan phase of the generic product development lifecycle introduced in section 3.3.1.

First, the PPs wanted to use cloud computing to be able to develop competitive advantages and compete on a global level. They saw cloud computing as a technology-push where the technology presents opportunities; the opposite would be a market-pull where customers demand a new technology (see also micro theme *Gain competitive advantages* in Figure 9). When products are provided through on-site installations at the customer's office, i.e. the traditional way of selling software products, it was necessary to cooperate closely with the customer. One example illustrating the need for cooperation is the necessity of including the customer's IT department in the sales process. The PPs had to ensure that the IT department would allow the installation of any necessary hardware and provide ways to access the company's network (e.g. get access to databases). With cloud computing it becomes more of a hands-off approach for the customer's IT department as Project Partner 1 stated (in the following Project Partner 1 will be referred to as PP1, Project Partner 2 as PP2, and so on). Because the PPs are now in control of the provision of the necessary computing resources in the cloud, it becomes easier to set up the software for the customers and provide access to their users.

Being in control of the provision has enabled the PPs to achieve two goals. First, they are able to give potential customers access to demo versions of their products. In the past, the inability to show potential customers how the product works has been a major issue (see also Alshamaila et al., 2013, who made similar findings). PP1, for example, tried to get around this by acquiring high performance laptops that could run several virtual machines to provide the necessary infrastructure their product requires.

Sales representatives of PP1 could then take the laptop to potential customers and demonstrate the product. Using high performance laptops, however, had two drawbacks. First, the demo environment was often slow and didn't contain any real customer data. Customers could, therefore, only imagine how the product would run with their own data. Second, PP1 would have had to acquire several laptops in order to demonstrate the product to different customers at the same time. Having to buy several high performance laptops, however, involves a large financial commitment for SMEs. In the cloud, PP1 can setup one demo environment that can be used for all customers. It is even possible to connect the demo environment to the customer's database so that the customers can experience the full potential of the product. PP2 and PP3 also set-up a demo environment in the cloud and all PPs have reported a smoother transition from potential customers to paying customers. PP2, for example, reported that in the cloud, it is possible to give customers access to a demo version within a few hours and if the customer should decide to buy the version, they simply have to change the licensing mode.

The second goal the PPs achieved by being in control of the provision is a change in the payment model of their products. PP4 reported that before being in the cloud, it was possible for customers to buy one product license and everyone in the customer's company was able to use the product. With cloud computing, where it is possible to have different payment models (among them subscription or pay-as-you-go), the PPs are able to charge e.g. per user or per transaction. PP4 hopes to make the pricing of their software fairer. PP1 is coming at the payment model from a different angle. Their product is relatively high cost, which required many people in the customer's company to sign-off on the product when a traditional on-site license was sold (i.e. the higher the cost of an investment the higher people in the hierarchy of a company have to approve the investment). By offering a renting model in the cloud, PP1 transformed the costs from capital expenditure to operating expenditure, i.e. a relatively small monthly fee. PP2 and PP5 adopted a renting model for similar reasons.

The second general commonality that emerged for the plan phase has to do with the mission of each PP and the evolution of the company over time (see also micro theme

*Keep mission of the company* in Figure 9). All PPs believe that they have a mature product (or products) that can be migrated into the cloud. The way they go about migrating those software products, however, varies significantly. PP3 and PP4, for example, used cloud computing as an opportunity to take a fresh look at their products. PP3 dismantled existing products and combined services from them into one cloud product. PP4 is currently thinking about offering a light version of their product in the cloud. PP1, PP2, and PP5, on the other hand, migrated their existing products into the cloud and only made small changes. Most of these changes were performance tweaks and none of them added cloud exclusive features.

The steps that would be necessary to migrate products into the cloud were mostly unclear to all PPs. The reason for this is that all PPs changed the way they develop and operate in the cloud and that none of the PPs had a clear driver for migrating their software products into the cloud (e.g. the business model could have been a driver). PP2 and PP5 had to reinvent their product development lifecycle. Before migrating their software products into the cloud, they were doing mostly bespoke software development and reacted to customer requests immediately. In the cloud, they have more customers and were forced to take a more structured approach (see Guvendiren et al. 2014 for an extensive discussion of the transition from bespoke to standard software product development). Now they collect updates and only release new versions of their products a few times a year. PP3 experienced a similar development, although from a different perspective. Before the migration of software products into the cloud their Research & Development (R&D) activities were very much ad-hoc. For the migration of their software products into the cloud, they developed a roadmap for feature development. Part of the reason why they had to adopt a more structured approach was that they moved from a purely project-oriented company, where their consultants were responsible for most of the revenue, towards a product-oriented company, where software products are responsible for a larger share of revenue. In order to develop the roadmap they acquired outside help through knowledge transfer programs that got them in contact with consultants and academics (the author of this thesis was not among those academics).

The third and last general commonality that emerged for the plan phase was that all of the PPs experienced a steep learning curve (see also micro theme *Steep learning curve* in Figure 9). The steep learning curve can, for example, be seen by the cloud-operating model they adopted. None of the PPs adopted a pure cloud computing approach (at least not during the timeframe of the multi-stage study). All of them continued to offer alternatives to the cloud version, e.g. on-site installations. The underlying reason for offering both the cloud and non-cloud version can best be illustrated by the operating model PP3 adopted. PP3 reported that some of their customers are concerned about their data being stored in the cloud. PP3 will, therefore, not only continue to offer the on-site version of their product until customers have more confidence in the new technology (PP3 is trying to educate their customers by explaining that cloud providers have more expertise about security than they, as an SME, could ever have) but they are also offering the option to install the software in a private cloud (e.g. dedicated servers). In other words, the PPs are trying to create a smooth transition from the on-site version to the cloud version. PP1 stated that their customers had similar concerns about their data being stored in the cloud. Over time, however, these concerns have mostly disappeared. This shows, not only the PPs have to learn about the new technology (in this case cloud computing) but their customers have to analyse how the technology will affect them too.

#### **4.1.2 Migrating products into the cloud**

When the PPs started to migrate their products into the cloud they had to address issues concerning the differences in designing and developing a cloud product compared to an on-site product. The PPs were able to increase the efficiency of their product development efforts. At the same time, however, some of their tasks became more complex. The reasons for both are explained in turn.

The PPs were able to increase the efficiency of their product development efforts mainly because they designed internal operations more efficiently (see also micro theme *Increase efficiency of product development* in Figure 9). PP2 and PP5 used the migration into the cloud to reduce the internal IT and virtualise (or outsource) other services. PP2, for example, reduced the number of internal servers from 14 to 2 and virtualised Email, telephone, source code control, help desk, and the company portal.

They think that the cloud “makes [them] more resilient”, by giving the example, that their headquarters could burn down and they would still be operable.

Moving to the cloud also changed the work of the software engineers and the PP’s attitude towards other emerging technologies. PP2, PP3, and PP5 reported that their software engineers are excited to move into the cloud as it enables them to learn skills (to increase their attractiveness as an employee). For PP2 the software development in the cloud is a lot more focused on usability. Part of the reason for this is that PP2 switched from having a mature desktop application interface, which included many years of customer feedback, towards an immature web interface. PP1 reported that the work of their software engineers has become more dynamic. They believe this happened because they are able to give potential customers access to a demo. Before the customer commits to the product they sometimes request changes to it, of things they miss or do not like in the demo. The software engineers make temporary changes to the product demo, to show the customer what it could look like. If the customer then decides to buy/rent the product, the changes are made permanent. PP3 is achieving similar objectives from an organisational perspective. By migrating their software products into the cloud they will appear more professional because they make the installation of the product more user friendly, they can deliver the product globally, and update it more easily.

The PPs became more complex after migrating their software products into the cloud because their responsibilities started to change. For example, they are now responsible for the dependable operation of the products (see also micro theme *More responsibilities* in Figure 9). PP2 and PP4 both explained that they did not know everything they had to do in order to migrate their products into the cloud successfully. Both were also aware that there were things they did not know (this reflects the idea of unknown unknowns). Once the first products were migrated into the cloud, PP2 discovered that they, as a company, had to transform from mainly being a technical company towards being technically savvy *and* well organised (e.g. in terms of operation and support). PP2 “did not realise that was going to happen” and had to invest time in order to catch up with the responsibilities quickly so as not lose customers.

PP1, PP2, and PP5 also described that once they moved products to the cloud they discovered performance issues, which needed to be addressed immediately before customers would complain. PP1, for example, was trying to serve customers on the east coast of the USA and customers on the west coast of Australia with a cloud data centre located in California (USA). Customers in Australia, however, were frustrated with the performance and PP1 realised that it was not feasible. Instead of modifying the application, they decided to run two instances in the cloud nearer to the two locations.

PP1 started to alter the roadmap for their main product once they migrated it into the cloud, from a technical towards a functional focus. For example, they are thinking about splitting their product according to role-based functions and offer these on tablets or smartphones. Introducing role-based functions would enable them to tailor the product to specific roles users fulfil in their organisations (e.g. make a differentiation between a manager and an Oil platform worker). At the same time, it would make their product more complex with potential adverse effects. Introducing new features or fixing bugs, for example, can have wide-scale effects and “introduce new bugs in different places because of interdependencies and cascading events”, as PP1 stated.

PP4 summarised the tension between introducing new features more frequently and increasing the complexity of the product best. PP4 stated that “the company is becoming more complex but with simpler solutions for the clients”. The PPs are able to get more out of their resources and can focus on the tasks that matter to them: developing software. PP2 noted a similar conclusion. PP2’s software engineers want to focus on mission critical tasks and “not to look after hardware.”

### **4.1.3 Running products in the cloud**

Once the PPs’ software products were migrated into the cloud, they had to think about how they would continue their path in the cloud. Two issues played a particular role while making decisions about the future of their products: trust and control

The PPs had to build trust with their cloud provider and, at the same time, build trust with their customers (see also micro theme *Build trust with provider and customers* in Figure 9). The PPs had to ensure that the cloud provider would provide a secure, accessible, and reliable service. They also had to ensure that customers would trust the cloud product in terms of confidentiality and data security. One way to achieve this is through the SLA with the cloud provider and customers. PP1 invested a lot of time into the Service Level Agreement (SLA) with their cloud provider. They are working with a niche provider, which gave them more influence over the SLA than they would have had with a major provider, like Amazon or Google (PP3, for example, is working with Amazon and had to accept the standard SLA because as a SME they had an inferior negotiating position, PP5 reported a similar finding with Microsoft Azure). For PP1 it was particularly important to clarify issues such as uptime and penalties in case of SLA violations. They spent 9 months and 20 iterations negotiating the SLA with their provider (PP2 negotiated the SLA with their cloud provider in a similar way although with fewer iterations). Afterwards they made sure they had a “back to back agreement” with their customers so that PP1 would not get penalised in case the cloud provider is experiencing downtime. PP1 still made a negative experience with their cloud provider, although for different reasons (see next section).

PP2 is also aware of the change in responsibilities. They noted, “no matter how much you outsource, you cannot outsource the final responsibility of the product”. PP2 is well aware of the fact that the customers will first see them as the source of failure even if the cloud provider is ultimately responsible.

Despite the challenges of having to negotiate an appropriate SLA and being the first one to take responsibility if the cloud provider is experiencing downtime, the PPs see two characteristics in the cloud providers that can help them establish trust with their customers. First, the cloud providers have more expertise in terms of security because they are able to employ people that only focus on security. Employing people that solely focus on security is something the PPs (and probably most of their customers) would not be able to do. Since many of the PPs’ customers are concerned about data security and most of the PPs had two groups of customers after their initial migration

into the cloud (cloud and non-cloud customers), they see it as their responsibility to educate customers about security issues. Although PP3 was not at the stage of marketing their product at the time of the interview, they believe having the right marketing will be a key factor to convince people to migrate to the cloud. Other PPs, as noted above, see the issue of data security concerns disappearing over time and do not take any specific actions.

Second, the PPs are gaining back control over their product, although they lose control over their computing resources (see also micro theme *In control of provision and updates* in Figure 9). One of the questions asked during the interviews was, if the PPs feel they are losing control to the cloud provider because they outsource computing resources. The question was always negated. In fact, instead of losing control, the PPs feel they are being more in control (which is in contrast to what Sultan, 2011, found out for cloud computing, likely because they investigated IT departments and not software vendors). The PPs feel more in control for two reasons.

First, they are responsible for the dependable operation of their products because the products run in the data centres of the cloud provider that the PPs administrate. Controlling the operation of the products enable the PPs also to control the installation of updates. Before, with the on-site version of their products, the PPs could provide updates (e.g. bug fixes) but could not ensure that all customers were installing them properly or at all. The reason for this is that the customer's IT department was responsible for the dependable operation of the product. In the cloud, however, the PPs can release an update and it is instantly available to all customers.

Second, the PPs can track and analyse how their products are being used. Tracking and analysing product use enables them to see which functions or features are used or not used. They can then use this information, for example, to develop role-based versions of their product (as PP1 did, see section 4.1.2). PP2 goes beyond that and has developed a control panel where they can enable and disable features for certain customers. In the cloud, PP2 is having more users than before (when they were a bespoke software development company). It was a difficult task for them to decide which features should be included in the cloud version, as they only wanted to include

those features that are being used by the majority of customers. If they would have included features that are only used by a few customers they believe it would have “irritated the majority of users”. PP5 is going even further and tries to anticipate features that customers might want in the future. They develop these features at their own risk and only get a return on investment if these features are actually implemented.

## **4.2 Investigating the effects of cloud computing on internal areas**

The previous section investigated how cloud computing affects product development of the PPs. This section will focus on internal factors by adopting a balanced scorecard approach (Kaplan & Norton 2007). The four areas of the balanced scorecard, (1) customers, (2) internal business processes, (3) learning & growth, and (4) financials, have been used to structure the interview questions. The findings have revealed, however, that the four areas of the balanced scorecard are not entirely appropriate to explain the most pressing internal challenges the PPs faced. The following description of the results from the second stage has been organised around the following areas, that were identified by analysing, categorising and grouping the terms and expressions used by the interviewees (Glaser & Strauss 2009): (1) customer demands, (2) internal processes, (3) human resource management.<sup>2</sup>

Furthermore, the results from the data collected can be classified into actions and effects.

Actions have the following characteristics:

- They follow decisions made by the companies while migrating software products into the cloud;
- They are consciously taken by companies to exploit benefits of the cloud;

Effects have the following characteristics:

- They result from the use of cloud computing and may not exist outside the cloud;

---

<sup>2</sup> PP5 was unavailable for this stage.

- Some are generic and are experienced by all cloud adopters (e.g. different payment models), others are specific to the PPs (e.g. obstacles to sale reduced);

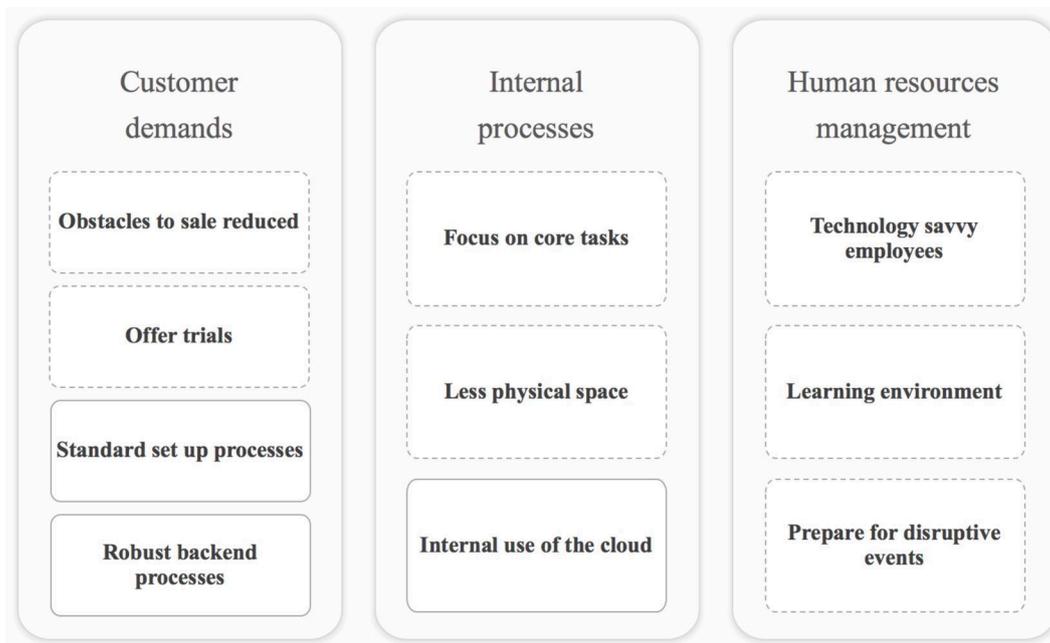


Figure 10 - Second round macro and micro themes from the notes and recordings taken during the interviews. Micro themes that are actions have solid lines, those that are effects have dashed lines.

#### 4.2.1 Customer demands in the cloud

By migrating their software products into the cloud, the PPs gained access to new ways of selling and distributing their products. To exploit the opportunities presented by the new sales and marketing channels, the PPs had to take internal actions. Not all PPs took the same actions and experienced the same effects.

PP1 perceived that the barriers to sales are reduced in the cloud, especially from a customer perspective (*effect*, see also *Obstacles to sale reduced* in Figure 10). In the cloud, it is easier for customers to adopt the product because infrastructure costs have been reduced and the customer's IT department is less involved (i.e. independent implementation). PP1 stated that “many costs for clients disappear with cloud computing”. Furthermore, it has also reduced the overall project times because PP1 does not have to wait for the customers IT department anymore. As a result, they are able to set up new customers in just one day (compared to several weeks before the cloud). PP2 made a similar experience and is also able to set up new customers in one

day. In addition, the PPs started to expand their customer base outside of their core market. PP3, for example, is planning to expand into industries such as mining and pharmaceuticals. PP3 thinks the cloud makes it easier to expand, as they have better ways to analyse the use of the product. They can, for instance, analyse how an engineer uses their product to anticipate which functions a managerial role would need (similar to what PP1 stated in the first stage).

Migrating products into the cloud enables the PPs to offer trial versions of their products, further reducing obstacles to sale (*effect*, see also *Offer trials* in Figure 10). PP2 is making extensive use of trials to win new customers globally. As setting up new customers is easier in the cloud than it was for on-site products, new customers can sign up for the trial by visiting PP2's website. A less personal communication with potential customers has reduced the need to have local offices in the regions of their customers. When new customers want to migrate to the full version of the product PP2 can do it "with the flick of a switch".

Two actions the majority of PPs took to address customer demands stand out. First, the PPs developed a standard process to set up new customers (*action*, see also *Standard set up process* in Figure 10). As new customers can be set up within one day, it puts pressure on internal processes. In order to fulfil the promise to customers, internal business processes need to be efficient and effective (e.g. through a less personal communication with customers as fewer face-to-face meetings are necessary). Increasing the efficiency and effectiveness of internal processes also increased the profitability of the PPs (as they can service a larger number of customers per employee) and reduced their overhead costs (as they only rent the resources they really need and release them afterwards).

Second, the PPs had to build more robust backend processes such as support, negotiating contracts with cloud providers and customers, and leading discussions about data security concerns (*action*, see also *Robust backend processes* in Figure 10). PP1 invested more time into contracts and SLAs with cloud providers and customers. The reason was that PP1 made a negative experience when it came to terminating the contract of one of their customers. The termination rules PP1 had with the customer

were different from the ones PP1 had with the cloud provider. Although the concerns about data security in the cloud become fewer, the PPs still see a need to engage in discussions with some customers. In most of the cases, however, the PPs are able to convince customers that their data is safe in the cloud. In fact, PP1 and PP2 both stated that it is sometimes easier to acquire, in particular, larger customers as they are better able fulfil the list of requirements for data security from the customer's IT departments. This finding is in contrast to what Brender & Markov (2013) found. They concluded that companies that migrate their software products into the cloud experience a loss in IT Governance which would make it harder to acquire larger customers, e.g. due to loss of ISO certifications. The reason for differing results might be that the PPs did not have any means of acquiring ISO certifications before the migration of their software products into the cloud, due to their limited resources. By being in the cloud, and their cloud provider having ISO certifications, they are able to pass the benefits on to their customers.

#### **4.2.2 Internal processes in the cloud**

New opportunities to sell and market the PPs' products influenced the evolution of the companies. Some of the PPs saw differences in their flexibility or innovation ability. The majority of PPs ensured that they keep their vision.

PP1 and PP2 both stated that there are no unique problems related to cloud computing, "it is just general change", as PP2 stated. A major change after migrating their software products into the cloud has been for PP2 the ability of employees to focus on core tasks rather than having to deal with infrastructure issues (*effect*, see also Focus on core tasks in Figure 10, and see Buxmann et al., 2013, for similar findings). The ability to focus on core tasks has had an effect on the flexibility and innovation ability of PP2. For example, employees can pursue more opportunities if they want to and it is easier for the PP to explore new environments, such as mobile computing. The ability to focus on core tasks also had, for PP1, an effect on overall risks threatening the company. If PP1 were to host the products internally, for example, they would need to have people and knowledge to maintain the infrastructure (e.g. ensure backup and recovery). In the cloud, PP1 is paying for these services, where they are no experts in.

Migrating their software products into the cloud enabled the PPs to require less physical space (*effect*, see also Less physical space in Figure 10). The PPs were, for example, able to move their development and testing environments to the cloud. Given that all the PPs are operating in Aberdeen (UK), where renting space is very expensive, moving infrastructure to the cloud saves the PPs money and enhances their ability to expand. Both, the focus on core tasks and requiring less physical space, enable the PPs to use their resources more effectively and in a more targeted way, which increases the number of slack resources that can be used for adapting capabilities, for example.

The use of cloud computing to distribute products to customers encouraged in some cases the internal use of cloud computing (e.g. Office 365, *action*, see also Internal use of the cloud in Figure 10). PP2 uses Office 365 and moved their source code, help desk and phone system to the cloud. Moving secondary processes to the cloud helps the employees of PP2 focus “on tasks and not the system”. As a result, they introduced a flexible working at home policy because through cloud computing everything is accessible everywhere. PP2 stated, however, that they do not know yet if it makes employees more or less productive. They only know that they appear to be happier. PP3 also allows their employees to work from home. Furthermore, by moving other internal processes to the cloud, PP3 was able to achieve an increase in efficiency, mainly by being able to automate tasks that were done manually before.

### **4.2.3 Human resource management in the cloud**

The above descriptions of how cloud computing affects customer demands and the general evolution of the company showed that some of the responsibilities of employees changed. As the PPs try to expand into new industries and continue to explore other technologies (e.g. mobile computing), employees were required to learn new skills. Furthermore, moving internal processes to the cloud also affected everyday tasks of employees (e.g. the ability to work at home). Both have wider effects on human resource management and the PPs adopted different approaches to exploit the opportunities presented.

PP2 noticed that some of their employees are more technologically savvy than others (*effect*, see also Technology savvy employees in Figure 10). PP2 stated “some rush to new technologies, while others stay with the old technology because it works”. PP2 made a conscious decision to create an informal learning environment with a free movement of information. For example, they gave the responsibility to gain knowledge in a particular technology to one employee who was then responsible for permeating the knowledge through the company (e.g. with the help of wikis or meetings to discuss issues together).

The necessity to create a learning environment becomes important as the PPs have less control over the platform their customers use the product on (*action*, see also Learning Environment in Figure 10). By migrating software products into the cloud, customers access it via a web browser. PP3 stated that at work most employees use Internet Explorer. Once they leave work, however, they use all kinds of different browsers. PP3 and PP1 spent a lot of time and effort making their product compatible for different browsers. Furthermore, they are planning to offer their products through different database technologies (e.g. MS SQL and Oracle). Lastly, although infrastructure tasks have been outsourced to the cloud provider, the PPs still need to retain some knowledge about infrastructure, as they are responsible for conceptualising and designing the cloud environment. Offering the products for different browsers, using a range of database technologies and designing cloud infrastructures required the developers of the PPs to learn new skills as they moved from a relatively mature desktop environment (with years of development) towards an immature web environment.

Moving secondary processes like source code management and help desk to the cloud allows employees of the PPs to work from anywhere with an Internet connection (*effect*, see also Prepare for disruptive events in Figure 10). In addition to providing employees with more flexibility, it also has an effect on the overall risk mitigation and the level of resilience of the PPs. In fact, during the course of the multi-stage study PP2 experienced an outage of their Internet connection. The Internet connection was down for an entire day, but the majority of employees were unaffected. They went

home or to a place with Wi-Fi and worked from there. Thus, PP2 suffered neither damage to the business nor a loss of reputation.

### **4.3 Identifying the capabilities**

With the results from the *first* and *second stage* of the multi-stage study, it is possible to identify the capabilities the software vendors adapted while migrating their software products into the cloud. In addition to presenting the results in the form of macro and micro themes, they can also be presented as three tensions and one overall tension that span through the generic product development lifecycle used to structure the interviews of the first stage. The three tensions the PPs are likely to experience during a migration of their software products into the cloud roughly correlate with the stages of the generic product development (see Figure 5 in section 3.3.1).

The macro and micro themes from the first and second stage have been re-analysed with a focus on identifying relationships of the micro themes across the first two stages. Three overall themes emerged (see Figure 11 and Figure 12) that form the basis of the three tensions (see Figure 13). The first overall theme comprises the following micro themes: gain competitive advantages, keep mission of the company, steep learning curve, obstacles to sale removed, offer trials, technology savvy employees and learning environment (see blue boxes in Figure 11 and Figure 12). All micro themes aim to advance the companies goals through cloud computing. In other words, these micro themes deal with decisions around what cloud computing will be adopted for and what the implications are on the business model (a detailed explanation for why each micro theme has been grouped into one of the three groups can be found in Appendix C).

From the above micro themes the first tension the PPs are likely to experience during a migration of their software products into the cloud can be defined (see Tension 1 in Figure 13). The first tension correlates with the plan phase of the generic product development lifecycle and means that the PPs were keen to try the possibilities of cloud computing, for example exploring new ways to market their products or delivering it to customers (i.e. *progression of business model*). At the same time, they also had to listen to their customers and consider any concerns about the new

technology or their willingness to adopt a cloud-based product (i.e. *progression of technology*).

As the PPs are all SMEs they only have a few products (two PPs essentially have only one product they sell). Having only a few products means the PPs do not have the means to experiment with their products. If the revenue of one of their products should break away because customers are unwilling to use it in the cloud the PPs would quickly experience financial distress. Thus, they need to be sure that cloud computing works for their products before committing large resources to the cloud. This could be one reason why, for the majority of PPs, the first step was migrating their software products into the cloud without making any significant changes to it.

Moving existing products had two advantages for the PPs. First, it enabled them to gather data about customer experience before investing more time and money to add cloud specific functionalities or fully exploit cloud advantages. For example, some PPs noticed an increase in softer costs (e.g. because they now have more customers they need better support capabilities). Second, it seems that the PPs and their customers both need time to understand and appreciate cloud computing. The need for time can best be seen by the concerns about data security some customers of PP1 and PP3 had. For PP1's customers (in a later stage of the cloud adoption than PP3) concerns about data security started to disappear after having been in the cloud for a while. It is reasonable to assume that PP3 will make a similar experience once their products are in the cloud, mainly for the reason that the PPs have convincing arguments to ensure that the customer's data is secure in the cloud (see previous section).

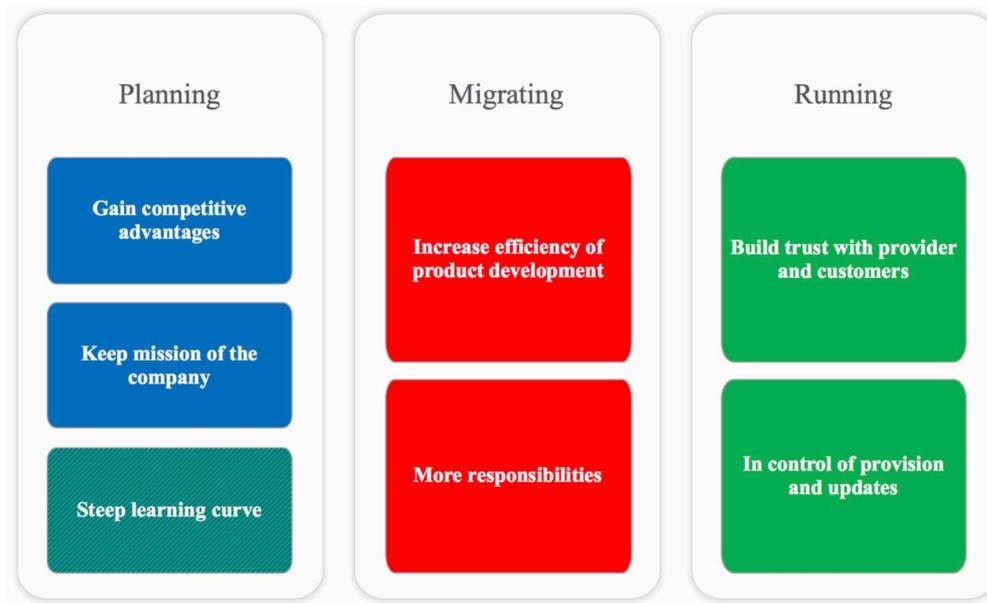


Figure 11 - Revised micro themes of the first stage (Steep learning curve is both blue and green)

The second overall theme comprises the following micro themes: increase efficiency of product development, more responsibilities, standard set up process, robust backend processes, focus on core tasks, and prepare for disruptive events (see red boxes in Figure 11 and Figure 12). All micro themes deal with the initial internal implications of the migration of software products into the cloud. In other words, as cloud computing can be used to extend the customer base and increase the attractiveness of the business model, the PPs are required to change internal business processes in order to be able to provide the product, and additional services, customers require.

From the above micro themes the second tension can be identified (see Tension 2 in Figure 13). The second tension correlates with the migration phase of the generic product development lifecycle and means that the PPs are able to be more efficient and more effective in the cloud by focusing on mission critical tasks (i.e. *lean operations*). At the same time, some of their tasks, mainly feature development, are becoming more difficult due to potential interdependencies and cascading failure events (i.e. *complex operations*).

The tension between lean operations and complex operations is likely to exert a higher pressure on companies during the transition phase (from the on-site product

towards the cloud-based product) in the way the PPs experienced. During this time the PPs had to maintain two environments, for the cloud and non-cloud customers, as not all customers were willing to adopt the cloud-based product right away. The PPs are also just starting to understand how cloud computing works and how internal processes need to be changed in order to make the migration into the cloud successful.

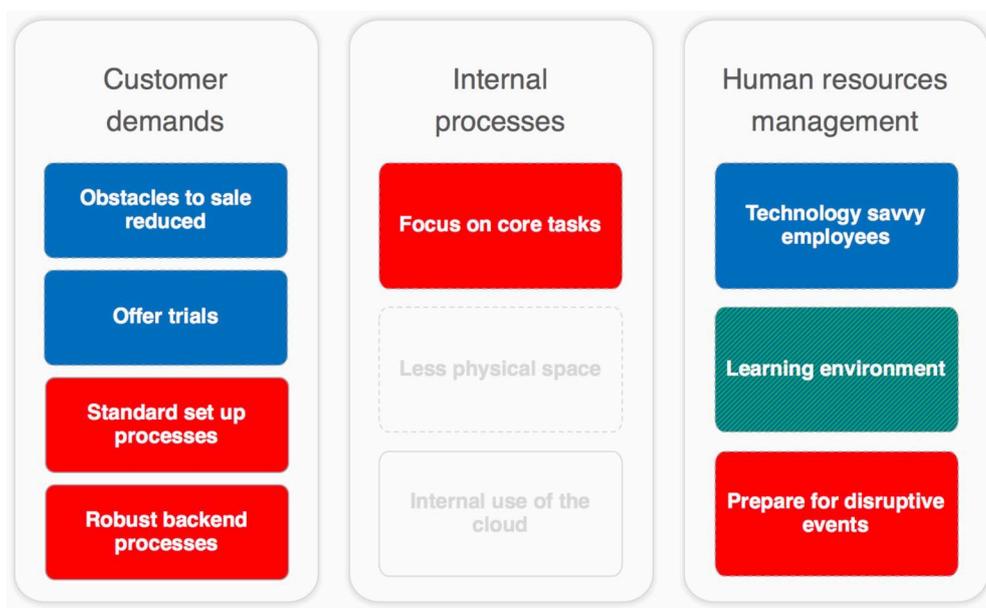


Figure 12 - Revised micro themes of the second stage (Learning environment is blue and green, Less physical space and Internal use of the cloud are excluded because they are not directly related to product development)

The third overall theme comprises the following micro themes: steep learning curve, build trust with providers and customers, in control of provision and updates, and learning environment (see green boxes in Figure 11 and Figure 12). All micro themes reflect the effects the PPs experience once one of their products has been successfully migrated into the cloud and customers started using it. Cloud computing is still new to the PPs and their customers. Thus, many of the micro themes illustrate the need to establish trust between all parties in the cloud environment. The micro themes steep learning curve and learning environment are part of both the first and third overall theme. The two micro themes exemplify the importance of the evolution of the PPs over time and how they use the experiences and findings made to inform future steps, i.e. for the next iteration of the product development cycle.

From the above micro themes the third tension can be defined (see Tension 3 in Figure 13). The third tension correlates with the run phase of the generic product

development lifecycle and means that the PPs and their customers need to establish trust between each other and with the cloud provider. At the same time, the PPs are more in control of their products and how updates are installed, despite outsourcing computing resources and infrastructure to the cloud provider.

The tension between *trust in services* and *control over services* requires the PPs to be aware of the position they are in and the need to balance the relationship between them and the cloud provider and between them and their customers. During the initial migration into the cloud, the relationship between the PPs and the cloud provider is likely to be more important. The PPs main objective is finding the right cloud provider (e.g. niche or established cloud provider) and negotiating a SLA that suits them (e.g. do they have special requests in terms of uptime or do they require additional services). Only if they trust the cloud provider, will they feel more in control of their products despite having outsourced their infrastructure. Once the PPs migrated their software products into the cloud and customers start using them, the relationship between the PP and their customers becomes more important. Now the PPs needed to establish trust between them and their customers (in particular new customers that use the cloud version right away). In addition, the PPs needed to decide on a strategy on how to react to customer requests or changing market requirements. As PP2 described, they had to adopt a more structured approach because they were getting more requests from customers simply because they have more customers. They decided to collect requests first, track which customer requested which feature and only update the product a few times a year. All the other PPs adopted a similar approach and were not interested in updating their products more often just because it is now possible. Part of the reason for this might be the interdependencies PP1 stated, as updates containing bugs will affect every customer instantly.

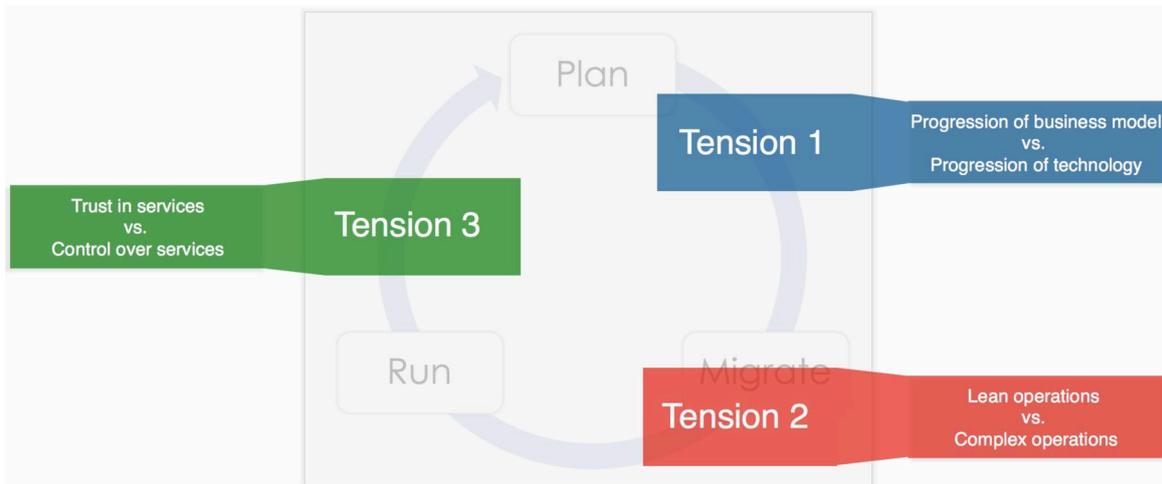


Figure 13 - Three tensions have been overlaid onto the generic product development lifecycle

#### 4.3.1 Overall tension: Managing cloud services vs. managing service delivery

Investigating the micro themes from a systemic perspective by going a level higher in the hierarchy reveals that two aspects influence decisions for the majority of micro themes. First, decisions are influenced by the cloud resources the software vendors adopt, e.g. if they only adopt computing resources from their cloud provider or additional services such as support. Second, decisions are influenced by the products the software vendors develop and additional services they offer, e.g. consulting or support. Depending on the cloud resources and products, the software vendors pursue different paths to exploit the micro themes. Hence, an overall tension is revealed as decisions made regarding the adoption of cloud resources can affect decisions made for the software vendors' products: the software vendors open up new technological possibilities through the migration of their software products into the cloud (as described with the second tensions); at the same time, they also need to ensure that they develop the products and services customers actually need (as described with the first and third tension). Therefore, the overall tension all software vendors faced during the migration was the acquisition of appropriate cloud resources *versus* the delivery of software products.

Investigating the tension in more detail, reveals two capabilities the PPs adapted during the migration of their products into the cloud: *cloud service management* and *service delivery management*. The cloud service management capability addresses one side of the tension: the acquisition of appropriate cloud resources. The capability

is called cloud *service* management because the majority of software vendors get more from their cloud providers than raw computing resources such as help with the migration or technical expertise. The service delivery management capability addresses the other side of the tension: the delivery of software products. The capability is called *service* delivery management because in the cloud the software vendors aim to offer their products as services (i.e. Software as a Service model) and some of them offer more than just their product, such as support or consulting.

The two capabilities the PPs adapted to cloud computing provide examples of capabilities that fit into the definition of dynamic capabilities (see section 2.1). Hence, the multi-stage study advances findings of Winter (2003) and Eisenhardt & Martin (2000) by providing examples of dynamic capabilities that were developed based on change triggered by the adoption of a new technology (not market change as for Winter and Eisenhardt & Martin). The capabilities of the PPs can be considered as dynamic capabilities because their responsibilities and relationships, e.g. with the cloud providers and customers, are not fixed. For the cloud service management capability, for example, it can be, at one end of the spectrum, that cloud services are simply used as computing resources on which the software vendors' products and services are executed. In other words, the software vendor and their cloud provider are loosely coupled. At the other end of the spectrum, it can be that the cloud provider becomes a more integral part of the software vendor to enable new product features. In other words, the software vendor and their cloud provider are more tightly coupled.

For the service delivery management capability similar ends of the spectrum can be identified. The customers of the software vendors can choose how much support and which additional services they want. At the one end of the spectrum, the software vendors provide the basic version of their product. With some software vendors, the customers can sign up online and there is no real interaction with the software vendor. Similar to the above, the software vendor and their customers are loosely coupled. At the other end of the spectrum, the software vendors offer extended customer support and user training, and additional services like help with data migration from the on-site product to the cloud product. Some even offer consulting services to tailor the

product to the customer's business processes. In this case, the software vendor and their customers are more tightly coupled.

The cloud service management capability and service delivery management capability contain tangible elements, like computing resources, and intangible ones, like skills. As everyone can adopt the tangible elements of cloud computing, software vendors will only manage to develop competitive advantages if they integrate the computing resources (and other services provided by the cloud provider) in a way that benefits them, their products and services, and, ultimately, their customers. Subsequently, the relationship between the cloud service management capability and service delivery management capability plays a central role. Both capabilities need to work in sync and towards the same goal in order to have the right cloud services for the right product and services offered to the right customer on the right technology.

The above discussion identified two capabilities the software vendors adapted during the migration of their software products into the cloud. The two capabilities have been identified as being critical for a successful adoption of cloud computing. There are, however, potentially other capabilities the software vendors adapted during the migration of their software products into the cloud, that this study did not identify. In addition, the software vendors are likely to have capabilities that are important for being successful as a software vendor regardless of the use of cloud computing. Hence, the two capabilities described above should be considered a subset of a larger network of capabilities. The two capabilities are potentially coupled to other capabilities within the software vendor.

#### **4.4 Conclusions**

This chapter presented and discussed the results of the first and second stage of the multi-stage study with five SME software vendors that migrated their software products into the cloud. The first stage investigated the effects of cloud computing on product development processes. The second stage investigated the impact of cloud computing on internal processes. The findings provide two major conclusions.

First, the PPs adapted two capabilities to cloud computing that appear to be necessary in order to migrate their software products into the cloud successfully: (1) cloud service management capability, and (2) service delivery management capability. The cloud service management capability is responsible for acquiring and integrating the right cloud resources from the cloud provider. The service delivery management capability is responsible for developing and delivering the services the customers need. Both capabilities need to work in sync in order to have the right cloud resources for the right services and customers.

Second, with the collected data it is not yet possible to identify the underlying processes of adapting capabilities to technological discontinuities. However, this chapter provided a necessary intermediate step towards that goal, as the adapted capabilities need to be identified first to design subsequent stages in a way that allows the identification of the underlying processes of adapting capabilities to cloud computing.

## **5 Emergence and relationships of cloud capabilities**

This chapter has two major objectives. The first major objective is the presentation and discussion of the third and fourth round of the multi-stage study that followed five SME software vendors (PPs) during the migration of their software products into the cloud. The third stage identified major decisions made during the migration of their software products into the cloud (see section 5.1). The fourth stage investigated the relationship of the software vendors with their cloud providers and the effects of this relationship on customer satisfaction (see section 5.2). Based on the results of the multi-stage study, the second major objective of this chapter is the introduction of a framework that captures the underlying processes of adapting capabilities to cloud computing (see section 5.3). The framework aims to answer RQ1: *For SME software vendors migrating their software products into the cloud, what are the underlying processes of adapting core capabilities to cloud computing?*

### **5.1 Major decisions during the adoption process of cloud computing**

The following section investigates how the PPs adopted cloud computing by explaining two decision making approaches. The first approach describes an anticipative approach to cloud computing, where companies see the need to provide customers with a cloud solution in the (near) future. The second approach describes a reactive approach to cloud computing, where companies react to changes in their environment, e.g. new customer demands. Individual decisions that are made during the adoption are largely consonant with traditional decisions related to change management issues. The order and implementation of decisions, however, differs. This can be seen by the fact that the PPs have adopted a hybrid approach that contains some anticipative decisions, and some reactive decisions that emphasize different foci of the adaptation of capabilities, as illustrated below.

#### **5.1.1 Anticipating the use of cloud computing**

The first type of approach describes the situation where companies look into the (near) future and anticipate the use of cloud computing (Figure 14 shows the main decisions involved). Companies anticipate the need to use cloud computing when

trying to expand into new markets or to optimise internal business processes. At the time the company develops the new (cloud-based) services and products, however, its customers may be perfectly happy with the current (non-cloud) services and products that they use.

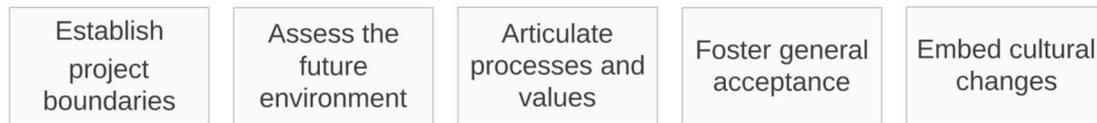


Figure 14 – Anticipative approach to cloud computing

#### 5.1.1.1 *Establish project boundaries*

In order to explore the opportunities that cloud computing presented, the PPs established project boundaries first (see also Christensen & Overdorf 2000). Establishing the project boundaries meant creating a cloud project team that could adapt the capabilities to cloud computing independently. Independence is necessary to ensure that the new cloud project would not be unduly influenced by existing capabilities, processes, and values (chapter 1 explained how existing capabilities can become inappropriate in the cloud).

At PP3, for example, the initial project team consisted of two people who wanted to change Research & Development from being focused on developing new consulting services to developing products that could be sold alongside consulting services. They saw an opportunity in this area: originally they could only grow by employing more consultants, whereas if they had a product, customers could be more inclined to use the product first and then make use of consulting services too, when they identified more general issues. The initial two-man team justified their project to senior management by calculating the potential return on investment. They quickly realised, however, that they had neither the necessary expertise internally nor enough time to develop the idea as it was overshadowed by their normal day-to-day tasks. In the end they decided to employ an intern to help them out, and applied for external funding for a knowledge transfer partnership.

PP4 had a product based on older technology and was exploring cloud computing as a way to update their product. However, they wanted to minimise the amount of extra work they had to do, so they too employed an intern to help them understand cloud computing. Afterwards they hired a developer who was given the responsibility for transforming the existing product into a cloud-based product.

#### *5.1.1.2 Assess the future environment*

Once project boundaries had been established, the PPs had to decide how to assess the future environment (see also Martin & Ching 1999). Assessing the future environment involved identifying the technical factors that could accelerate, slow, or even block the migration of products into the cloud. It also included identifying internal factors that could have impeded the adaptation of capabilities necessary for making the most of cloud computing.

Instead of immediately developing a cloud version of their product PP3 decided to investigate remote desktop as a solution first. Remote hosting appeared reasonable because one of their existing products is very computing intensive, so they thought it might underperform as a web application hosted in the cloud. By having a remote desktop version they could compare the feasibility with a cloud version. Afterwards they asked the intern to develop a simple web application to demonstrate the potential of the cloud. Even though the demo version only included one feature it allowed them to see that cloud computing might indeed not be suitable for their computing intensive product (PP3 still has not found a solution to that problem and is testing different options, such as using dedicated servers from the cloud provider).

PP4 also developed a simple product demo to test the feasibility of a cloud computing solution. Although they decided that the demo version would not be used as a foundation for any future cloud product, it still allowed them to see the potential of cloud computing.

#### *5.1.1.3 Articulate capabilities*

Once the PPs had assessed cloud computing they articulated future processes and values that would form the basis for the adaptation of capabilities (see also Christensen & Overdorf 2000). *Processes* refer to the way employees interact, coordinate, communicate and make decisions and how they turn resources into products and services of greater worth; *values* are the standards by which employees set priorities and decide if an idea is worth pursuing. As the competitive environment changes through a migration into the cloud, the PPs needed to change existing processes and values accordingly, otherwise the adaptation of capabilities could have been impeded.

PP3, for example, focused on creating a SaaS development framework. The framework will be used to develop a product vision, merge existing products, and guide developers that join the company. As part of this framework PP3 decided to integrate three existing tools into one cloud-based product. The company also decided to have weekly meetings with management to plan future products and track development. The weekly meetings incorporate workshops to develop new features. They intend to introduce more products, one at a time, at a later stage.

PP4 offers a product that is not mission critical, so they had to develop their cloud product in a way that would create the minimum amount of extra work for their existing customers. To achieve that, they offer the option to rent their software instead of buying it. PP4 also started to focus on the development of a network of dealers and partners to distribute their product more efficiently.

#### *5.1.1.4 Foster general acceptance*

After the PPs decided how they would adapt their capabilities, they needed to foster general acceptance for the migration of software products into the cloud (see also Leonard-Barton & Kraus 1985). Each of groups affected by cloud computing had to be considered from a different perspective and addressed accordingly. For example, management is likely to be concerned with return on investment or long-term objectives, whereas developers are more concerned with their skills and product vision.

PP3 decided to build a more sophisticated demo at this point for two reasons. First, they wanted customers to provide feedback on their demo. Second, they wanted to address the technical issues associated with the cloud, such as scalability, multi-tenancy, integrity, and security. Together they provide evidence to convince senior management and developers that cloud computing can constitute the future of their company. Senior management, for example, wanted to see a working cloud version of their product. After that senior management became a part of the project team to refine and finalise the product vision, develop user stories and plan the development of features. They realised, however, that it is difficult to identify features that both have a business value and are important for the future of the product. For this reason they extended the project team to include 4-5 customers that have agreed to test beta versions. The beta customers will get a discount on the final product in return.

#### *5.1.1.5 Embed capabilities*

The last decision made by the PPs ensured that the adapted capabilities and cultural changes made by the project team were embedded into the company. Embedding the capabilities meant transferring the adapted capabilities from people to processes and finally to organisational culture. Transferring them from people to the organisational culture was important as it ensured that the use of cloud computing within the company is independent of individual people.

In order to embed cloud computing into the culture PP3 decided to adopt a SCRUM approach with 3 monthly development cycles. After the first cycle they will decide if they do another cycle or if the product is mature enough to be tested with customers. Once the first product is mature enough they will do other activities for 6 months before starting another one-year development cycle. After the one-year development cycle they expect to have a product that is closer to what the user wants and that can be tested with customers to gather feedback for further refinement before the product is released. They are starting with products they feel will generate the largest return on investment.

### 5.1.2 Reacting to cloud computing demands

Reacting to demands means that companies migrate their software products into the cloud after a customer approaches them or after a competitor decided to migrate their software products into the cloud (see Figure 15 for the decisions involved). When a customer approaches the company it does not mean that they are demanding a cloud solution. It is often the case that the current way of distributing the software is not satisfactory to the customers. PP1, for example, moved to the cloud because it took too long to install their software on-site as the customers' IT department had to approve the process first, which sometimes took too long. In other cases the customers simply could not install the software because they did not have an IT department or their policies would not allow it. In both cases, customer demands and requirements were the primary force for a migration into the cloud.



Figure 15 – Reactive approach to cloud computing

#### 5.1.2.1 Set context for change

When reacting to cloud computing demands the PPs had to set the context for change because everyone in the company was affected by cloud computing immediately. They defined the business problem that lay behind their reasons for migrating their software products into the cloud (see also Duck 1993). They also needed to prepare their employees, understand what they knew (and did not know) about cloud computing, give them feedback about the progress, and establish a continuing dialogue with them.

A customer outside of PP5's main market approached them to ask if they could offer a cloud solution similar to that of a competitor. PP5 already had an 85% market share in their core market and knew that companies in that market prefer to keep data on-site. Together, these lines of reasoning led PP5 to look at other markets they could expand into.

PP5's experience was not unique. PP1's customers also preferred to keep data on-site but PP1 needed to find a different way of providing their products because on-site installation took too long. PP1 was often constrained by their customer's IT department which had to approve the installation of the product. The constraint led the project partner to adopt a hybrid approach, initially remote hosting the product on their own site, and moving it to the customers' site once the IT department had approved the on-site installation. Thus, the customer could use the product immediately and PP1 was no longer constrained by the customers' IT department. In order to establish the hybrid approach, PP1 found that the only changes they had to make to the software were related to security, performance, and some configuration parameters. Afterwards they tested the hybrid solution with a local customer for one year before using it with other customers.

#### *5.1.2.2 Allow short term gains*

After defining the business problem and preparing the company, the PPs needed to decide how they could allow short-term gains. Allowing short-term gains was an important step affecting customers and employees equally. Making the effects visible quickly helped them to conclude whether cloud computing works for them (see also Duck 1993). The customers ended up with a solution that stopped them from moving to a competitor. With the commitment to produce short-term gains employees were forced to think analytically and clarify the vision about cloud computing (see also Kotter 2009). It was also necessary to exploit existing capabilities to the largest possible extent before the PPs could think about adapting them to make them more suitable to the cloud.

Two project partners, PP1 and PP2, achieved short-term gains by hosting existing products internally on their own servers. For the customer it did not really matter where the product was hosted. For the PPs, however, this decision was important. From a strategic perspective they were able to test a new distribution channel and how their customers would react to the cloud without committing and investing too much time and money. From an operational perspective they were able to see how the product needs to be changed and how capabilities need to be adapted before going

further. Both PPs, however, were aware that this could only be a short-term solution because they could not provide the levels of scalability, performance and resilience of cloud providers.

PP5 initially tried to develop a private cloud but quickly ran into various constraints, which forced them to use a public cloud. By migrating their existing software products into the cloud and changing the pricing model they were able to satisfy short-term customer requirements. PP1, PP2 and PP5 already had solutions in place that would allow them to host the products on behalf of their customers, e.g. if they did not have their own IT department. With the emergence of cloud computing they started selling this hosted version as a cloud product. In other words, customers who had previously bought the whole product were now able to rent the software, and pay using transaction based pricing or a subscription e.g. per month.

#### *5.1.2.3 Align capabilities to new approach*

After creating short-term gains and satisfying customer requirements the PPs needed to monitor existing capabilities and align them to the new approach (see also Christensen & Overdorf 2000). At this point the majority of PPs reviewed their approach to the cloud and, if necessary, decided how they would redevelop their products to make them more suitable for the cloud, how secondary functions such as customer support would need to evolve and how capabilities would need to be adapted. As cloud computing can make existing capabilities obsolete, reviewing the approach to cloud computing was essential at this point.

PP1 and PP2 decided that it made more sense to move their products to an external hosting provider. PP1 kept running out of internal capacity with every customer they added. Because they only hosted the product until the customers' IT department was ready to install the product on-site it did not make sense to keep buying more hardware without receiving any return. By acquiring global customers, PP1 requires 24/7 support, which the company did not want to invest in. The external hosting provider they chose offered to provide the necessary support in addition to other secondary functions, like technical expertise the company did not have, in areas such as Citrix.

PP2 decided to migrate their software products into the cloud for similar technical reasons. Initially they were hosting the products in their own data centre. With an increasing number of customers they realised that internal hosting could not be the long-term solution for two reasons. First, they felt the need to increase the resilience of the products so that they could provide the service the customers were expecting (e.g. high availability). Second, PP2's developers wanted to concentrate on solving business problems rather than looking after physical hardware.

PP5 decided at this stage to redevelop their product to make it, as the Managing Director expressed it, a "true cloud product" with transaction based billing. In order to find out how they would need to change their products, they did a gap analysis of their core market with potential markets to expand into. The results allowed them to make three decisions. First, they decided to build a series of native tablet and smartphone applications. Second, they decided how to commercialise their products in other markets. Third, they decided that they had to redevelop their database structure. The redevelopment of their product is an on-going continuous process, although the product is mature enough to be sold.

#### *5.1.2.4 Institutionalise new approaches*

The fourth decision the PPs had to make was about how to institutionalise the new approaches (see also Kotter 2009). Before this decision, many PPs explored a wide range of possibilities for using the cloud. PP2 and PP5, for example, considered developing a private cloud and PP1 tried migrating existing databases to the cloud. Over time, the PPs developed a better understanding of how the cloud works, and used their experiences to develop a strategy that they believe will be successful.

PP2 developed a scalable architecture in 2001 that allowed them to gradually scale up bandwidth and capacity as they needed to. They separated their design into three parts (load balancer, application servers, and databases) and they could scale up any part. They had had to develop a scalable architecture themselves because they were procuring their own hardware and placing it with an external hosting provider at a data centre (in 2001 cloud computing and virtual machines were not commercially

available yet). PP2 used this approach until recently; they are now using virtual machines because the physical servers were getting old. If they had to go through the same development process again, however, they would adopt a virtual environment right away and move to a cloud provider.

PP5 tested their redesigned cloud products with one customer. Testing the new product helped them to revise their pricing models. They found out that the existing pricing model was too complicated and that two different models would be sufficient. The new pricing models are both transaction based but one includes additional services such as an upfront analysis of the customers' environment, support for the implementation, and training for employees. When PP5 started to migrate their product into the cloud, however, they anticipated that customers would only demand the pricing model without the additional services. One reason for having these pricing models was the need to be different to competitors.

#### *5.1.2.5 Embed capabilities*

The last decision the PPs had to make while migrating their software products into the cloud was how they would embed the capabilities and cultural changes that arose out of using cloud computing. At this point in time companies had to decide how they would transfer the adapted capabilities from people to processes and finally to culture to make cloud computing independent of individual people (similar to the anticipative approach described above).

PP5 initiated cultural changes by creating manuals and user guides for the cloud version. More extensive hands-off guidance is necessary as the relationships with customers change. In a cloud environment face-to-face meetings with customers are no longer necessary. Instead companies engage in a continuous dialogue with customers. The changes in the relationship with customers also affect the way new features are developed. Whereas before the cloud they were trying to sell existing products and features, in the cloud they try to anticipate features customers will demand in the future. The aim to anticipate feature requests required an internal cultural shift because the work of developers would not immediately provide a return

on investment. The changes at other PPs were similar, although not always to such a large extent.

### 5.1.3 Switching between anticipating and reacting

In reality none of the PPs rigorously followed either of the two approaches described above. Instead, companies combined various elements of the approaches. For example, a company might start off taking an anticipatory approach but then gradually move to a more reactive approach. In other words, the two approaches are more like the ends of a continuum (see Figure 16). The five PPs have been plotted on this decision continuum to give some indication of the approaches they have followed, and how far they have proceeded. In the following, it will be illustrated how PP3, PP5, and PP2 fit into the decision making continuum.

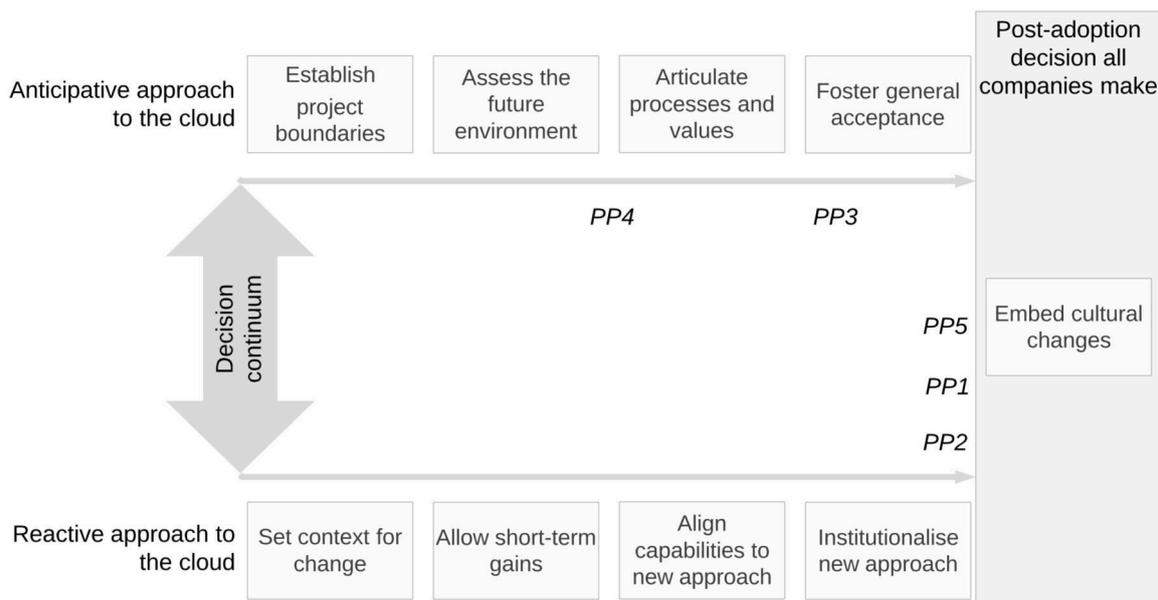


Figure 16 - Decision continuum between the two decision making approaches

PP3 is mostly aligned with the anticipative approach. They have proceeded through all the decisions until Foster general acceptance, which is where they are right now. They established a project team that initially consisted of two people and gradually grew as the company followed the steps in this approach. PP3 created software demos to assess the future environment and foster general acceptance. In addition, they developed a software development framework to articulate processes and values and

guide product development in the cloud. Currently, they are planning how to embed cultural changes, such as using SCRUM for product development.

PP5 are not very strongly aligned with either particular approach. For example, they decided to completely redevelop their products and create native smartphone and tablet apps very early in the adoption of cloud computing; a decision more common among companies that follow the anticipatory approach. On the other hand, PP5 allowed short-term gains by migrating their existing software products into the cloud before redeveloping them; a decision more common among companies that follow the reactive approach.

PP2 are most strongly aligned with the reactive approach. They mobilised the entire company to move into the cloud and allowed short-term gains by migrating their existing software products into the cloud. PP2 offered remotely hosted products to their customers before virtual machines or cloud computing were commercially available. They had to develop their own scalable architecture in order to align tasks to the new approach and institutionalise them. PP2 used this scalable architecture for more than 10 years. Currently they are moving from a physical infrastructure to a virtual one in order to respond to cultural factors because developers want to focus on solving business problems instead of looking after hardware.

Figure 16 shows that the decisions during the adoption of cloud computing can differ between companies. The data suggests that the PPs follow a flexible approach, which allows them to switch between elements of the anticipative and reactive approach. After adoption—when at least one product has been successfully migrated into the cloud—both approaches consider how to embed the adapted capabilities into an organisational culture that facilitates and supports the use of cloud computing. How long this post-adoption step will take depends on the company and the number of products that are migrated to the cloud. PP2, for example, has several products of which only a few are in the cloud. Still having on-site products means PP2 has to retain organisational processes that existed before the migration into the cloud. PP5 has only one product, which was successfully migrated to the cloud. They can optimise all their processes for the cloud.

#### **5.1.4 Different priorities for the adaptation of capabilities**

The *third stage*, by investigating major decisions made during the migration into the cloud, reveals different priorities in the adaptation of the two capabilities: cloud service management and service delivery management. The different priorities stem from previous capabilities and knowledge the PPs possess, as explained in the following.

When following an anticipative approach, the software vendors place greater emphasis on adapting the cloud service management capability first, before adapting the service delivery management capability. They want to explore the technical capabilities of cloud computing and how that can affect the development of new products and services. As the software vendors had no prior experience in hosting a product remotely, they had to focus on understanding the technology first. The adaptation of technical capabilities had to be informed through a Greenfield approach (Hopkins & Jenkins 2008), as any existing technical capabilities were rendered obsolete by cloud computing. The software vendors achieved this by developing demos that tested the technology, not the product. PP3, for example, first developed a simple demo to test the basic elements of cloud computing, e.g. database performance. Gradually they expanded that demo to see the effects of security in the cloud, multi-tenancy, automation etc. In addition, PP3 received outside help from consultants, and corporations with knowledge transfer networks and universities. Once the software vendors understood how the technology worked, they could focus on adapting the other capability necessary to be successful in the cloud: service delivery management. To adapt this capability, the PPs made extensive use of demos to collect customer feedback, i.e. demos to test the product. PP3, for example, worked closely with three of their customers. The customers were also part of review meetings at PP3's offices. They will receive discounts on the final version of the cloud product to compensate for their efforts. Developing the necessary technical knowledge and evaluating the impact cloud computing can have on customers takes a long time. All software vendors that followed the anticipative approach required much more time to migrate their software products into the cloud than the ones that followed the reactive approach. In some cases, it took them three times as long (PP1

required 9 months to complete the migration, PP3 is in the migration process for 3 years). The anticipative software vendors were much more cautious in the migration of their software products into the cloud and started very small, 1-2 people in the project team, and gradually expanded the project team (to include managers) and the features of the cloud-based version of their product.

When following a reactive approach, the software vendors placed greater emphasis on adapting the service delivery management capability. The reactive software vendors had prior experiences in remote hosting their product. Although neither of the software vendors had prior knowledge of cloud computing, they were able to adapt their existing capabilities in a way that suited cloud computing. Furthermore, the software vendors committed to cloud computing faster than the ones that followed the anticipative approach. The faster commitment was another reason for them to exploit existing capabilities to the greatest possible extent, as everyone in the organisation was immediately affected by cloud computing. Yet, the software vendors also had to extend their capabilities, as seen by the sudden need to increase support, which surprised all software vendors. The software vendors were able to focus more time and effort on extending their capabilities required by cloud computing by migrating their existing software products into the cloud without changing them significantly. In contrast, the software vendors that followed the anticipative approach redeveloped their products to make them more suitable for the cloud. Although the software vendors that followed the reactive approach are, therefore, not able to immediately benefit from all the technical advantages of cloud computing, it gives them more slack resources to focus on adapting their capabilities. Furthermore, in order to adapt necessary capabilities to cloud computing, the software vendors were supported by their cloud providers. All software vendors that followed the reactive approach used niche providers (except PP5), which were able to tailor their services to the particular needs of the individual software vendor. For PP1, for example, the cloud provider has taken over customer support because they are able to offer 24/7 support, for which the software vendor has not the resources.

## **5.2 Influence of the cloud provider on customer satisfaction**

The fourth stage of the multi-stage study was designed to capture the various factors that influenced the decisions around the PPs' cloud providers, and how this has changed over time. The results of the analyses below are shown using influence diagrams<sup>3</sup>. The influence diagram is a method used to investigate situations where many different bodies, groups, and individuals influence the outcome of a situation. Influence diagrams consist of blobs and arrows. Blobs represent entities that influence each other. Arrows show the flow of influence between the blobs.

Blobs in influence diagrams can represent any component of a system such as groups or individuals but also things like culture, mood, or salary levels. Influence diagrams represent a snapshot of a situation at a point in time. They do not show how situations change over time nor do they show the sequence of events or processes.

Figure 17 shows an example of a simple influence diagram. It has two blobs (*Working Conditions* and *Staff Morale*) and one arrow of influence: working conditions influence staff morale.

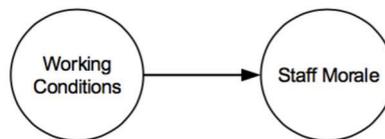


Figure 17 - Simple influence diagram to show that working conditions influence staff morale

With the collected data three aspects associated with choosing a cloud provider could be identified. First was the choice of the type of cloud provider to use. Second was the achievement and maintenance of customer satisfaction once the products were being used in the cloud. Third was the consideration whether to use multiple cloud providers to expand the customer base. They are described in the following.

### 5.2.1 Choosing the type of cloud provider to use

---

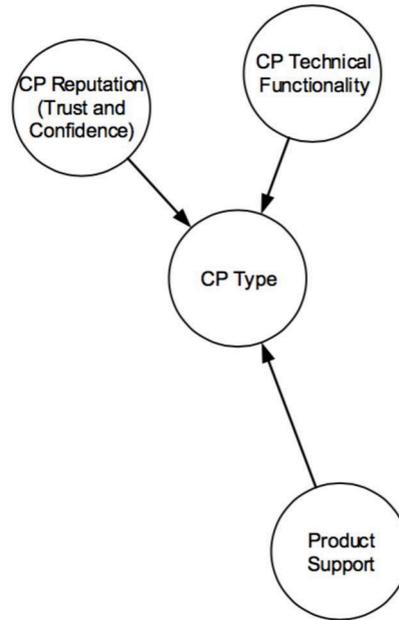
<sup>3</sup> PP5 was unavailable for this stage

Figure 18 shows the influence diagram for the initial migration into the cloud. At this point in time it was most important for the PPs to choose an appropriate cloud provider that could help them migrate their products to the cloud.

The PPs developed a shopping list of requirements that candidate cloud providers had to satisfy (*CP Technical Functionality* and *CP Reputation* in Figure 18). Most of the PPs had a very specific idea of what they needed from their potential cloud provider. These PPs only evaluated 2-3 cloud providers. The cloud providers were a mix of niche and major providers (*CP Type* in Figure 18). In general, the PPs preferred flexible cloud providers that could satisfy their own particular needs and sometimes offer services over and above the standard computing resources (virtual machines, storage, databases and network capabilities).

The following requirements were most commonly mentioned by the PPs (in no particular order):

- Uptime, although the majority of PPs expressed that their applications are not mission critical and customers would not suffer any losses if the applications would be down for a few minutes or hours
- Monitoring capabilities, e.g. measure performance or costs
- Transparent cost structure, with no hidden costs
- Confidence that their data is secure with the cloud provider
- Single point of contact that can take care of any special needs



*Figure 18 - Influence diagram showing how the project partners chose the type of cloud provider to use (CP stands for cloud provider).*

None of the PPs had detailed knowledge of cloud computing when they first looked at choosing a cloud provider. They had to evaluate candidate cloud providers by asking questions (e.g. about availability), and trust that the information given was correct. Some of the PPs assessed the dependability of the candidate cloud providers using free and demo services. The section below explains how trust and confidence in the cloud provider (CP Reputation) and the technical functionality (CP Technical Functionality) influenced the type of cloud provider (CP Type) to use.

PP2 considered a local cloud provider, as well as Amazon and Microsoft. One of their major aims was to move their IT infrastructure into the cloud without fundamentally changing the commercial aspects of their products. They already had a long standing relationship with a local cloud provider who also acted as their ISP (Internet Service Provider). PP2 felt that the local provider's cloud offering was easy to understand with a transparent cost structure. In contrast, Amazon could not provide accurate answers to the technical questions that emerged when testing the Amazon cloud. Because of the long relationship with the local provider PP2 already had an established single point of contact and physical access to the provider's facilities. PP2 felt that the local provider was more flexible which gave them "more confidence that they were the one". PP2 is currently using the local provider as their main cloud

provider and Amazon for one small project to see how Amazon continues to develop their cloud services.

PP3 is still in the process of choosing a cloud provider and currently considering a local provider and Amazon. Their main reason for using Amazon is that they provide data centres in those geographical areas where PP3 operates. For PP3's customers it is important that the data remains within their country. PP3 feels that the backend services Amazon currently offers are not sufficient to meet their needs, so they are considering a hybrid approach to compensate for this, using Amazon to deploy their products and other (local) providers for data storage, archive and other services like support.

PP4 expressed that they did not want to rely solely on the information given to them by the cloud providers. In order for them to seriously consider a particular cloud provider they would have to "come with good recommendations", e.g. from business partners. PP4 wants to get more from the cloud provider than just computing resources. They see the relationship as being more like a partnership where the cloud provider helps them, but they can also help the cloud provider.

Only PP1 evaluated a large number of cloud providers. They interviewed six different cloud providers of which only one suitably met their needs. Although the chosen provider did not provide everything they needed at the time, they were sufficiently flexible to work with PP1 to develop the missing features in the long term. PP1 stated, "no one else was flexible enough".

Once companies start to migrate some of their products into the cloud the influence CP Type has on Product Support gains importance. Some PPs required more than just computing resources from their provider. Niche providers often helped the PPs to migrate their products into the cloud or provided other services like first line support for customers or managing the cloud environment for the PPs (similar to what Brender & Markov 2013 found out for Swiss companies, although in their context the niche providers helped with cloud security questions). The PPs also have a single point of contact with the cloud provider, which means that end-user problems or

requests can be passed on from the end-user to the PP and finally to the cloud provider quicker. With major cloud providers, like Amazon or Microsoft, the PPs do not believe that they would get the same level of personalised attention from the cloud provider because the PPs are comparatively small customers.

The majority of PPs do not offer an off-the-shelf product solution. They have to bid for new projects. Some of the new customers may require additional functionality over and above that offered by the PP's products. Niche providers can help the PPs to develop and provide additional end-user requirements. PP1, for example, was about to acquire a new customer who mandated the use of Citrix, an area in which the PP lacked expertise. Their cloud provider was able to provide the necessary expertise so that they were able to acquire the new customer. Now PP1 offers Citrix as part of its package to all customers.

### **5.2.2 Achieving and maintaining customer satisfaction**

As the PPs migrated their products into the cloud and customers started using them, the cloud-related requirements of the PPs have evolved to reflect their experiences (see Figure 19). Once the PPs gained confidence in the new technology their focus shifted to how cloud computing can help them achieve higher levels of customer satisfaction. Based on the data collected, two areas have a significant influence on customer satisfaction: data location and product functionality. Each of these are considered, in turn, below.

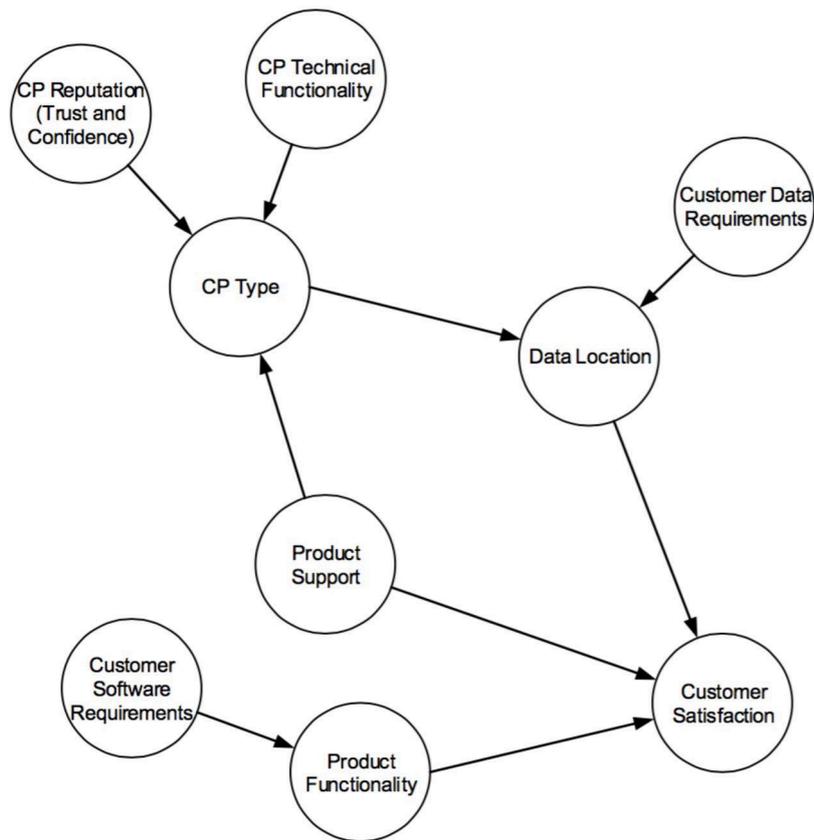


Figure 19 - Influence diagram showing how the project partners achieved and maintained customer satisfaction in the cloud (CP stands for cloud provider).

The decision about where data should be stored is sometimes influenced by the choice of provider (see relationship between CP Type and Data Location in Figure 19). When PPs prefer to work with a niche provider they know where the provider is located and that their data centres are co-located. The decision about where data is stored is then linked to the choice of niche provider. When PPs decide to work with a major cloud provider the decision about data storage can often be made separately as all of the major cloud providers operate data centres in several countries across the world.

Whilst it is generally not important for the PPs where data is stored, it is important for some of their customers (see relationship between Customer Data Requirements and Data Location in Figure 19). In general, the larger the customer, the more important it is for them where data is stored, and the more questions they ask about the cloud provider. This finding is similar to what Alshamaila et al. (2013) found out for cloud computing and customer data.

PP1 explained that most of their customers (and potential customers) only ask questions related to security, uptime, and disaster recovery. Some are more interested in the provider's reputation and make comments such as, "we never heard of these guys" or ask questions like "Why are you using these guys?" as PP1 stated, however, their customers cannot really object "as their contract is with us". From PP1's experiences, their customers are given a list of requirements and constraints from their own IT department and their main issue is that "we as a software vendor tick all the boxes they are concerned about".

PP2 reported a similar experience. Their customers also asked only in very general terms about the encryption of the data, security audits, etc. PP2, however, was "surprised about how little they care about the cloud provider". If a new customer asked for their data to be stored in a country outside the UK, where PP2's current cloud provider is not located, they would simply use an additional provider outside the UK.

PP3 explained that their products contain sensitive customer data that customers want to remain in their own country. As a result they might have to work with multiple providers in different countries increasing the overhead costs, as it is more difficult to manage such an environment. Instead of working with multiple cloud providers, the PP is considering three alternatives.

First, they are thinking about a hybrid approach where they run the application with Amazon, for example, but store the customer data with local cloud providers. PP3 would then have one version of the product that needs to be maintained. In addition, they are considering a push-approach for customer data which means that the customer makes sure that the cloud application gets the information it needs (the alternative would be a pull-approach where the cloud application accesses the customers' network to collect the data).

Second, PP3 might follow a remote hosting approach rather than a pure cloud computing approach. Their sister company followed a remote hosting approach by

renting a dedicated server from a cloud provider. PP3 would ensure that only they as a software vendor have access to those servers and that they are not shared among customers (i.e. avoiding multi-tenancy).

The third approach is to avoid offering the cloud version to some customers, such as those in emerging markets where there are no local cloud providers available. PP3 is currently evaluating all of these options. First, they will develop strategies on how to approach customers about the cloud version and see how two companies, which agreed to test their demo, react to the different approaches. They believe it is more important to approach customers directly instead of spending money on advertising, because it gives them “the chance to remove fears of the cloud” at the earliest opportunity.

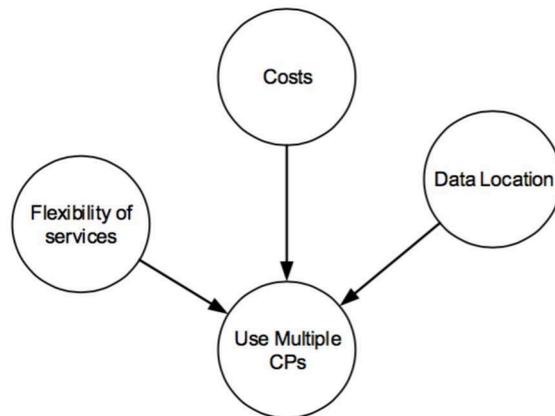
The approaches taken by the PPs to increase product functionality to maintain and achieve customer satisfaction differ significantly (see relationship between Product Functionality and Customer Satisfaction in Figure 19). PP2 developed a control panel where they can enable and disable features of their cloud products. The control panel allows them to develop and enable features that have been requested by only a few customers (Buxmann et al., 2013, make a similar suggestion). If a new feature is deemed to be successful they can potentially enable it for all users. PP1 takes a similar approach. Their customers can pay for the development of new features. In the initial stage only the customer who requested the feature are able to use it. After a while it is potentially rolled out to every customer in the next release of the product.

### **5.2.3 Expanding the customer base**

The data suggests that the PPs are considering and, in some cases, already use more than one cloud provider, as this offers a way of broadening their customer base. It allows them to offer a wider range of services, possibly at a lower price (see Figure 20). It also enables the customers to have a say in where data should be physically stored.

All PPs initially intended to work with one cloud provider. Over time, however, three of the four PPs interviewed decided to consider using multiple providers. PP1 and

PP2 started working with a niche provider and are now thinking about adding Amazon as a major provider to their portfolio. They say Amazon is becoming more flexible and offers a competitive price. PP1 and PP2 are not planning to replace their current provider. They are thinking about using Amazon alongside them.



*Figure 20 - Influence diagram showing how some project partners use multiple cloud providers to expand their customer base.*

PP2 evaluated Amazon as a potential cloud provider when initially planning the migration of their software products into the cloud but concluded that they could not offer what they needed at the time. As the cloud computing market is very dynamic and fast moving they decided to keep evaluating Amazon as an option for the future. They currently use Amazon for a small project, “partly out of curiosity to see how Amazon works”. PP2 is concerned about the fact that their current cloud provider has only one data centre. The cloud provider has decided to build another data centre but it could take several years until it is in operation. PP2 is therefore thinking about using Amazon as a backup cloud provider until the second data centre is ready. They are also considering the option of storing some servers in their own company in case the cloud provider breaks down.

PP1 also evaluated Amazon and Google when they planned the migration of their software products into the cloud but concluded that the one size fits all approach of these providers would not have been suitable enough for them. They too chose a local provider and are currently thinking about using Amazon because of the increased flexibility in the services they offer and their competitive pricing. They plan to carry out an evaluation soon by moving some of their instances into Amazon’s cloud. The

aim is not to replace their current cloud provider but to have alternatives. If Amazon proves to be a good alternative they might put new customers on Amazon and use both cloud providers, at least in the short term.

For PP3 the situation appears to be different. They are still in the stage of migrating their software products into the cloud and do not have any customers productively using their products in the cloud. They expect that customers will care where the data is physically stored, which databases they use and that these are not shared among customers (see relationship between Data Location and Multiple CPs in Figure 20). For a global company, with customers across the world, this can make the migration of software products into the cloud particularly challenging (see first stage results for approaches PP3 considers to resolves these issues).

#### **5.2.4 Capabilities in a network**

The *fourth stage* investigated the relationship of the software vendors with their cloud provider(s) and the wider impacts of this relationship on the PPs' customers. The following section explains how the two capabilities, cloud service management and service delivery management, are coupled with the cloud provider and the PPs' customers. It shows how the capabilities behave in a network, not only during times of technological change but also in everyday activities. Afuah (2000) and Spedale (2003) concluded that, in times of technological change, capabilities are affected by co-opetitors (suppliers, customers, competitors, etc.). The following discussion extends the findings of Afuah and Spedale by explaining how the capabilities of software vendors that migrate their software products into the cloud are affected by the software vendors' cloud providers and their own customers. This section identified three issues as being of fundamental importance to the relationships with cloud providers and customers:

1. The additional service functionality enabled by the cloud service provider.
2. The development of product functionality based on customer preferences.
3. The location of customer data.

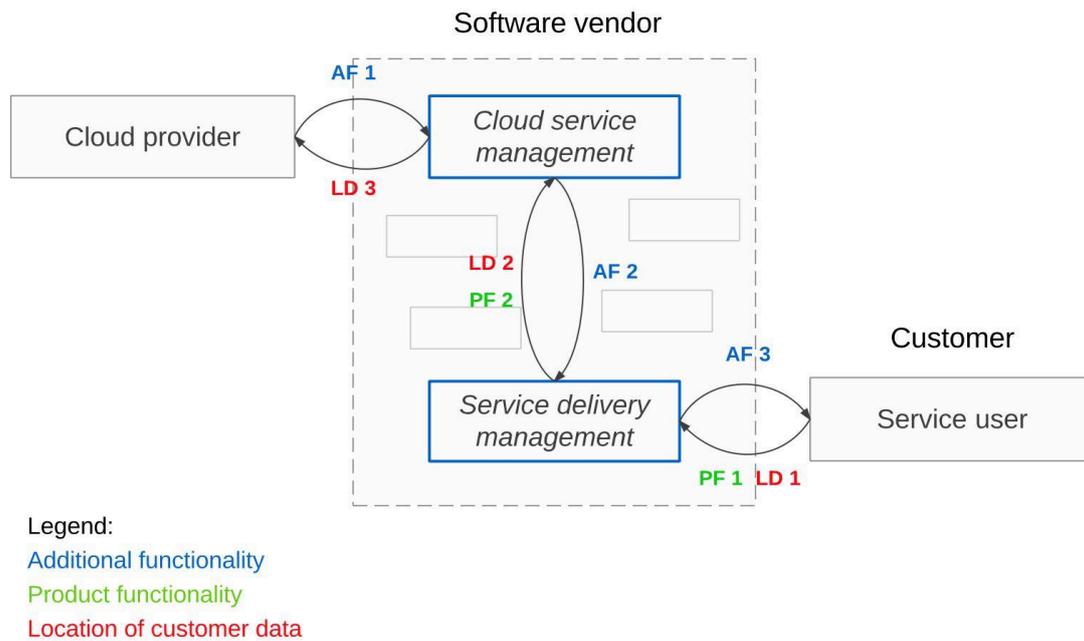


Figure 21 – Model to illustrate the tight coupling of the capabilities with each other and the cloud provider and customer

The provision of additional service functionality (to the customer) depends first on the services the cloud provider is providing to the cloud service management capability (AF 1 in Figure 21); second on how the cloud service management capability integrates the additional functionality into their products (AF 2 in Figure 21); and, finally on how the additional functionality is perceived by the customers (AF 3 in Figure 21).

The development of product functionality depends first on the demands the service user is putting on the service delivery management capability (PF 1 in Figure 21); and, second on how the service delivery management capability is able to use existing cloud services provided by the cloud service management capability or on how the two capabilities are able to work in sync to acquire appropriate new services and integrate them accordingly (PF 2 in Figure 21).

The location of customer data depends first on the preferences the service user is formulating to the service delivery management capability (LD 1 in Figure 21); second on the communication between the service delivery management capability and the cloud service management capability so that the cloud service management capability knows which cloud services to acquire (LD 2 in Figure 21); and, third on

the relationship between the cloud service management capability and the cloud service provider to acquire the right services and resources from the right cloud service provider (LD 3 in Figure 21).

The PPs have to consider the cloud provider and their own customers when adapting their capabilities to cloud computing as exemplified above. Furthermore, the cloud provider and customers can also actively influence the adaptation of capabilities. For example, PP1 started developing a tablet version of their product until a potential customer requested a specific feature. PP1 had to suspend the development of the tablet version in order to concentrate their resources on acquiring the new customer.

The above discussion proves that actions taken by cloud providers and customers can have effects on software vendors. Software vendors are then required to react to or anticipate the actions by cloud providers and customers. The software vendor, cloud provider, and customer are, however, tightly coupled as the provision of additional functionality in Figure 21 shows: AF1, AF2, and AF3 span through the entire model. Thus, if the cloud provider and customer should take opposing actions at the same time, the software vendor needs to decide which party to satisfy. In some cases, however, the software vendor might be forced to satisfy the cloud provider, for example, if the cloud provider discontinues a service the software vendor has been relying on (e.g. a type of database). Hence, software vendors should have decision making procedures in place, in case they are not able to satisfy both, the cloud provider and their own customers. To develop decision making procedures it is necessary to carry out further studies, to understand how software vendors behave in these kinds of situations.

### **5.3 The underlying processes of adapting capabilities**

The goal of the multi-stage study was the identification of the underlying processes of adapting capabilities to cloud computing. As the term ‘underlying processes’ is very broad, chapter 1 defined it more specifically as: identifying the technical *and* organisational challenges that influence the adaptation of capabilities, identifying the steps companies take to adapt capabilities and in what order they take these steps, and identifying the reasons for taking these steps. The four stages of the multi-stage study

addressed all three aspects that make up the term ‘underlying processes’. In the following, the results from all four stages will be incorporated into a framework that captures the underlying processes of adapting capabilities to cloud computing.

### **5.3.1 Four viewpoints influence the adaptation of capabilities to cloud computing**

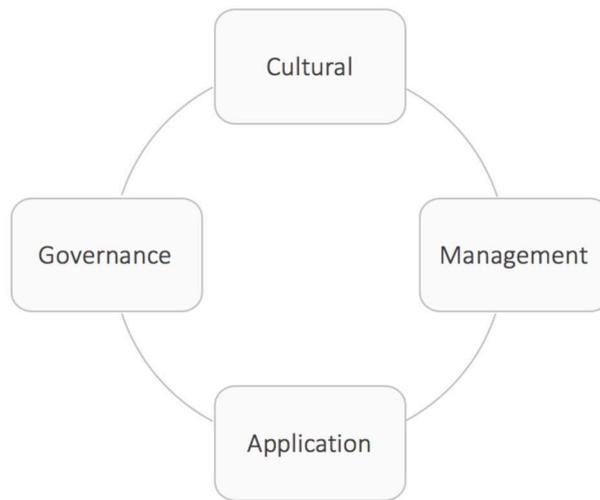
The multi-stage study has shown that software vendors that want to adopt cloud computing have to deal with change that is emerging from four areas. The first area regards the culture of the software vendor that adopts cloud computing and the culture of those that are affected by the adoption of cloud computing, e.g. the software vendor’s customers. The second area regards the impact of cloud computing on existing business processes as cloud computing might render some obsolete or require different ones. The third area regards the application the software vendor develops and distributes as cloud computing may affect features or the distribution process. The fourth area regards laws, regulations and corporate policies that might prescribe or constrain what changes the software vendor can make during the adoption, in particular when the software vendor has international customers or works with international cloud providers.

From these areas, four viewpoints have been defined that capture how the areas influence the adaptation of a company’s capabilities to cloud computing (cultural, management, application, and governance, see also Figure 22). The four viewpoints build on and extend the works of Rasmussen (1997), Sommerville & Sawyer (1997), and Sommerville et al. (1998) by making their findings more appropriate for adaptive STSs and complex systems (see Table 1 in section 1.1). Rasmussen (1997) explains how complex systems, similar to cloud computing, need to be interpreted at different levels of organisation in order to deal with technological change, e.g. government, company, staff, etc. His main argument is that, traditionally, each level was interpreted individually but that complex systems make it necessary to adopt a systemic perspective “based on functional abstraction rather than structural decomposition”. Although the focus of his work was risk management, parts of it also apply to adapting capabilities, e.g. for the adoption of new technologies. Sommerville & Sawyer (1997) and Sommerville et al. (1998) introduced the concept of viewpoints

for the requirements engineering process of software development to provide a way of capturing the needs of different types of users. In contrast to Sommerville & Sawyer (1997) and Sommerville et al. (1998) the aim of the framework developed below is not the capturing of viewpoints from users of software products, but the capturing of viewpoints that influence the adaptation of capabilities to cloud computing. By being aware of the viewpoints, software vendors are better able to accommodate the different factors that influence the adaptation of their capabilities to cloud computing. The framework, by combining the existing work with the results of the multi-stage study, aims to:

- Consider organisational and technical issues during the adoption of cloud computing in a systemic manner, which is why the four viewpoints in Figure 22 are connected and influence each other;
- At different stages of the adoption, e.g. planning, migration, post-migration;
- On different levels of organisation, e.g. software product, software vendor, industry wide, etc. to make software vendors aware of the fact that they are part of a wider network with the cloud provider and end-users and that it is important to be aware of differing boundaries when making decisions;

The four viewpoints represent a way of capturing the diverse issues software vendors encounter during the adoption of cloud computing. At the same time, the viewpoints do not restrict how software vendors may adapt their capabilities. Hence, the four viewpoints provide the right balance between a structured yet flexible and adaptable framework (this is in line with the original goal of the viewpoints developed by Sommerville & Sawyer 1997).



*Figure 22 - Four viewpoints to inform the adaptation of capabilities*

In the following, each of the viewpoints will be described in more detail. After each description, the viewpoints are applied to data collected during the multi-stage study to illustrate how they influence the adaptation of capabilities to cloud computing in practice.

#### *5.3.1.1 Cultural viewpoint*

The cultural viewpoint refers to the effects of new technologies on the way employees and customers work. Employees can be directly affected by new technologies, for example, if current business processes are changed to support the use of a new technology. Customers can be indirectly affected, for example, if the new technology leads to the company altering the products they develop or manufacture. In the case of cloud computing, customers are affected because their data will be stored outside of their direct control.

In more general terms, the cultural viewpoint captures the internalised rules and norms of behaviour that employees follow. The internalised rules and norms are not necessarily written down and employees learn over time and from interactions with other employees what actions are right and wrong. In times of change, the cultural viewpoint ensures that employees can act quickly and without consulting superiors or policy documents. At the same time, employees might need to adapt their behaviour because of changes in the environment and existing procedures might become

inappropriate. To develop and sell software products, the software vendor also needs to be familiar with the customer's culture.

The cultural viewpoint was defined as described above because the multi-stage study has shown that cloud computing affects employees and customers. The effect on both employees and customers is particularly well exemplified by the following micro themes from the first and second stage: *Build trust with providers and customers*, *Steep learning curve*, and *Technology savvy employees*. The third stage with the anticipative and reactive approach to cloud computing reflects similar cultural issues that emerged through the use of cloud computing. Particularly with the steps *Foster general acceptance* and *Embed cultural changes* of the anticipative approach and the steps *Allow short term gains*, *Institutionalise new approach* and *Embed cultural changes* of the reactive approach. The fourth stage that showed influence diagrams for the software vendors in different situations also reflects cultural elements. *CP Type* and *CP Reputation* are mostly reflecting cultural elements because the decision of what cloud provider to use depends on what the software vendor has done before, what knowledge they have and what experiences they have made prior to the cloud.

When investigating the data from the multi-stage study through the lens of the cultural viewpoint it becomes clear that in the cloud the PPs had to adapt their culture to accommodate a fast changing environment. The accommodation to a fast changing environment led to two changes in the responsibilities of employees. First, employees had to think differently about customers as some PPs shifted from bespoke software development or consultancy towards a product-oriented company. PP2, for example, reported that they “bumped into support issues” because they were growing their customer base rapidly. Developers were asked to support customers before they created a support function with dedicated support employees and tools. For PP1 24/7 support became essential as they acquired customers from different time zones.

Second, it is not easy for the PPs to predict which new product features will be successful. PP5 explained how they transferred developers away from their day-to-day jobs so they could work on developing new features. PP5 said that this went against everything they had previously been doing. The advantage of this approach is

that PP5 is able to react to new customer demands quicker. The disadvantage is that the work on the new features is self-funded, and some of the new features may never be used.

Overall, to adapt the culture to a fast changing environment, software vendors need to pay more attention to the preferences and behaviours of customers and users. Additionally, software vendors need to monitor the market situation and the behaviour of competitors more extensively.

#### 5.3.1.2 *Management viewpoint*

The management viewpoint relates to the lifecycle of a technology. Companies need to develop a plan to adopt a new technology. They need to evaluate how employees will react to the new technology and how they can be convinced to accept it. Sometimes employees feel threatened by new technologies, e.g. because they feel it replaces their job or they have to learn new skills. Companies also need to investigate how internal business processes and other areas such as sales and marketing, human resources, or financial planning are affected by new technologies. More specifically, the management viewpoint also relates to issues around coordinating everyday tasks. Everyday tasks can be internal, e.g. coordinating feature development or creating a roadmap for product development, but they can also be external, e.g. supporting the customer in using the product or increasing customer satisfaction.

The management viewpoint has been defined as described above because cloud computing needs to be integrated into the business processes of the software vendors and, afterwards, needs to be continually managed. The need for integration and continuous management is exemplified by the following micro themes of the first and second stage: *In control of provision and updates*, *Standard set up process*, *Focus on core tasks*, and *Robust backend processes*. The third stage reflects the need for integration and continuous management similarly well. Particularly the steps *Establish project boundaries* and *Articulate processes and values* of the anticipative approach and *Set context for change* and *Align capabilities to new approach* of the reactive approach show that. The fourth stage, with its influence diagrams, also shows the need for managing the cloud environment. Particularly the following blobs require

the software vendors to manage their products in the cloud and the cloud environment: *CP Technical Functionality, Customer Satisfaction, and Using multiple CP's*.

Several PPs explained that the migration of their software product into the cloud created a positive attitude amongst employees towards new technologies (see also Alshamaila et al. 2013 who made similar findings when investigating the adoption of a new technology). Since having moved to the cloud, PP5 has developed a smartphone and tablet version of their product. PP1 is currently developing a tablet version. PP2 now supports smart pens. The smart pens write like normal pens but have a memory. After writing on a special sheet of paper users can connect the pen to a computer where the handwritten text is converted and tasks like the sending of an email are executed automatically.

The management viewpoint is further relevant for all PPs as in the cloud they are required to adopt a more structured approach to product development. In the cloud, the PPs release updates on a regular basis, in contrast to ad-hoc updates before the cloud. The ability to deliver updates on a regular basis requires the PPs to adapt internal business processes to ensure that updates are feasible and developers acquire the right skills. Furthermore, it requires the PPs to have appropriate software development environments.

A last factor that is part of the management viewpoint closely relates to the cultural viewpoint. The cultural viewpoint stated above that the PPs have to adapt their culture due to a larger number of customers that increase the number of support requests. Some PPs, PP2 in particular, acquired software to manage their support requests and feature development. The software only provides an advantage, however, if it is appropriately managed. Connected systems, e.g. email, also need to be managed appropriately.

### *5.3.1.3 Application viewpoint*

The application viewpoint relates to the development and distribution of software applications. Companies have to be aware of the dependencies that can enhance or

stifle the development of their product (e.g. APIs) and have to avoid unnecessary risks and cascading failure events. The application viewpoint also captures issues affecting the daily work of software developers, e.g. what skills they need or have to develop, and what programming languages the software vendor uses. It can also exist in relation to the software vendor's mission, in order to decide what features will be developed or how customer feature requests will be handled. As the application viewpoint is also concerned with the distribution of software applications, the set up of a software vendor's product at the customer site can influence decisions made with regard to this viewpoint.

The application viewpoint has been defined as above because cloud computing affects the way the software vendors develop and distribute their products. The need to change the way software vendors develop and distribute products in the cloud is exemplified by the following micro themes of the first and second stage: *Increase efficiency of product development* and *More responsibilities*. The need is also exemplified by the step *Assess future environment* of the anticipative approach to cloud computing. The following blobs described for the influence diagrams of the fourth stage suggest similar conclusions: *Product functionality* and *Customer software requirements*.

When investigating the data from the multi-stage study through the lens of the application viewpoint it becomes clear how the PPs adapted their software development approaches to accommodate cloud computing. PP3 combined several, previously independent products into one cloud product. They also decided to develop their products for the cloud using Java because a sister company had previously moved one of their products to the cloud and used Java. Senior management hopes to build on the experiences of the sister company.

PP1 was able to offer additional features to customers. PP1 had a customer who required the use of Citrix, for example. Whilst PP1 did not have Citrix experience, their cloud provider offered support to help develop new features for that customer.

The above examples illustrate how closely the application viewpoint is coupled with the cultural and management viewpoints. Decisions around the distribution of a software vendor's product are influenced by the culture of the software vendor's customers. The influence of customer's culture on the distribution processes can be seen, for example, by the fact that all PPs continued to offer their non-cloud product alongside the cloud product in case customers were concerned about data being stored in the cloud. Decisions around the development of product features or the product roadmap, as part of the management viewpoint, are influenced by the application viewpoint as developers need to develop the right skills, dependencies that can enhance or stifle product development need to be investigated, and the approaches to product development need to be managed, e.g. SCRUM.

#### *5.3.1.4 Governance viewpoint*

Technologies and software applications need to comply with various governmental and institutional requirements and laws as well as corporate policies. Companies might also have to be aware of potential customer requirements and policies. Governmental and institutional requirements and laws can be particularly important for industries that are highly regulated, e.g. air traffic management, oil and gas, or health care. In contrast to the cultural viewpoint, the governance viewpoint relates to the formal rules and norms of behaviour. Governance is important for cloud computing because it becomes easier for software vendors to enter new markets abroad, where different laws and regulations might exist (e.g. data protection laws and security policies).

The governance viewpoint has been defined as above because in the cloud software vendors have the opportunity to operate more internationally and rely on cloud providers that are not necessarily co-located. Hence, customers and cloud providers might have to adhere to different laws and regulations that can affect the way software vendors operate. The opportunities to expand into new markets are particularly well exemplified by the following micro themes of the first and second stage: *Gain competitive advantages* and *Obstacles to sale reduced*. The need to be aware of differing laws and regulations is exemplified by the following blobs that

have been described for the fourth stage: *Data Location* and *Customer Data Requirements*.

Investigating the data from the multi-stage study through the lens of the governance viewpoint reveals to what extent governance constraints the options of the PPs. PP1 reported a negative experience relating to the contracts with a customer, and with their own cloud provider. The customer terminated the contract early and as a result PP1 wanted to terminate the related contract with their cloud provider, but could not do so. The customer stopped paying for the product but PP1 still had to pay for the resources in the cloud for the remainder of the contract.

PP3 explained the challenge they face with customer data. Many of PP3's customers want their data to remain in their own country. The problem for PP3 is that in some countries where they have customers there are no cloud providers, e.g. in some African countries. One reason for customers wanting their data to remain in their country are, as PP3 stated, the local laws and regulations.

The governance viewpoint can also act as a constraint on the decision-making continuum for any of the three other viewpoints (i.e. cultural, management, and application). Corporate policies, e.g. the informal learning environment of PP2, can affect decisions made for applications (i.e. application viewpoint) with regard to the development of new skills of software developers. Handbooks or product documentations, e.g. those of PP5, can influence cultural or management decisions with regard to the handling of customer support requests (i.e. cultural and management viewpoint).

### **5.3.2 Advantages of four viewpoints over other frameworks**

Chapters 1 and 2 explained that there are no frameworks that capture the underlying processes for adapting capabilities to technological discontinuities adequately. The viewpoints introduced above represent an attempt to develop such a framework. It is worth comparing the viewpoints to the frameworks introduced in section 2.1 to draw out advantages and disadvantages.

Section 2.1 introduced several frameworks for adapting capabilities and classified these into structural and gradual change frameworks. It was concluded that the existing structural change frameworks are inappropriate for SMEs, mainly due to the limited resources of SMEs. Gradual change frameworks, with their focus on dynamic capabilities, appeared to be the more promising candidates. However, gradual change frameworks lacked specific examples of dynamic capabilities and how these were developed. Section 4.3 identified and described the capabilities the PPs adapted to cloud computing. It was concluded that these are specific examples of dynamic capabilities. Therefore, section 4.3 advanced the theory of dynamic capabilities.

However, section 2.1 identified more substantial drawbacks of gradual change frameworks other than the lack of specific examples. Two drawbacks were most notable. First, neglecting interdependencies between different parts of companies, whereas it was argued that interdependencies could influence the adaptation of capabilities due to the nature of today's complex systems (see Table 1 section 1.1). Second, focusing on one-time adaptation of capabilities, whereas it was argued that it is necessary to assist companies in continually adapting their capabilities as the internal and external environment changes due to the nature of complex systems like cloud computing.

The framework with the four viewpoints introduced in the previous section was developed on the background of the drawbacks identified in section 2.1. By identifying the four viewpoints that influence the adaptation of capabilities when migrating software products into the cloud, companies are made aware of the interdependencies that play a role during the migration. The interdependencies are present in the framework on three levels. The first level regards interdependencies between the software vendor and their external environment. The cultural viewpoint captures developments in the market and customer behaviour that can influence the cloud migration. The governance viewpoint captures the fact that the software vendor is subject to laws and regulations that might exist in different markets or countries they operate in and that can influence the decision making continuum available. The second level regards internal dependencies that can influence the migration steps into the cloud. The cultural and management viewpoint capture the internal culture of the

software vendor and its business processes. Both can influence the efficiency and effectiveness of the cloud migration. The application viewpoint captures the work of software developers, which can influence how products are migrated into the cloud. The third level of interdependencies regards the different stages of the adoption, e.g. planning, migration, and post-migration. By focusing on viewpoints that influence decisions rather than plotting the processes of adapting capabilities, the framework provides an appropriate balance between a structured yet flexible approach. This balance allows the application of the framework not only during the migration of software products into the cloud but also to prepare the migration and to continue to investigate the impact of cloud computing after the migration. Hence, the framework allows software vendors to continually adapt their capabilities as the internal and external environment changes.

### **5.3.3 Limitations of the multi-stage study**

The multi-stage study achieved its goal of identifying the underlying processes of adapting capabilities to cloud computing by investigating the software vendors from a systemic perspective. However, there are limitations to the multi-stage study that are discussed in the following.

In hindsight, stage 3 of the multi-stage study revealed many insights about the PPs and their approach to cloud computing. If this stage would have been carried out first, instead of investigating the impact of cloud computing on product development, it might have been possible to structure subsequent stages in a more targeted way, i.e. get to the issues that are important to the interviewees faster during interviews. However, there was a reason to design the first stage of the multi-stage study around the impact of cloud computing on product development (see section 3.3). The participants of the multi-stage study were all SME software vendors, thus, it was important for them to address the impact of cloud computing on product development first so that they could make sure their products succeed in the cloud.

Focusing on general issues (like product development and internal impact) allowed the study to take a high level perspective. However, when taking a high level perspective, it is possible to miss minor actions that can have a large impact. In other

words, the results of the minor actions are represented in the results but the action itself has not been identified. The way the study was set up tried to address this by (1) sending the high level interview questions to the interviewees in advance and (2) staying flexible during the interviews to address issues that appear important to the interviewee (e.g. asking the interviewees which questions are important to them).

Asking questions about the migration of software products into the cloud after it happened often lets the interviewees only reflect the positive experiences while neglecting any setbacks. The way the study was set up tried to address this by asking the same questions from different angles and following the PPs over a long period. Round 3 in particular addressed this issue by going through the decisions of the adoption process. While asking the questions it was always pointed out that the interviewees should not only recount positive experiences or the option they took but also explain why they dismissed options.

For some of the results of the study it is not possible to state with absolute certainty if they are due to a migration of products into the cloud. They can also be the result of general product development, e.g. the development of a feature many customers have been waiting for. While analysing the results, an attempt was made to account for this by looking for similarities between the PPs. It is highly unlikely that two PPs develop a feature that is requested by many customers at the same time. It is not, however, possible to eliminate the factor entirely.

The interviews were done with Managing Directors or leaders of product development. Due to the high-level perspective that was taken for the study it was desirable to interview people that have a birds-eye view on the company. Furthermore, for the investigation of capabilities a high-level perspective is advantageous. For some questions, however, it could have been helpful to interview people lower in the hierarchy, e.g. developers. In some cases, the interviewee confirmed information they provided by checking with colleagues lower in the hierarchy or by providing internal documents (e.g. presentations).

## **5.4 Conclusions**

This chapter presented and discussed the results of the third and fourth round of the multi-stage study with five SME software vendors that migrated their software products into the cloud. The third round identified major decisions the PPs made during the adoption and categorised these into an anticipatory approach and a reactive approach to cloud computing. The fourth round investigated the relationship of the software vendors with their cloud providers to understand the effects of that relationship on the software vendors' customers. The findings provide two major conclusions.

First, a framework in the form of four viewpoints has been developed that captures the underlying processes of adapting capabilities to cloud computing. The framework provides the answer to RQ1: *For SME software vendors migrating their software products into the cloud, what are the underlying processes of adapting core capabilities to cloud computing?* The software vendors adapt their capabilities to cloud computing by investigating how the four viewpoints (cultural, management, application, and governance) influence their existing capabilities in the cloud. Investigating the influences on existing capabilities enables software vendors to adapt their capabilities to cloud computing appropriately.

Second, the framework should be extended to allow other software vendors to follow the underlying processes of the PPs in a similar way. The extension should also allow a more focused investigation of resilience. The multi-stage study highlighted throughout how actions the PPs took increased their resilience. Thus, the four viewpoints already reflect some aspects of resilience. It would be desirable, however, to relate both concepts, adapting capabilities and increasing resilience, more closely to allow aspects of resilience actively influence the adaptation of capabilities. Influencing the adaptation of capabilities through resilience (i.e. create processes that are robust yet flexible enough to succeed under varying conditions, see section 2.2) would also be a step towards answering RQ2. The next chapter tests the Functional Resonance Analysis Method (FRAM) as a first step towards answering RQ2 and relating the concepts of capabilities and resilience more closely.

## **6 Increasing resilience by migrating software products into the cloud**

The objective of this chapter is testing the Functional Resonance Analysis Method (FRAM) as a method that can inform steps to increase and measure the resilience of software vendors that migrate their software products into the cloud. Thus, FRAM is tested as the answer to RQ2: *How can SME software vendors that plan to migrate their software products into the cloud increase their resilience with the adaptation of capabilities affected by cloud computing and how can the increase be measured?* In this chapter FRAM has been applied with PP1 from the multi-stage study to find out (1) if they became more resilient by migrating their software product into the cloud and (2) if FRAM can explain why they became more resilient.

This chapter is structured as follows. The next section decomposes the above goals to explain how they can be achieved (see section 6.1). The motivations for using FRAM to investigate the resilience of software vendors were laid out in section 2.4.2. Section 6.2 goes into more detail by explaining the elements and steps of a FRAM analysis. Afterwards the methodology for the FRAM analysis is described and the reasons for carrying out the FRAM analysis with PP1 are explained (see section 6.3). The last two sections of this chapter analyse and discuss the results of the FRAM analysis with PP1 (see sections 6.4 and 6.5).

### **6.1 Goals of the FRAM analysis**

The multi-stage study illustrated what effects cloud computing had on the software vendors and what actions they took to exploit the opportunities presented by cloud computing while avoiding adverse effects. This chapter will build on the classification of effects and actions by carrying out a FRAM analysis to identify in what areas of the company PP1 took actions first, if these actions had an effect on the resilience of PP1 and how the actions were gradually rolled out to the entire company. Thus, it will be possible to conclude if FRAM can inform steps that are necessary to increase system resilience.

The goals of the FRAM analysis with PP1 will be achieved by creating two separate FRAM models. The first FRAM model will show PP1's functions before the migration of their product into the cloud, i.e. the traditional way of doing business. The second FRAM model will show the functions after the migration of their software product into the cloud. With two FRAM models it is possible to understand how and why individual functions have changed and what the impact on the company as a whole was. Furthermore, it is possible to conclude what actions PP1 took to exploit opportunities presented by cloud computing while avoiding adverse effects.

## 6.2 Elements and steps of a FRAM analysis

Section 2.4.2 briefly introduced FRAM (Hollnagel 2012b), which consists of *functions* that are connected with each other through *aspects*. *Functions* are abstractions to capture work routines and related resources, tangible and intangible ones. The six aspects are **I**ntput, **O**utput, **T**ime, **C**ontrol, **P**recondition, and **R**esources. Figure 23 shows an example of a FRAM model.

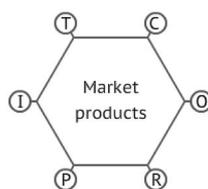


Figure 23 - Example of a single FRAM function and its six aspects responsible for marketing

It is possible to define functions on different levels of organisation. The level of detail and the number of functions depend on the purpose for which FRAM is being used. <Market products>, for example, could be a function, like in Figure 23, but it is also possible to go into more detail by decomposing <Market products>, as shown in Figure 24 at the end of this section (page 122). The first step in a FRAM analysis should therefore always be the identification of functions that are necessary to succeed in everyday situations. It is possible and sometimes advantageous to concentrate on high-level functions at first and go into more detail in later stages of the analysis. It is not important which function is identified first. In a FRAM analysis there are not always clear start and end functions. Furthermore, the aspects of the functions ensure that all necessary functions are identified.

Once an initial set of functions has been identified, the functions need to be described in more detail by defining (some of) their aspects:

- **Input:** is used or transformed by the function to produce the Output. Input can be anything like material, information, etc. The Input starts a function.
- **Output:** the result of what the function does. The Output can be anything like material, information, etc. When the Output has been produced, the function is completed.
- **Precondition:** has to be true or verified in order for a function to start. It does not in itself, however, constitute a signal to start a function. The Input starts a function. This distinction can be used to decide whether something should be an Input or a Precondition.
- **Control:** regulates or supervises a function so that the desired (or planned) Output is produced. Control can be a plan, a set of guidelines or rules, a schedule, etc. Control can also be social expectations, e.g. those by management or supervisors.
- **Resource:** is consumed when the function is executed. A Resource can be anything like matter, information, a machine, a software tool, etc.
- **Time:** captures the different ways in which time can affect a function. Time can be considered another form of Control. For example, a function may need to be carried out before, after, or in parallel to another function. Time can also relate to a single function that needs to start at a certain point in time.

Functions need to have at least an Input or Output. Only the Output of a function can be connected to other aspects of other functions, i.e. connecting Precondition to Control is not allowed. It is often useful not to describe all aspects of a function at first, as this can make the analysis complex and it is easy to lose sight of the bigger picture. It is recommended to describe only those aspects that are deemed appropriate for the analysis and for which information is available. For all other cases *background functions* can be defined. Background functions only have an Input *or* Output and are assumed to be stable during the execution of the function (background functions are grey in FRAM models).

FRAM allows the analysis of *resilience* (see section 2.4.2 for a description of the underlying principles of FRAM). As software vendors that migrate their software products into the cloud take over responsibilities from the customer and, at the same time, outsource responsibilities to the cloud provider, understanding the resilience of the software vendor and *how it can be increased* enables them to react to and anticipate changing circumstances faster. In addition, a higher level of resilience will also help the software vendor bounce back to normal performance after a major negative event, e.g. if the cloud provider has an outage.

Section 2.4.2 introduced the concept of performance variabilities, as it was the main reason for choosing FRAM as a method to investigate resilience. The concept of performance variabilities captures how the variability of individual functions in FRAM models can sometimes reinforce each other, causing the variability of one function to exceed its limits. The consequences can be negative as well as positive (Hollnagel 2012b). The identification of performance variabilities in FRAM requires the instantiation of a FRAM model. As the aim of a FRAM model is to capture functions and their couplings for everyday situations, a FRAM model always shows potential couplings between functions until the model has been instantiated. Instantiating a model means investigating the functions and couplings in/for a specific situation in a specific context. For the purpose of this chapter it is the migration of software products into the cloud. By knowing how functions behave in this situation and context, it is possible to identify performance variabilities. Three types of performance variabilities can be distinguished.

First, the function itself can experience performance variabilities, so called *internal performance variability*. In this case a function can fail due to organisational pressures that affect human performance or because equipment has not been maintained properly (wear and tear). Second, the function can fail due to a change in the working environment, so called *external performance variability*. In this case, a function fails because it operates outside its designed parameters, e.g. extreme weather conditions. Third, the output from other functions can affect *downstream functions*. Downstream functions are those functions that use the Output of other

functions as Input, Precondition, etc. If an upstream function varies in performance (or fails) and the Output is not as it should be (or not available at all) it can affect downstream functions. For example, if the Output of an upstream function is the Precondition for a downstream function, the downstream function will not start as not all conditions are fulfilled (similar to the example in Figure 24 at the end of this section).

The Output of functions can be affected by the internal, external, and downstream performance variability in terms of time and precision. Performance variability can cause functions to produce Output (1) *too early*, (2) *on time*, (3) *too late*, or (4) *not at all*. With regard to precision, performance variability can cause functions to produce Output that is (1) *precise*, (2) *acceptable*, or (3) *imprecise*. Investigating performance variabilities only with regard to time and precision represents a simplified version of performance variabilities. There is also a more elaborate way, which includes the identification of performance variabilities through common conditions (CCs). CCs are, for example, the availability of procedures and plans, conditions of work, circadian rhythm and stress, etc. (a complete list can be found in Hollnagel 2012a). The more elaborate way has been deemed inappropriate for the adoption of cloud computing because the majority of CCs deal with accidents where people's lives are at risk (and thus the CCs cover factors related to relevant causes). The reason for this is that FRAM has originally been developed for accident investigations (retrospective analysis) and safety assessments (prospective analysis). Although in today's environment all software vendors rely on technologies such as cloud computing, they are business critical but do not necessarily cause the loss of life in case of disruptions.

When using the more elaborate way to identify performance variabilities it is necessary to categorise each function:

- Human (M-functions): functions that are carried out mainly by people, thus, performance variabilities depend on the performance of people;
- Technology (T-functions): functions that depend on technology to perform appropriately, thus, performance variabilities depend on the performance of the technology, such as automated functions;

- Organisation (O-functions): functions that are carried out by a group of people with additional means, e.g. a technology, thus, performance variabilities depend on the performance of the whole group;

When investigating the performance variabilities of a software vendor that plans to migrate their software product into the cloud, the classification of functions into the three categories will not be done. This study aims to investigate how PP1 has migrated their software products into the cloud from a technical *and* organisational perspective. Hence, the study will only capture those functions that contain both perspectives, i.e. O-functions.

The FRAM model from Figure 23 has been extended to show <Market products> in more detail in Figure 24. To connect the first three functions only their Input and Output aspects have been described. The description of the aspects is shown as text boxes on the lines between the hexagons. <Provide advertising budget> is different from the rest as its Output is a Resource aspect for <Develop under 20s commercial> (it is a Resource aspect because budget is consumed during the execution and needs to be available throughout the execution of the function). <Develop under 20s commercial> can only start (and be continued) when budget is available. The functions in Figure 24 have also been analysed for performance variabilities. <Develop under 20s commercial> is likely to experience performance variabilities (shown by the wave symbol in the hexagon) due to the Resource aspect of <Provide advertising budget>. If the budget is not sufficient, <Develop under 20s commercial> is likely to produce an Output that is imprecise. Thus, the performance of an upstream function affects the performance of a downstream function.

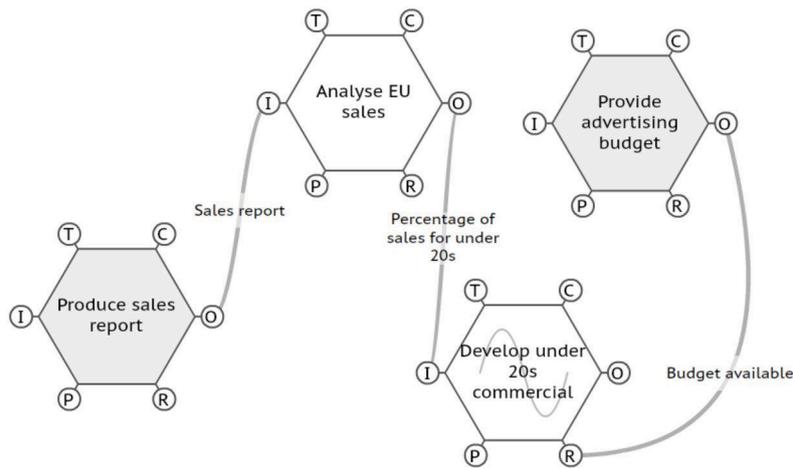


Figure 24 – Same FRAM as in Figure 4 to show performance variabilities (as a wave symbol in the hexagon). Background functions are shown in grey.

### 6.3 Methodology of the FRAM analysis with PP1

To conclude if FRAM is a suitable method to increase and measure the resilience of software vendors when migrating their software products into the cloud, it is necessary to apply it with a software vendor in practice, i.e. a case-study approach was chosen for the reasons elaborated in section 3.2. The following section will explain the methodology for testing FRAM. PP1 from the multi-stage study was chosen due to their operating model (see next section) and because sufficient trust had been established between PP1 and the researcher during the multi-stage study. As FRAM aims to identify functions necessary for everyday situations, a FRAM analysis might reveal competitive advantages of the participant's company. The participant will only feel comfortable enough to discuss functions in the required level of detail if trust has been established between the participant and the researcher.

#### 6.3.1 Motivation for doing a FRAM analysis with PP1

PP1, from the original sample of companies from the multi-stage study, was chosen for the FRAM analysis as they continue to sell the on-site and cloud version of their product at the same time. In fact, PP1 is a software vendor that offers five different sales models to customers:

1. On-site installation with perpetual license
2. PP1 hosts the solution in the short term until the customer is ready (with perpetual license)

3. Solution is hosted in the cloud in the short term until the customer is ready (with perpetual license)
4. Solution is permanently hosted in the cloud (with perpetual license)
5. Solution is permanently hosted in the cloud (with subscription)

Three of these five sales models, however, can be neglected. In general, there are no differences in the services customers receive regardless of whether they bought a perpetual license or use the subscription model. This means sales models 4 and 5 can be considered as being the same. PP1 recently discontinued to offer sales model 2 and offers sales model 3 instead. Since sales model 3 requires functionality from both sales models 1 and 5, this sales model is sufficiently represented by models 1 and 5. This leaves PP1 with only two different sales models that are worth investigating in more detail:

1. On-site installation
2. Solution is permanently hosted in the cloud

The FRAM analysis with PP1 was carried out during a 2-hour interview with the Managing Director of PP1. Data collected during the multi-stage study (see chapters 4 and 5) was used to prepare the FRAM analysis (see below). The FRAM analysis was carried out in four steps:

1. Identify functions and aspects that show PP1 before the migration to the cloud, i.e. on-site installation
2. Identify performance variabilities in the functions for the on-site installation
3. Identify functions and aspects that show PP1 after the migration to the cloud
4. Analyse the effects of cloud computing on the performance variabilities by comparing the before and after cloud migration FRAM models

### **6.3.2 A generic customer lifecycle to guide the data collection**

A generic customer lifecycle has been developed to guide the data collection with PP1 (see Figure 25). The lifecycle will assist in identifying functions and describing their aspects. It has been developed based on data collected during the multi-stage study and was considered as a model to guide the analysis and not one to prescribe it. Other software vendors might find that the generic customer lifecycle does not apply to

them, depending on their business model and the kind of products or services they distribute. In addition to presenting and explaining the generic customer lifecycle to PP1, a similar lifecycle that takes a product focus was presented (see Appendix D). PP1 deemed the customer lifecycle as more appropriate for their business model.

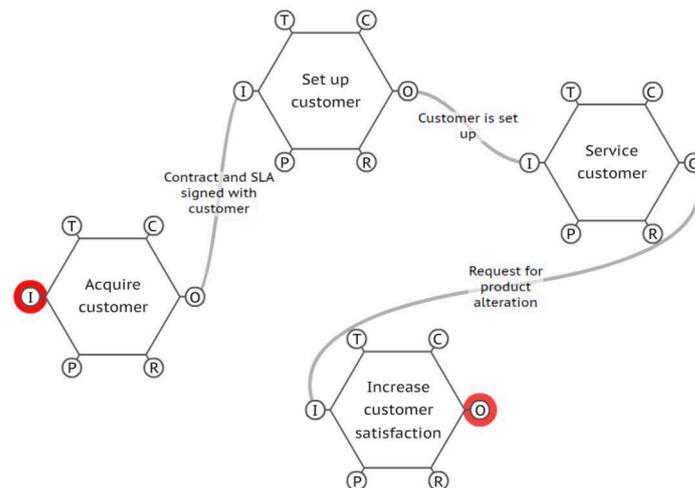
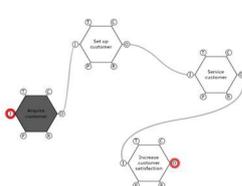


Figure 25 - Visualisation of the FRAM functions. Please note that the I (for Input) and O (for Output) in the Acquire customer and Increase customer satisfaction functions respectively are circled red because the aspects do not lead to another function.

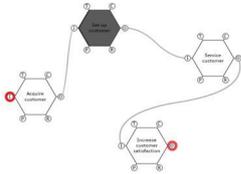
To keep an open mind for the subsequent analysis, only high-level functions and aspects have been defined in the generic customer lifecycle. In the following, the generic customer lifecycle will be explained using the FRAM model notion.

The customer lifecycle has four functions: (1) <Acquire customer>, (2) <Set up customer>, (3) <Service customer>, and (4) <Increase customer satisfaction>.

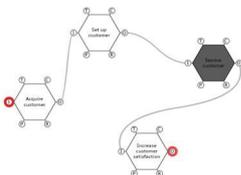


<Acquire customer> contains everything that is necessary to acquire new customers. This can, for example, include elements of Sales & Marketing such as advertisement, or calling potential customers. The function Acquire customer has two Inputs that can start the function: (1) [Project bid] where the software vendor bids on a publicly advertised project of a potential customer, (2) [Customer request] where a customer contacts the software vendor directly, e.g. because they came across an ad or searched for the product on the Internet. Within the function Acquire customer, a potential customer

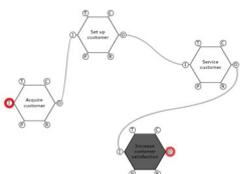
is converted to a customer. The Output of the function is, therefore [Contract and SLA (Service Level Agreement) signed] between the customer and the software vendor.



<Set up customer> contains everything necessary to enable customers to use the product or service the software vendor provides. This can include, for example, installing the product in the customers data centre, train the customer’s employees, etc. The Output of the function is [Customer is set up] at which point they are able to use the product or service distributed by the software vendor.



< Service customer> deals with everyday problems customers might have. In other words, it contains everything necessary to support the users of the software vendor’s product. This can mean responding to technical questions users might have, e.g. if something is not working, assisting the customer in updating the products, or capturing feature requests. The Output of this function is [Request for product alteration], because only the development of new product features is not covered by <Service customer>. That is the purpose of the last function in the lifecycle.



<Increase customer satisfaction> contains everything necessary to retain customers as long as possible. In other words, this function deals with long-term customer issues, such as developing product features, whereas the previous function deals with short-term user issues, such as bugs or technical difficulties. The Output is <Enhanced product functionality>. With this function, the lifecycle comes back to the start as a new customer has been acquired and their long-term satisfaction is ensured. It would be possible to connect <Increase customer satisfaction> to <Acquire customer>, as the Output [Enhanced product functionality] can function as a

Control aspect for <Acquire customer> (because new features can make it easier to expand the customer base).

During a FRAM analysis, a graphical presentation of the FRAM model can be produced by using the software tool FRAM Model Visualiser. The result for the functions described above is shown in Figure 25. A table presentation for one function is shown in Table 4. They can be useful to provide additional information, e.g. description of the function (the FRAM Model Visualiser provides the option to produce a report that contains the graphical presentation and tables). Depending on the number and couplings of the functions, the visualisation of the FRAM model can become complicated and referring to the tables might be more practical.

*Table 4 - Table presentation of the second function of the customer lifecycle FRAM*

<b>Name of function</b>	<b>Set up customer</b>
Description	Containing all actions and tasks to enable the customer to use the software vendor's products and services.
<b>Aspect</b>	<b>Description of aspect</b>
Input	Contract and SLA signed with customer
Output	Customer is set up
Precondition	
Resources	
Control	
Time	

### **6.3.3 Avoiding drawbacks from similar FRAM studies**

Studies that applied FRAM in practice are very limited. Those studies that did apply FRAM in practice had major drawbacks that this study aims to avoid. The majority of FRAM studies identified a large number of functions. Having too many functions can stifle the ability to understand situations clearly and to identify measures that can dampen performance variabilities. Herrera & Woltjer (2010), for example, identified 19 functions and Frost & Mo (2014) identified 24 functions. With a large number of functions people, who are unfamiliar with the situation, will struggle to understand the FRAM model due to cognitive limits. While creating a FRAM model, it is important to focus on functions that are necessary for the situation that is being investigated.

Other FRAM studies mix work-as-imagined and work-as-done (e.g. Frost & Mo 2014). Work-as-imagined shows how managers and decision makers imagine work is being carried out. Thus, work-as-imagined often closely resembles formal process and flow charts. Work-as-done shows how work is carried out in practice. Workers often have to adjust their behaviour to react to new information and they create workarounds that differ from formal process and flow charts. Thus, work-as-imagined often differs from work-as-done and their FRAM models can look very different. Two studies (Belmonte et al. 2011; de Carvalho 2011) have used FRAM without accounting for differences between work-as-imagined and work-as-done. They built their FRAM models based on existing reports and official documents, e.g. failure reports and process charts, and used FRAM only to validate the results of these reports without interviewing workers, i.e. only showing work-as-imagined. Furthermore, when building FRAM models based on the results of other reports, one has to account for differences in the underlying models and assumptions of each method. As every method represents an abstraction from reality they exclude certain aspects. If, however, method A excludes aspects that are important for method B, method B will not show the situation accurately and can even provide a false sense of security. Hence, it is important to describe functions and their aspects with raw data or, at least, use compatible methods, i.e. those that rely on similar underlying models and assumptions.

## **6.4 Results**

### **6.4.1 Before cloud migration FRAM model**

The functions from the customer lifecycle of the previous section were explained to the Managing Director of PP1 and their applicability discussed. With the information gathered it was possible to provide a much more detailed description of the functions and their aspects in comparison to the generic customer lifecycle (see Figure 26). In the following, each function of the before cloud migration FRAM model will be explained in more detail to show work-as-done for the on-site installation of PP1's software product.

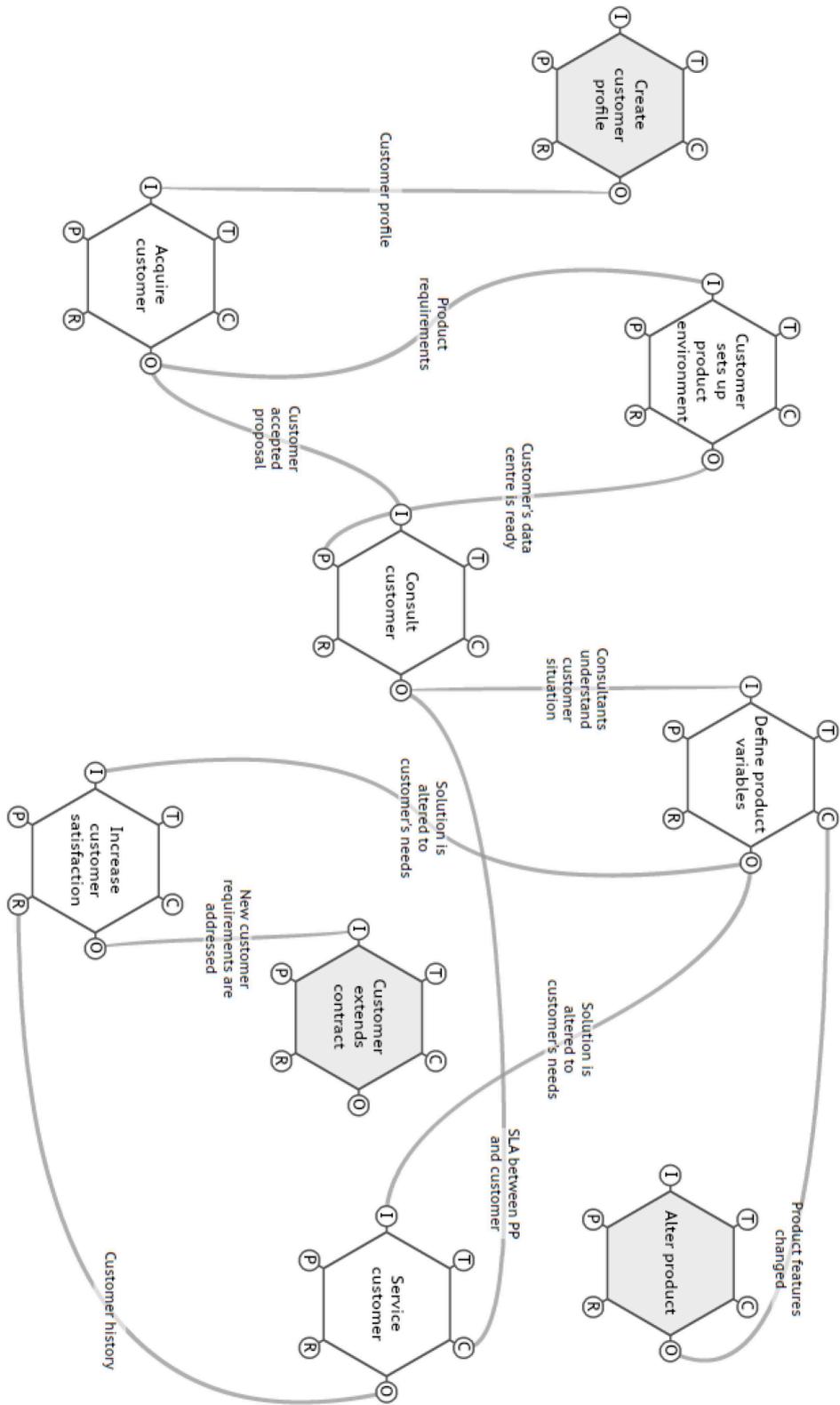
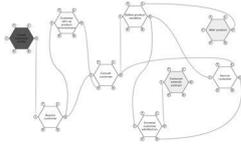
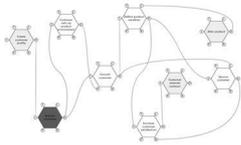


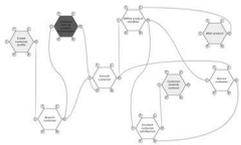
Figure 26 - Before cloud migration FRAM



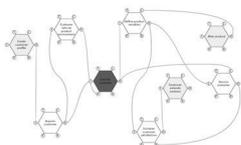
<Create customer profile> is a background function that starts the FRAM model. PP1 is working together with another company that creates profiles of potential customers that PP1 uses to contact them. [Customer profile] is therefore the Input for <Acquire customer>.



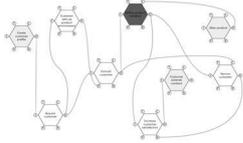
<Acquire customer> converts potential customers into actual customers. At the end of this function, the customer has accepted the proposal from PP1 and receives the product requirements. The product requirements describe what kind of hardware and access to databases PP1 needs from the customer so that the product can be installed in the customer's data centre.



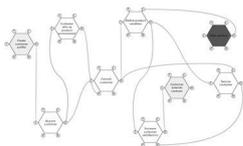
<Customer sets up product environment> represents the tasks related to setting up the hardware and access to databases. This function is outside of PP1's control. PP1 can provide guidance to the customer but the customer retains the final responsibility. Only when this function is completed and the Output [Customer's data centre is ready] has been produced can <Consult customer> start. Therefore, the Output of <Customer sets up product environment> is a Control aspect of <Consult customer> (since the data centre needs to remain ready during the execution of <Consult customer>, otherwise it would have been a Precondition).



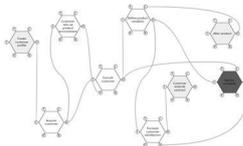
<Consult customer> collects information necessary for tailoring the initial installation of PP1's product to the specific needs of the customer, e.g. reflecting the customer's business processes. The function produces two Outputs that are necessary towards enabling the customer to start using the product. The first Output is [Consultants understand customer situation] that is the Input for <Define product variables>. The second Output is [SLA between PP1 and customer] that is a Control aspect for <Service customer>.



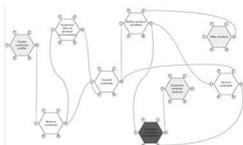
<Define product variables> uses the information collected in <Consult customer> to tailor the product to the customer's needs. The generic solution that is installed initially contains variables, such as the number of items on a report, which have been defined very broadly. The consultants can define the range of these variables more strictly to represent the customer's situation accurately. Therefore, the Output is [Solution is altered to customer's needs]. At this point, the customer can start using the product, which is shown by the fact that the Output is the Input for <Service customer> and <Increase customer satisfaction>. If, however, the product is not able to capture the customer's situation appropriately the background function <Alter product> becomes important.



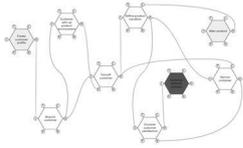
<Alter product> is a background function that captures the event if a software developer from PP1 has to alter the backend of the solution to satisfy customer requirements. Because this function is not relevant to every customer, its Output [Product features changed] is a Control aspect of <Define product variables>.



<Service customer> is carried out by PP1's support staff and deals with everyday problems users might encounter, e.g. a report is not being produced as expected. Customers have a phone number and FAQ (Frequently Asked Questions) sheets they can consult in these cases.



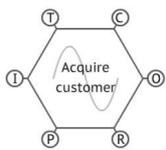
<Increase customer satisfaction> is handled by the sales department. The sales people who carry out this function are, however, different from those carrying out <Acquire customer>. The function's goal is to retain customers by convincing them to buy upgrades or new products. To achieve this goal, the function uses [Customer history] that is the Output of <Service customer> to know what kinds of issues the customer's users struggle with.



<Customer extends contract> is a background function that ends the FRAM model. <Increase customer satisfaction> produces the Output [New customer requirements addressed] which is the Input for <Customer extends contract> and is used as a means to show that customers will extend the contract if their requirements are addressed.

### 6.4.2 Identifying performance variabilities

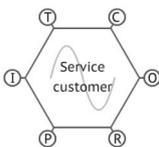
Together with the Managing Director the researcher discussed in which functions PP1 either sees potential performance variability or has experienced performance variabilities in the past (in terms of time and precision). Potential performance variabilities in functions are represented through a wave symbol in the hexagon. For three out of the six foreground functions potential performance variabilities could be identified. They are elaborated in the following.



<Acquire customer>: Customers sometimes have trouble signing off on buying the software as the sales price is high which requires people higher in the hierarchy of the customer to sign off on the investment. This means the process takes longer and there are more people that can potentially veto the purchase.



<Customer sets up product environment>: The customer often fails to accomplish this task on time as the customer's IT department needs a lot of time to buy new hardware and install it. In some cases this task takes longer than the time the customer needs PP1's product.



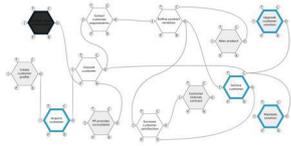
<Service customer>: The customer sometimes fails to install updates properly or at all. More specifically, customers try to save time by installing updates directly to their live environment, without testing them first.

### 6.4.3 After cloud migration FRAM

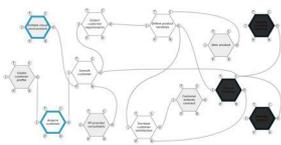
Cloud computing changes the responsibilities of PP1. PP1 takes over responsibilities previously held by the customer and, at the same time, passes on responsibilities over the infrastructure to the cloud provider. PP1 is required to reflect these changes in the

FRAM model. A FRAM model showing the functions after the migration into the cloud reveals how PP1 reacted to the changes in responsibilities (see Figure 27). They are explained in detail below.





<Customer sets up product environment> needs to adapt as it becomes obsolete through cloud computing. <Initiate cloud environment> has been introduced to replace <Customer sets up product environment>. <Initiate cloud environment> is responsible for renting the cloud resources from the cloud provider on which the customer's solution will be installed. Similar to the other two functions that are introduced below, this new function also takes over responsibilities from the customer.



<Service customer> needs to adapt, as it will take over responsibilities from the customer. Next to supporting the customer in their everyday use of the product, in the cloud this function is also responsible for maintaining the product and installing updates. PP1 introduced two new functions: <Maintain solution> to ensure that the product is running as is expected by the customer; <Upgrade customer solution> to fix bugs and install new features.

All remaining functions of PP1 do not need to adapt, as they are not immediately affected by the migration of PP1's software products into the cloud. Furthermore, PP1 decided that other functions could adapt but should not, at least at first, to make the migration into the cloud efficient.

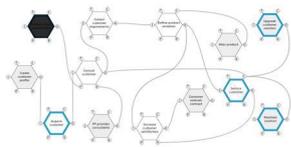
#### 6.4.4 The impact of cloud computing on performance variabilities

Three functions had performance variabilities *before* PP1 migrated their software product into the cloud. They were <Acquire customer>, <Customer sets up product environment>, and <Service customer>. The migration of PP1's software product enabled them to address some of the performance variabilities, as elaborated below.

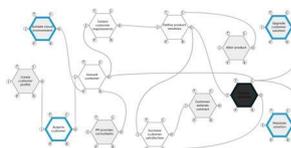


PP1 was able to dampen the performance variability in <Acquire customer> by changing their payment models. The reason for the performance variability was that PP1's product is high-priced, which required people high in the hierarchy of the customer to

approve the buy. In the cloud, it is possible to offer a subscription model, which spreads the initial price over a longer time (e.g. 12 months). By requiring customers to rent the product for a minimum of 12 months PP1 can ensure that they still get the same amount of money as before the cloud. Hence, the performance variability in <Acquire customer> can be dampened by a move to the cloud.



The performance variability in <Customer sets up product environment> existed because customers often needed too long to install necessary hardware in their data centres. Through a move to the cloud, this performance variability can be eliminated because the function became obsolete. The function is replaced by <Initiate cloud environment>. It is responsible for renting the cloud resources from the cloud provider on which the customer's solution will be installed. Therefore, PP1 is now in control of acquiring necessary computing resources



The performance variability in <Service customer> can be dampened but is replaced by a new performance variability. The reason for the performance variability was that customers sometimes failed to install updates correctly and made PP1 responsible for a faulty product. In the cloud, PP1 is responsible for installing updates. They can ensure that these are installed properly and in a timely manner. However, they have to rely on the cloud provider for their computing resources. If the cloud provider has an outage PP1's product is unavailable. Therefore, the performance variability in <Service customer> continues to exist, albeit it has a different nature.

## 6.5 Discussion

The FRAM analysis with PP1 provides two major findings. First, it shows how FRAM informs steps to increase and measure a company's resilience during the

migration of software products into the cloud. Second, it shows how FRAM can inform organisational changes necessary to increase resilience, i.e. how it is used outside of its original domain of incident investigations and safety assessments. The two major findings are explained in more detail in turn.

### **6.5.1 Actions taken by PP1 during the migration**

The Managing Director of PP1 stated during the study that the goals for adopting cloud computing were clear from the start. How to change the organisation to achieve the goals, however, was not clear from the start. Hence, PP1 adopted an ad-hoc approach in which they took necessary immediate actions to deal with the effects of cloud computing while leaving other actions for the future. PP1 managed to increase their resilience not only by dampening performance variabilities, as explained in section 6.4.4, but also by making complementary organisational changes cloud computing requires, as explained in the following.

By comparing the *before* and *after* cloud migration FRAM models, it is possible to understand what immediate actions PP1 took, that were necessary to adopt cloud computing. They took these actions to exploit the advantages of cloud computing, i.e. offer the product to customers faster and increase customer satisfaction by managing the product for them. To exploit the advantages of cloud computing the five functions highlighted in blue in Figure 27 had to change. PP1 also wanted to make the migration of their software products into the cloud as efficient as possible to save time and resources. Furthermore, they wanted to be able to retreat from cloud computing in case of adverse effects emerging. Being able to retreat in case of adverse effects was necessary for PP1 as they have only one core product from which they generate the majority of their revenue. If that product would fail in the cloud or customers would stop buying it, PP1 would quickly experience financial distress. Efficiency and the ability to retreat are the main reasons why the software vendor kept two core functions unchanged: <Consult customer> and <Increase customer satisfaction>. By keeping these two core functions unchanged during the initial migration into the cloud, PP1 reduced the uncertainty employees and customers experienced during the migration, to enhance the chances of adopting the new technology successfully. In addition, no immediate actions were necessary for these functions in order to be able

to adopt cloud computing. PP1 might change these functions in the long-term to make them more appropriate for cloud computing and exploit additional opportunities, i.e. anticipate product features customers might desire (similar to what PP5 from the multi-stage study is doing in the cloud).

By using PP1's available resources in a targeted way, i.e. choosing carefully which functions to change, PP1 was able to dampen the impact of use and technology uncertainty (see section 2.3). In the cloud, PP1 can monitor more closely how their products are being used. Monitoring product use allows them to dampen the impact of *use uncertainty*, by reacting to and anticipating customer expectations and market demands quicker. PP1 is able to see what functions of their products are being used and by whom, e.g. a manager or a technician. Thus, PP1 can customise the product for different user roles. In order to dampen the impact of *technology uncertainty*, the software vendor had to find ways to work around the loss of control to the cloud provider. PP1 reported that this can sometimes be an issue with customers as they are concerned with their data now being stored outside of their immediate control. PP1, however, managed to turn the implications of technology uncertainty into an advantage. In the cloud, it is easier to provide the product to customers and keep it up to date, enabling PP1 to increase overall customer satisfaction (despite the fact that customers give away control over their infrastructure). Customers get the latest version of the product without having to do anything themselves. To further deal with the implications of technology uncertainty, PP1 is working with a niche cloud provider, who is located close by and to whom they have direct contact (see section 5.2). PP1 knows, if something goes wrong, they can go directly to the provider and work with them (in contrast to bigger providers, like Amazon or Microsoft, where SMEs are more anonymous).

### **6.5.2 Applying FRAM to inform organisational changes**

As this work presents one of the first applications of FRAM to inform organisational changes necessary to increase resilience, it is worth discussing the approach taken with PP1 in detail. Overall, the approach that has been taken for this study provided useful results. However, further studies are necessary to prove that FRAM is indeed useful to inform organisational changes (a point chapter 8 will address). By creating

two separate FRAM models it was possible to investigate how and why individual functions changed. Investigating changes to individual functions while considering the bigger picture, i.e. the FRAM model as a whole, appears to be necessary in order to inform discussions around organisational changes. When incidents are being investigated, for example, it can be sufficient to create only one FRAM model that shows the functions necessary for everyday activities, in the way previous FRAM studies did (see section 6.3.3). Thus, by changing the steps of a FRAM analysis, this study increases the understanding of FRAM and widens its scope.

The two FRAM models were created together with the Managing Director of PP1. The description of the functions and their aspects was supplemented by data collected during the multi-stage study. Combining the FRAM analysis with existing data collected over an extended period enabled a more detailed FRAM analysis that could not have been achieved by limiting the data collection to the 2-hour interview of this study. Indeed, the FRAM analysis proved to be useful as a way of summarising the results from the multi-stage study.

The generic FRAM customer lifecycle, that has been created to structure the 2-hour interview with the Managing Director of PP1, appeared to be applicable to the pressing needs of a software vendor. The Managing Director was shown a similar lifecycle from a product development perspective with <Instantiate product environment>, <Set up product>, <Service product>, and <Enhance product functionality> (see Appendix D for a full description of the product development lifecycle). The Managing Director dismissed the product development lifecycle as they think about the impact on *customers* before making decisions. During the interview each function of the customer lifecycle was discussed and, if necessary, changed or decomposed to go into more detail. By doing this, it was possible to save time during the interview, compared to creating FRAM models on a blank piece of paper.

The FRAM analysis showed that migrating PP1's software product into the cloud does not solve all problems for PP1. Although they were able to dampen and even eliminate previously experienced performance variabilities, cloud computing exposes

PP1 to a new performance variability in <Service customer>. Currently, PP1 is using only one cloud provider to host their product for all customers. Thus, the cloud provider has become a single point of failure. If the cloud provider has an outage, PP1's product will be unavailable to all their cloud customers. PP1 is aware of this fact, however, they did not express great concern about it. The reason for this might be that their cloud provider never experienced an outage. PP1 is currently planning to use Amazon as an additional cloud provider. The exact setup is not clear yet, thus, it cannot be concluded if this will have an influence on the resilience of PP1. Chapter 2 explained that technology uncertainty makes risk management more difficult because technologies like cloud computing take away control from companies that adopt it. The above discussion shows that PP1 has difficulties reacting to the loss of control and finding appropriate measures to deal with it. Section 2.3.3 suggested a change in the mind-set of companies from *mean time between failures* towards *mean time to recovery*. Although, in theory, this change in mind-set could help PP1 in dealing with the loss of control, PP1 does not seem to apply it. Becoming more resilient, as section 2.4 explained, is a continuous process and a company can never be truly resilient (only more or less resilient). Therefore, using an additional cloud provider could be an appropriate step towards a *mean time to recovery* mind-set, thus, increasing the resilience of PP1 further.

Section 6.2 described two ways for identifying performance variabilities: a simple way where the Output of a function is affected in terms of time and precision; and a more elaborate way where performance variability are identified with the help of common conditions (CCs). The study with PP1 practiced the simple understanding of performance variabilities, which provided insights for PP1 on how to deal with performance variabilities, enabling them to increase the resilience of their company. The investigation of performance variabilities within the IT industry, however, requires further studies. The cloud FRAM model of PP1 shows, for example, that the performance variability in <Service customer> is not due to the function itself but due to its dependabilities, as the function relies on the cloud provider to produce its Output. The dependabilities are not obvious by looking at the description of the function, which states that the function assists customers in their daily use of the product. Hence, performance variabilities within the IT industry need to capture the

interdependencies between technological and organisational elements of companies (by collecting additional data chapter 8 will discuss this point in more detail). The relationship between technological and organisational elements is likely to be more complex for IT organisations than for the health care or air traffic industry. The health care industry, for example, relies on technologies for individual tasks and the technology carries out this task more independently, i.e. it does not change its behaviour based on how other technologies work, e.g. a medicine pump. Within the IT industry, however, if one technology fails it can have implications on other technologies and organisational processes, e.g. if the cloud provider has an outage customers of the software vendor cannot use its product. For future studies it could be helpful to create a list of CCs that apply to the IT industry or even software vendors (an attempt at creating CCs for the IT industry is provided in chapter 8).

While creating the two FRAM models with the Managing Director of PP1 it was difficult to decide when the two FRAM models would be complete and all performance variabilities identified. FRAM does not contain a stop signal that tells the applicant that the FRAM model is complete. This is, however, not due to FRAM but more due to the fact that FRAM is a qualitative method. Qualitative methods rely on the interpretation abilities of the applicant. Thus, FRAM will always only be as useful as the data that is provided by the users and the thoroughness of their analysis. In other words, depending on the role and responsibilities of the applicant, the FRAM models of PP1 can look very different, e.g. if a software engineer creates the FRAM models instead of the Managing Director. As this thesis aims to take a high level perspective, it was appropriate to create the FRAM models with the Managing Director and focus on those functions that they deemed appropriate. For future studies, however, it could be useful to validate the FRAM models with other people in the organisation (this was requested but not approved by PP1) or create the FRAM models with people in different roles from the start, i.e. in the form of a group discussion.

The Time aspect was the only aspect that has not been described for any of the functions in the two FRAM models. From previous studies that have used FRAM (see section 6.3.3) it is apparent that the Time aspect is being used and thus does have a

role to play. One possible explanation for the obsolescence of the Time aspect could be the domain in which FRAM is being used. In the health care or air traffic industry, two industries in which FRAM has been applied frequently, it can be important that a particular function starts at exactly 2pm or parallel to another function (e.g. when medicine pumps need to work in sync to infuse the right amount of medicine). Within the IT industry temporal aspects do not appear to be that important. For PP1, for example, it is not business critical if <Acquire customer> is carried out at 2pm or 2:01pm. For future studies in the IT industry it is dissuaded to eliminate the Time aspect from functions for two reasons. First, if FRAM is modified in such a way it is not possible to build on the existing documentation of FRAM, which is extensive compared to the documentation of similar methods, e.g. STAMP. Second, in the above study FRAM has been applied to one company in a very particular situation. Thus, the Time aspect might yet be important for the IT industry, e.g. in other situations or for different technologies.

### **6.5.3 Limitations of the FRAM study with PP1**

The findings from the FRAM analysis with PP1 need to be considered in the light of three drawbacks. First, some of the reasons why PP1 became more resilient are due to the nature of cloud computing and have less to do with the way PP1 migrated their software product into the cloud, i.e. they are an effect of cloud computing experienced by all companies that adopt cloud computing. However, in order to increase the resilience of PP1 in the long term it is important to understand how PP1 integrates the effects of cloud computing into their daily activities. The FRAM models showed in detail how PP1 integrated the effects provided by cloud computing to become more resilient.

Second, the FRAM analysis with PP1 was carried out retrospectively. In other words, by the time of the FRAM analysis PP1 had already migrated their software products into the cloud successfully. The FRAM analysis was still useful as it showed how FRAM needs to be adapted in order to be applied for migrating software products into the cloud. Furthermore, PP1 appreciated FRAM for enabling them to understand where their strengths and weaknesses are in terms of resilience. For future studies it

would be desirable to apply a FRAM analysis prior to the migration. Chapter 8 will address this point.

Third, the FRAM analysis was carried out only with PP1 from the multi-stage study. The other participants of the multi-stage study were unavailable due to time constraints. However, based on the data collected during the multi-stage study it can be concluded that the high-level functions, especially those in the customer lifecycle around which the FRAM analysis was organised, also apply to the other PPs and therefore their FRAM models are likely to show similar increases in resilience.

## 6.6 Conclusions

This chapter tested FRAM as a method to inform steps to increase and measure the resilience of software vendors migrating their software products into the cloud. The findings provide two major conclusions.

First, when FRAM is appropriately adapted, by creating a *before* and *after* cloud migration FRAM model to compare how functions change, FRAM is a suitable method to inform steps to increase and measure the resilience of software vendors migrating their software products into the cloud. The results of the FRAM analysis with PP1 show that they were able to dampen or even eliminate all of their performance variabilities they experienced before the migration of their software products into the cloud. However, the FRAM analysis also showed that cloud computing exposes PP1 to at least one new performance variability. Therefore, increasing system resilience is a continuous process. This study was very limited in its scope by applying FRAM to one software vendor. Chapter 8 will test FRAM (in a modified form, see chapter 7) with more companies to collect further evidence that confirm and validate the conclusions of this chapter.

Second, this chapter concludes that FRAM is the answer to RQ2: *How can SME software vendors that plan to migrate their software products into the cloud increase their resilience with the adaptation of capabilities affected by cloud computing and how can the increase be measured?* Software vendors planning to migrate their software products into the cloud need to adapt their functions to dampen existing

performance variabilities during the migration and to avoid the emergence of new performance variabilities. Adapting functions to address performance variabilities can ripple through coupled functions, as this chapter showed, requiring them to adapt as well, so that additional performance variabilities are avoided. Hence, adapting functions to dampen performance variabilities can have a significant influence on existing capabilities. If, for example, a function that has been adapted to dampen performance variabilities was part of a capability, the capability needs to adapt as its means have changed. Thus, companies need to reflect the changes they make to dampen performance variabilities in their capabilities. The next section introduces cFRAM to allow companies to inform the adaptation of capabilities based on the changes they make to functions. cFRAM will allow them to adapt their capabilities in a way that increases the company's resilience.

## 7 cFRAM – Planning the migration of software products into the cloud

The objective of this chapter is the development of a method, to be used in practice, that allows software vendors to adapt their capabilities to cloud computing in a way that increases the company's resilience. The method will be called cFRAM (c for capabilities) and extends FRAM through the four viewpoints that capture the underlying processes of adapting capabilities (see chapter 5). By combining the answers to RQ1 and RQ2 cFRAM provides the answer to the overarching research question: *How can companies adapt their capabilities to technological discontinuities to increase the company's resilience?*

cFRAM has four steps. In the first step data is collected to identify functions and describe aspects that represent the current way of doing business (i.e. creation of a *before* cloud migration FRAM model). In the second step, more data is collected to describe functions and aspects in more detail. Furthermore, performance variabilities of functions are identified. In the third step the existing capabilities of the company are identified and described by analysing functions and their resources with the four viewpoints. In the fourth step, the move to the cloud is planned by adapting the functions from the *before* cloud migration FRAM model to accommodate cloud computing and, if possible, dampen performance variabilities (i.e. creation of an *after* cloud migration FRAM model). Furthermore, the impact of the *after* cloud migration FRAM model on existing capabilities is analysed to inform their adaptation or the development of new capabilities.

This chapter is structured as follows. After laying out the motivations for relating FRAM to capabilities (see section 7.1) the intended users of cFRAM are described to identify requirements cFRAM needs to satisfy (see section 7.2). Afterwards the theoretical model and assumptions of cFRAM are explained (see section 7.3). The second half of this chapter explains the steps of a cFRAM analysis in detail and illustrates them through examples (see section 7.4). This chapter concludes by introducing a handbook in which cFRAM is explained to allow the application without an expert being present (see section 7.5).

## 7.1 Motivation for relating the FRAM to capabilities

FRAM was used as the basic method to investigate capabilities when migrating software products into the cloud for four reasons. First, functions in FRAM are similar to capabilities. Functions within FRAM contain everything necessary to carry out a task. Similar to capabilities, functions can contain tangible and intangible elements, such as machines, documents, personnel, etc. In other words, both capture organisational routines. The only difference between functions and capabilities is that one function can contain several capabilities and one capability can reside in several functions, i.e. n-to-n relationship.

Second, both FRAM and capabilities focus on what a system does rather than how it is structured. By focusing on what a system does, tangible *and* intangible organisational elements can be captured. Furthermore, emergent behaviour can also be captured and investigated (Hollnagel 2012b). In order for companies to become *more* resilient necessary activities have to be made an inherent part of everyone's daily activities. Thus, a higher level of resilience can only be achieved by what people do (where the structure of the system supports them in doing it, Roberts 1990).

Third, with the analysis of functional resonance the concept of resilience is already built into FRAM. Chapter 6 successfully tested FRAM for its ability to inform steps to increase and measure resilience. Through the identification of performance variabilities that exist in functions, software vendors can analyse how they might want to adapt functions during the migration into the cloud in order to eliminate or dampen them. To increase their resilience in the long-term, however, software vendors have to reflect the adaptations to functions in their capabilities.

Fourth, FRAM is flexible and easy to use. Hence, it does not require a lot of training and allows software vendors to focus only on the organisational challenges that are relevant to their circumstances. Through the identification and description of functions, software vendors can concentrate on investigating those functions in more detail that they deem important. For all remaining functions, background functions can be described that can be investigated in more detail at a later stage of the adoption.

## **7.2 Intended research users and their requirements**

cFRAM builds on the definition of the research users and requirements of the original FRAM but defines these more specifically so that they are applicable to software vendors that want to migrate their software products into the cloud. cFRAM targets Managing Directors or leaders of Product Development of SME software vendors that want to explore the possibility of distributing their software products via the cloud. From this goal, the following research users can be defined.

In the cloud, capabilities reside in a network rather than a single organisation (see chapters 4 and 5). cFRAM takes that into account by enabling the analysis of complex systems where dependencies between different actors of a system exist (similar to that of the original FRAM). Software vendors that migrate their software products into the cloud need to be aware of the dependencies between them, the cloud provider, and the end-customers. Otherwise, they might offer features to customers that are not supported by the cloud provider. cFRAM shows which functions of the software vendor have a connection with the cloud provider and which with the customer.

The following requirements for cFRAM emerge. To understand the complementary organisational changes cloud computing requires it is important to consider the viewpoints of different departments. Hence, the method needs to be applicable during group discussions with people from different departments where the Managing Director or leader of product development would take over the role as moderator and navigate through the steps of a cFRAM analysis. cFRAM should assist software vendors in answering organisational questions about the adoption of cloud computing. Technical difficulties of migrating software products into the cloud, e.g. the partition of databases, might be outside the scope of cFRAM.

## **7.3 Theoretical model and underlying assumptions**

Developing a method that assists software vendors in adapting their capabilities to cloud computing in a way that makes their company more resilient requires the definition of an underlying model and its assumptions. The model and assumptions

encapsulate a simplified version of the way software vendors deal with cloud computing. It is important for users of the method to understand the model and its assumptions as they describe what the method can do *and* what its limitations are (Hollnagel 2012b).

The ability to adapt capabilities to cloud computing is not built into FRAM. In order to extend FRAM to inform the adaptation capabilities, the steps of a FRAM analysis will be changed while retaining the original elements of a FRAM model, i.e. functions and aspects. Only changing the steps of a FRAM analysis lets users of the method build on the extensive documentation of the original FRAM.

Although functions and capabilities are abstractions to capture organisational routines, functions are abstractions on a lower level than capabilities. Because there is a difference in the layer of abstraction functions and capabilities have an n-to-n relationship, i.e. one capability can reside in many functions and one function can contain many capabilities. When planning the migration into the cloud, software vendors will want to aim to adapt those capabilities that are going to be affected by a migration into the cloud (positively or negatively) and that reside in more than one function, i.e. core capabilities.

Using cFRAM to adapt capabilities to cloud computing makes it necessary to add steps to a FRAM analysis after the functions and performance variabilities for the current way of doing business have been identified (i.e. after the creation of the *before* cloud migration FRAM model). The steps that are added assist software vendors in abstracting from individual functions and their resources to tasks that several functions, together, aim to achieve. cFRAM abstracts from functions and their resources through the framework with the four viewpoints (cultural, management, application, and governance).

The four viewpoints are the result of the multi-stage study that investigated software vendors during the migration of their software products into the cloud from a systemic perspective (see chapters 4 & 5). They capture the underlying processes of adapting capabilities to cloud computing. Thus, by being aware of the viewpoints, software

vendors are better able to accommodate the different factors that *influence* the adaptation of their capabilities to cloud computing.

By incorporating the four viewpoints into cFRAM they are adapted so that they can be used to *inform* the adaptation of capabilities to cloud computing. Informing the adaptation of capabilities is done in two steps. First, the viewpoints are used to identify existing capabilities within a software vendor. Second, the viewpoints are used to inform the adaptation of existing capabilities to cloud computing by allowing software vendors to react to and anticipate the critical success factors of cloud computing that the four viewpoints capture.

To identify the existing capabilities of a software vendor, i.e. step 3 of cFRAM, the resources of functions that show the current way of going business (*before* cloud migration FRAM model) need to be listed. The focus is on resources of functions as, according to the definition of capabilities, capabilities combine different *resources* in a structured way to achieve a specific task (see section 2.1). The resources can either come from other functions that are connected through the Resource aspect of a function, or they can be internal resources such as people, documents, machines, etc. After listing the resources of a function, they are assigned to one of the four viewpoints depending on the factors that influence or constrain the use of the resource. The use of a resource called ‘customer history of support issues’ is, for example, influenced by the management viewpoint, as the resource needs to be managed in order to inform the development of features (support issues between customers need to be compared, the feasibility of new product features needs to be discussed, etc.). In other words, just having the resource is not sufficient. Identifying the influences and constraints across resources of functions will reveal the capabilities that are going to be affected by cloud computing (section 7.4.3 below will go into more detail with additional examples).

To inform the adaptation of existing capabilities when planning the migration into the cloud, i.e. step 4 of cFRAM, it is necessary to repeat step 3 of cFRAM. Instead of using the functions and resources of the *before* cloud migration FRAM model, software vendors use the *after* cloud migration FRAM model that shows how they

plan to adapt functions and resources to accommodate cloud computing. After having listed the resources available in the cloud and having assigned them to one of the four viewpoints, software vendors can compare the resources and viewpoints with those of the *before* cloud migration FRAM model. The differences between both lists inform the adaptation of capabilities. In very general terms, the higher the number of differences between both lists, the more likely it is that existing capabilities are inappropriate for cloud computing. Thus, they require adaptation or entirely new capabilities need to be developed.

The following list suggests what kind of resources are often related to what viewpoint

- Cultural: Often plays a role for resources where people are involved. People can be the resource (or part of the resource) or people can be affected by the use of a resource.
- Management: Often plays a role when the use of software needs to be coordinated with other resources, e.g. people or business processes.
- Application: Is likely to play a role for resources that are part of product development or distribution.
- Governance: Often plays a role where documents are involved, which describe how a task needs to be carried out.

#### **7.4 Steps of a cFRAM analysis**

A cFRAM analysis has four steps. Step 3 and 4 of cFRAM contain several sub-steps as the following listing shows:

1. Collecting data to identify functions
2. Describing the main functions and identifying performance variabilities
3. Identifying existing capabilities
  - 3.1. Identify the three most important functions
  - 3.2. Define the resources of the functions and assign them to one of the four viewpoints
  - 3.3. Identify capability (or capabilities)
  - 3.4. Extrapolate capabilities to other functions (or identify additional capabilities)
4. Planning the move to the cloud

- 4.1. Adapt functions to the cloud
- 4.2. Investigate impact of adaptations on performance variabilities
- 4.3. Investigate impact of adaptations on existing capabilities and inform development of new capabilities

In the following, each step of a cFRAM analysis will be explained in more detail.

#### **7.4.1 Step 1: Collecting data to identify functions**

The data collection of a cFRAM analysis was designed to be flexible. The guidance proposed in this step has been informed by the findings from chapters 4, 5, and 6, and the original FRAM handbook (Hollnagel et al. 2014). As this step is very similar to what has been done in section 6.4.1 with PP1, the focus will be on explaining how software vendors can apply step 1 without an expert of FRAM being present. It should be regarded as a suggestion as some software vendors might find that it does not apply to them, depending on their business model and products.

Within SME software vendors, everyone is part of product-development, -distribution, or -support. Therefore, a group discussion is suggested to start the data collection. Although group discussions can be more difficult to organise for companies and one person might overshadow others' opinions, the advantages outweigh the drawbacks (Cabrerizo et al. 2010). In a group discussion people from various departments can take part and offer their views (Chosokabe et al. 2015). Different views are necessary to identify the organisational changes cloud computing requires, e.g. in the areas of HR, Sales & Marketing or Finances. It is recommended that the form of a group discussion be maintained for all stages of a cFRAM analysis.

The following list is only a suggestion of departments that could offer valuable input for the data collection:

- Sales & Marketing
- Software development
- Technical infrastructure
- Support

- Other roles or departments depending on the kind of software product or services being sold

There should be a moderator to guide the discussion and to ask opening questions. The job of the moderator is also to encourage everyone to speak openly, not only about successes but also failures. The moderator should ideally be someone in a high-level position who has a good overview of the different departments and product areas. It could be, for example, the Managing Director or the leader of product development.

The following high-level questions are suggested to start a discussion to identify the functions of the FRAM model that represents the current way of doing business (i.e. *before* cloud migration FRAM). The questions are based on the generic customer lifecycle that has been successfully tested in chapter 6:

1. How do we acquire new customers?
2. How are new customers set up to use our products?
3. How are users supported in their everyday use of our products?
4. How do we achieve long-term customer satisfaction?

While the proposed questions are being discussed, it is important to focus on the identification of functions that are necessary for everyday activities, and order all other information around these functions. At this stage, the group should agree on the name of functions (should be a verb or verb phrase), their descriptions (which should also include the organisational role performing the function), and the definition of Input and/or Output aspects.

#### **7.4.2 Step 2: Describing the main functions and identifying performance variabilities**

The following questions aim to go into more detail than the high-level questions of the previous section, to define aspects of the main functions, identify background functions and identify potential performance variabilities:

Regarding question 1: How do we acquire new customers?

- i. How do we advertise our products? How do we find potential customers?

- ii. How do we contact potential customers?
- iii. How do we demonstrate our products?
- iv. How long does the process take until a potential customer becomes an actual customer?

Regarding question 2: How are new customers set up to use our products?

- i. Who is involved on our side?
- ii. Who is involved on the customer's side?
- iii. How long does it take to set up a new customer?

Regarding question 3: How are users supported in their everyday use of our products?

- i. Who in our company is responsible for user support?
- ii. How are bugs reported and fixed?
- iii. How are our product(s) updated? Are they updated on a regular basis?

Regarding question 4: How do we achieve long-term customer satisfaction?

- i. How are new customer requirements implemented?
- ii. How do we develop new features/products?
- iii. Who in our company is responsible for long-term customer satisfaction?

The above listed questions are not exhaustive and users of cFRAM are encouraged to think about additional questions that might be appropriate for their particular circumstances or products.

When all functions and aspects have been identified and described, potential performance variabilities can be identified. The underlying theoretical model of performance variabilities has been explained in section 2.4.2 and the steps to identify performance variabilities have been explained in section 6.2 and illustrated through examples in section 6.4.2. As part of the group discussion, it should be discussed, for each function, if there is potential performance variability in the production of the Output (in terms of *time* and *precision*) and how this might affect downstream functions. Not only those performance variabilities that occurred in the past should be identified but also potential ones. The performance variabilities will be important for the fourth step of cFRAM (Planning the move to the cloud). The aim is to dampen the identified performance variabilities through the migration of software products into the cloud, to increase the overall resilience of the software vendor.

### 7.4.3 Step 3: Identifying existing capabilities

Existing capabilities of a software vendor are identified on the basis of the *before* cloud migration FRAM model and the four viewpoints (see section 7.3). The four viewpoints (cultural, management, application, and governance) will assist software vendors in understanding how the resources of functions are influenced or constrained. Table 5 provides a short overview of the sub-steps carried out in the third step of a cFRAM analysis.

*Table 5 - Overview of the sub-steps to identify existing capabilities*

<b>Sub-step</b>	<b>Purpose</b>
1. Identify the three most important functions	cFRAM is interested in core capabilities from which the company derives competitive advantages. Core capabilities reside within several functions.
2. Define the resources of the functions and assign them to one of the four viewpoints	Define the purpose of the three most important functions and list their resources. Then investigate how the resources are influenced or constrained by assigning them to one of the four viewpoints.
3. Identify capability (or capabilities)	Identify the influences and constraints across resources of the three most important functions to reveal the capabilities that are going to be affected by cloud computing.
4. Extrapolate capabilities to other functions (or identify additional capabilities)	Investigate if identified capabilities are appropriate for other functions in the FRAM model. Otherwise, repeat the above steps.

The three most important functions of the FRAM model need to be identified in order to start the identification of capabilities. The reason for focusing on three functions is that a cFRAM analysis is about adapting core capabilities and not all capabilities that might exist. Furthermore, it is important to investigate the impact of cloud computing on those capabilities from which the software vendor derives competitive advantages. Trying to identify the capabilities of all functions can become complicated depending on the number of functions in a FRAM model. The capabilities of other functions, i.e. those not part of the three most important ones, can be identified in later stages by repeating the sub-steps explained in this section. When identifying the three most important functions, background functions should be excluded. They should be excluded because they generally do not have a large impact on the FRAM model as a whole and thus are unlikely to contain core capabilities.

After identifying the three most important functions, the resources for these functions need to be listed and assigned to one of the four viewpoints depending on the factors that influence or constrain the resources (Table 6 is included in the cFRAM handbook to help software vendors with the viewpoints). By identifying the influences and constraints that span across resources and functions, software vendors reveal their capabilities by summarising how they exploit resources and minimise their constraints. The identification of capabilities is best illustrated by an example. Step 3 of cFRAM has been applied to PP1 based on the data collected in chapter 6 and the multi-stage study. The results of the application of step 3 (and step 4, see further below) have also been shared with PP1 for feedback.

*Table 6 - Description of the four viewpoints*

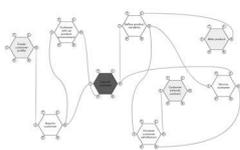
<b>Viewpoint</b>	<b>Description of viewpoint and how to identify it</b>
Cultural	<p>Captures resources that either deal with soft issues of the software vendor's customers or their own employees.</p> <p>The two groups are often affected by organisational changes, such as adopting a new technology, as business processes might change and they might have to learn new skills.</p> <p>The cultural viewpoint is also about the internalised rules and norms of behaviour that employees (and customers) follow. These are not necessarily written down and employees learn over time and from interactions with other employees what actions are right and wrong.</p> <p>To develop and sell software products, the software vendor also needs to be familiar with the customer's culture.</p>
	<p>Often plays a role for resources where people are involved. People can be the resource (or part of the resource) or people are affected by the use of a resource.</p>
Management	<p>Captures resources around coordinating everyday tasks. These tasks can be internal, e.g. coordinating feature development or creating a roadmap for product development, but they can also be external, e.g. supporting the customer in using the product or increasing customer satisfaction.</p> <p>The management viewpoint also plays a particular role when it comes to adopting a new technology, like cloud computing. In this case, the management viewpoint captures issues around the adoption process of the technology, e.g. which decisions need to be made during the adoption, and the lifecycle of the technology.</p>
	<p>Often plays a role when the use of software needs to be coordinated with other resources, e.g. people or business processes. It is about finding the right balance between exploiting what the software can offer and retaining elements other resources require.</p>
Governance	<p>Shows how the software vendor has to adhere to governmental or institutional laws and regulations and corporate policies. The software vendor needs to make sure whether what they would like to do satisfies these laws and corporate policies.</p> <p>They might also have to be aware of potential customer requirements and</p>

	<p>policies. This can be particularly important for certain industries that are highly regulated, e.g. air traffic management, Oil &amp; Gas, or health care. In contrast to the cultural viewpoint, which captures the informal rules and norms of behaviour, the governance viewpoint captures the formal rules and norms of behaviour.</p> <p>Difficult to deal with as it cannot be easily changed and some of it might be out of the control of the software vendor or the customer. That is why The governance viewpoint, in contrast to the other viewpoints, often only constrains what can and cannot be done.</p>
Application	<p>Captures resources around developing and distributing software. Companies have to be aware of the dependencies that can enhance or stifle the development of their own product and to avoid unnecessary risks and cascading failure events. The application viewpoint also captures issues around the daily work of the software developers, e.g. what skills they need or have to develop and what tools to use for product development.</p> <p>The application viewpoint can also exist in relation to the software vendor's mission in order to decide what features will be developed or how customer feature requests will be handled.</p> <p>Is likely to be at the centre of many resource constraints. It is likely to be found in connection with resources of the products the software vendor develops or the use of them.</p>

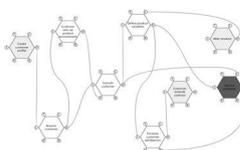
#### 7.4.3.1 Example of applying step 3 of cFRAM

The three most important functions from the *before* cloud migration FRAM model shown in Figure 26 (see section 6.4.2) are (sub-step 1):

- Consult customer
- Service customer
- Increase customer satisfaction

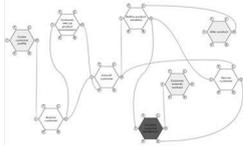


<Consult customer> is one of the most important functions as PP1 makes a large share of their revenue from consulting services that go together with the sale of their software product. When customers buy PP1's product, it comes in a very generic form and needs to be tailored to the customers' needs. Consultants from PP1 tailor the installation by going to the customer and examining their business processes and other information.



<Service customer> is one of the most important functions as it is responsible for supporting the users of the customer during everyday activities. In other words, this function is responsible for

ensuring that customers can use the product in the way they need to. If customers are unhappy with the support, bugs are not being fixed fast enough, or requested features are not introduced, the customer might acquire a different product.



<Increase customer satisfaction> is one of the most important functions as it is responsible for ensuring long-term customer satisfaction. This function works in tandem with <Service customer>. Whereas <Service customer> is more about short-term satisfaction of users, <Increase customer satisfaction> is responsible for renewals of licenses and the sale of new products or upgrades, for example. Thus, this function tries to ensure on-going revenue from customers.

From the description of the three most important functions, it becomes clear that all of them work towards the same overall goal. The overall goal is enabling the users of a customer to use the software product in the way they require to on a daily basis.

To further understand the overall goal, the resources of these functions have been listed and assigned to one of the four viewpoints depending on how the resources are influenced or constrained (sub-step 2). Table 7 shows a listing of the functions, their resources, and to which viewpoint they have been assigned.

All four viewpoints are represented in Table 7. *Application* (6 resources) and *Cultural* (3 resources), however, appear more often than *Management* (2 resources) and *Governance* (2 resources). Indeed, if considering the overall goal the three functions aim to achieve, the number of resources for each viewpoint seems plausible. *Application* is by far the most represented viewpoint as PP1 aims to enable customers to use their product. Hence, *Application* influences the offer to customers in terms of functionality and product usage. *Cultural* is the second most represented viewpoint, as PP1 might have to adapt to the customer's needs and behaviour. That will influence the actions and behaviour of PP1. *Management* is less often represented but still of importance as the three functions need to be managed and coordinated (since they are

all working towards the same goal and <Service customer> and <Increase customer satisfaction> can only start if <Consult customer> has produced its Output). *Governance*, as the final viewpoint, is of importance as a control instrument since the functions (and employees of PP1) are constrained by the product of PP1 in terms of what they can offer customers.

With the categorisation of the resources into viewpoints it is possible to identify a core capability PP1 possesses: *service delivery management* (sub-step 3). PP1 has to deal with three overall issues that make up the core capability. First, all three functions' objective is the provision of the product the customer needs. Initially one might therefore call the capability product delivery management. The description of the functions and their resources show, however, that PP1 provides more than the software product, i.e. consulting services and support—*service delivery management* capability. Second, all three functions are organised around delivering and enabling the customer to use the product. Furthermore, the functions customise the software product for the customer and continually update it—*service delivery management* capability. Third, all three functions have to be managed as they need to be coordinated and procedures need to be followed. Furthermore, the continued communication with the customer needs to be managed—*service delivery management*.

The identified core capability has some applicability to other functions in the *before* cloud migration FRAM model, e.g. <Collect requirements> and <Define product variables> (sub-step 4). The other two functions are not appropriately covered by the capability. The function <Customer sets up product environment> contains no capability, as the customer carries out this function and PP1 has only a very limited amount of control over it. The function <Acquire customer> is a sales function and some of it is outsourced to a third party that creates potential customer profiles (see description of function in chapter 6). As it is the only function for sales it is more appropriate to keep the function instead of identifying a capability on top of it.

Table 7 - Showing the functions with their resources and related viewpoints

Function	Resource	Viewpoint	Explanation
----------	----------	-----------	-------------

<Consult customer>	Consultants	Cultural	Consultants are confronted with the culture of the customer, which influences their proposed solutions. Furthermore, they are also influenced by the unwritten rules of PPI.
	The software product	Application	The product is influenced by <i>Application</i> as it determines what the product can and cannot do and thus what the consultants can offer the customer. It might also not be technically feasible or desirable to develop every feature a customer requests.
	Procedure documentation	Governance	Procedure documentation is influenced by <i>Governance</i> as the documents inform about corporate policies and formal rules of behaviour that must be followed.
	Product documentation	Governance	See Procedure documentation
<Service customer>	Support personnel	Cultural	See Consultants
	Help desk software	Management	Help desk software is influenced by <i>Management</i> , as it depends on how the help desk software was chosen, how it is integrated into the company, and how it is being used.
	Software developer	Application	Software developers are mainly influenced by <i>Application</i> as it depends on how the software has been developed in the past, how it utilises other software products, e.g. through API's, and what skills the developers have.
	The software product	Application	See above
	Software product updates	Application	See The software product and Software developer
<Increase customer satisfaction >	Sales people	Cultural	See Consultants and Support personnel
	Software developer	Application	See above
	The software product	Application	See above
	Customer history	Management	Customer history describes issues customers had to deal with in the past, e.g. support requests they had. Therefore, it informs <Increase customer satisfaction> about what customers like and do not like.

#### 7.4.4 Step 4: Planning the move to the cloud

The previous section investigated the functions and resources of the *before* cloud migration FRAM model to identify existing capabilities. At this point, the software vendor should have a good understanding about what they can and cannot do. In this section, this knowledge is used to plan the move to the cloud. It is investigated how the functions in the *before* cloud migration FRAM model need to adapt to accommodate cloud computing. Furthermore, it will be concluded what steps need to be taken to dampen performance variabilities, if the existing capabilities are likely to enhance or stifle the changes to the functions and, if necessary, what new capabilities need to be developed. Table 8 provides an overview of the sub-steps carried out in this section.

Table 8 - Overview of the steps for planning the move to the cloud

Sub-step	Purpose
1. Adapt functions to the cloud	Go through every function of the FRAM model and investigate the impact of cloud computing by discussing changes in responsibilities. If necessary, introduce new functions to respond.
2. Investigate impact of adaptations on performance variabilities	Conclude if the performance variabilities can be dampened through the adaptations of sub-step 1 or if further adaptations are necessary.
3. Investigate impact of adaptations on existing capabilities and inform development of new capabilities	List resources of functions in the cloud and assign to one of the four viewpoints. Afterwards compare the list with the list of step 3 of cFRAM to inform the adaptation of capabilities.

To adapt functions to the cloud and create the *after* cloud migration FRAM model, software vendors need to investigate how functions in the *before* cloud migration FRAM model need to adapt to accommodate cloud computing (sub-step 1). When investigating if functions have to change, can change or do not need to change, the resources of functions can assist. If, for example, a resource becomes unavailable in the cloud or resources are added, the related function(s) have to change. If resources are not affected, but software vendors see a potential to exploit existing resources differently, the function could change (now or at a later stage) to enhance the capabilities in the cloud. Otherwise, there is no immediate need to change a function for moving to the cloud.

The software vendors should also investigate how the adaptations will affect the performance variabilities that were identified in the *before* cloud migration FRAM (sub-step 2). Cloud computing can be used as an opportunity to eliminate or dampen performance variabilities. The options that software vendors have in doing so depend on the nature of the performance variability. Further adaptations of functions might be necessary to eliminate or dampen performance variabilities or to avoid new performance variabilities. When carrying out sub-step 2, it is important to consider that adapting a function to dampen performance variabilities can have an impact on coupled functions. Thus, adaptations can ripple through the FRAM model. When adapting functions, whether it is necessary or deemed appropriate to dampen performance variabilities, software vendors need to investigate the wider consequences. Otherwise they risk decreasing their resilience, instead of increasing it.

When all functions have been adapted to the cloud, software vendors need to investigate how the adaptations will enhance or stifle existing capabilities (sub-step 3). Software vendors should ask themselves if the capability is able to work without the functions and resources that are being eliminated through a migration into the cloud or if the capability is able to integrate any new functions and resources. Furthermore, software vendors are encouraged to ask themselves how capabilities need to be adapted in order to dampen performance variabilities further. To answer these questions, it is necessary to list the resources of the functions in the *after* cloud migration FRAM model and assign these to the four viewpoints in the same way as in step 3 of cFRAM. Software vendors can start with the three most important functions identified in step 3 of cFRAM. If one of the three most important functions becomes obsolete through the cloud, then it is sufficient to start the analysis of capabilities with the two remaining functions. By comparing this list with the list that was created in step 3 of cFRAM for the *before* cloud migration FRAM model software vendors can inform the adaptation of their capabilities (see below). Afterwards, the step should be repeated for other functions that changed (or functions that are new) by investigating if the adapted capability is appropriate for them or if new capabilities need to be developed. It is, therefore, ensured that all functions in the cloud FRAM model are investigated to inform the adaptation of capabilities.

To guide the adaptation of capabilities, software vendors moving to the cloud should investigate in particular changes to responsibilities. Does the software vendor outsource any responsibilities to the cloud provider? If yes, related resources are likely to be outsourced too, which can make existing functions and capabilities obsolete. Furthermore, does the software vendor take over responsibilities from the customer? If yes, new resources are likely to be introduced which may require the company to introduce new functions and develop new capabilities. Existing capabilities might need to be adapted for either of (or all) of the following three reasons:

1. Functions have been adapted that included the introduction of new resources or elimination of existing resources.
2. New functions have been introduced or existing ones have been eliminated.
3. The viewpoint of a resource changes, i.e. other factors influence or constrain the use of a resource in the cloud, e.g. from cultural to management.

#### *7.4.4.1 Example of applying step 4 of cFRAM*

To illustrate step 4 it has been applied to PP1 in a similar way as step 3 (see above). For step 4 the *after* cloud migration FRAM model that is shown and explained in section 6.4.3 forms the basis. With the *after* cloud migration FRAM model it is possible to investigate if the capability of PP1 (service delivery management) needs to adapt to be appropriate for cloud computing or if it is necessary to develop new capabilities. The focus of the section below is on sub-step 3 (Investigate impact of adaptations on existing capabilities and inform development of new capabilities) as sub-step 1 (Adapt functions to the cloud) and sub-step 2 (Investigate impact of adaptations on performance variabilities) have already been explained in sections 6.4.3 and 6.4.4.

To understand the impact of the changes on the service delivery management capability, PP1 needs to investigate if the capability is likely to enhance or stifle the adaptations of functions in the *after* cloud migration FRAM model. Section 6.4.3 showed that of the functions in which service delivery management resides (<Consult customer>, <Service customer>, and <Increase customer satisfaction>) only <Service customer> has to change for a move to the cloud. In the cloud, <Service customer>

takes over responsibilities from the customer to maintain their solution and install updates, etc. The existing capability, service delivery management, however, is only responsible for customising and delivering the product to the customer’s needs and supporting the customer during everyday activities. In other words, the capability is more about soft than technical skills. The two new functions that have been introduced, <Maintain solution> and <Upgrade customer solution>, require more technical than soft skills. PP1 developed a new capability that resides in <Service customer>, <Maintain solution>, and <Upgrade customer solution> and is responsible for managing the cloud environment. Table 9 lists the resources of the three functions and their assignment to the viewpoints to inform the development of the new capability. According to Table 9 the new capability needs to be responsible for acquiring and integrating cloud resources and other cloud-based services into the company. It is also responsible for releasing the resources and services when they are no longer needed. Therefore, the new capability will be called *cloud service management* (sub-step 3).

The service delivery management and cloud service management capability have a close relationship (as both reside in <Service customer>). The service delivery management capability needs to inform the cloud service management capability about the requirements for new cloud resources and services, and the cloud service management capability is then responsible for acquiring these. Only if both capabilities work in sync can the software vendor provide the products and services customers need.

Table 9 - Summary of changes that have an impact on capabilities

Adapted functions	Changes to resources	Viewpoint	Explanation
<Service customer>	Support personnel	Changes from Cultural to <i>Management</i>	Personnel has to take over responsibilities from customer to manage the product installation
	The software product	Application	Back-end of the product changes which impacts the responsibilities the function has to fulfil
	Software product updates	Changes from Application to <i>Management</i>	Instead of the customer the PP is now responsible for installing updates

<Maintain solution>	Cloud environment	Management	PP1 is responsible for managing the computing resources on which the product is executed
	The software product	Management	PP1 is responsible for managing the installation of every customer and offer a reliable operation
	Procedure documents	Governance	Documents should describe how the tasks of managing the cloud environment are to be carried out
<Upgrade customer solution>	Software product updates	Management	PP1 is responsible for updating every customer solution properly and in a timely manner
	Procedure documents	Governance	Documents should describe when and how updates are installed for a customer so that every customer gets the same service

#### 7.4.5 Using the cFRAM as a long-term planning method

Applying cFRAM throughout the migration allows software vendors to track their progress of the migration and react to or anticipate changes faster. The advantage of applying cFRAM continuously during the migration is that it makes software vendors aware of systemic changes that cloud computing might require. In other words, if we change customer acquisition, does customer support need to change in a similar way to provide customers with a coherent experience?

To use cFRAM throughout the migration two steps are recommended. First, software vendors will need to go through the questions proposed in the first two steps to collect the data to create a FRAM model several times, to show the latest functions and performance variabilities during the migration process. Some of the questions may need to be changed in order to accommodate organisational changes that have been carried out successfully. After having created the FRAM model that shows the functions and performance variabilities in the current stage of the migration, software vendors should compare the FRAM model to the FRAM model from step 4, i.e. compare status quo with the desired state. Then the software vendor can track the progress of the functions they have already adapted and those they still need to adapt.

The results from this comparison can inform the next steps of the migration, e.g. which function to adapt next.

How often cFRAM should be applied throughout the migration and in what intervals depends on the software vendor and their products. The multi-stage study has shown that for some software vendors it takes a few months to move to the cloud, whereas for others it takes several years. Software vendors may want to apply some of the steps of cFRAM before and after a major organisational change. By applying it before an organisational change, they can understand what functions and couplings they have at the moment, to work out how they need to be changed to accomplish the organisational change. By applying it afterwards, they can use cFRAM as a control instrument to check if the organisational changes have been carried out as planned.

## **7.5 cFRAM handbook**

In order to allow companies and academics to apply cFRAM and to allow for a dissemination of cFRAM, a handbook has been written that explains the steps of a cFRAM analysis and illustrates every step with examples. Allowing an autonomous application of cFRAM is important as people can learn the method alone and thus apply it alone. This is likely to increase the usefulness of cFRAM as, after learning it once, they can apply it again and again.

The examples in the handbook are based on the FRAM analysis with PP1 (see chapter 6). The handbook is structured as follows. The introduction chapter lays out the motivation for developing cFRAM and makes the reader aware of the organisational changes cloud computing requires to adopt it successfully (similar to the introduction page of this thesis). The second chapter introduces and explains FRAM (in its original form). The third chapter lays out important preliminary information about cFRAM such as an overview of the steps, a description of the generic customer lifecycle that was developed and tested in chapter 6, and a description of PP1 as they were used for the examples. The following four chapters explain every step and how they should be carried out in detail (similar to sections 7.4.1 - 7.4.4). Every chapter ends with an application of the step to PP1 to illustrate it. The handbook concludes with a chapter that lays out how cFRAM can be used in the long-term (similar to section 7.4.5).

## 7.6 Conclusions

This chapter introduced cFRAM to allow software vendors to plan the migration of their software products into the cloud by investigating the impact of cloud computing on their resilience and capabilities. The goal, by applying cFRAM during the migration process into the cloud, is to inform the adaptation of capabilities in a way that increases the company's resilience.

A cFRAM analysis provides the answer to the overarching research question of this thesis (*How can companies adapt their capabilities to technological discontinuities to increase the company's resilience?*) through four steps software vendors that plan to migrate their software products into the cloud need to carry out. In the first and second step functions are identified, described, and connected through the six aspects and performance variabilities are identified to create a FRAM model showing the current state of doing business, i.e. *before* cloud migration FRAM. In the third step, core capabilities that reside within the company are identified. The identification of core capabilities is done through the four viewpoints that reveal how resources of functions are influenced or constrained. In the last step the move to the cloud is planned by adapting the functions in the *before* cloud migration FRAM model to make them suitable for cloud computing. At the same time, the implications of the adaptations are analysed in terms of performance variabilities and capabilities, i.e. to suggest how performance variabilities can be dampened and how capabilities need to be adapted to increase the company's resilience.

A handbook has been written that explains the steps of a cFRAM analysis similar to the explanations provided in this chapter. The handbook has been designed specifically for the industry to allow companies to use cFRAM without an expert being present. cFRAM and its handbook require evaluation in order to identify its strengths and weaknesses. The evaluation is carried out in the next chapter.

## **8 Evaluation of cFRAM**

The objective of this chapter is the evaluation and improvement of cFRAM and its handbook with 14 companies. An evaluation study of cFRAM is necessary to prove that cFRAM achieves its goal of enabling companies to adapt their capabilities to cloud computing in a way that increases the company's resilience. Furthermore, the evaluation study aims to identify the strength and weaknesses of cFRAM, validate the results it produces and understand if there is a need for such a method in the industry.

This chapter is structured as follows. First, the methodology of the cFRAM evaluation is explained (see section 8.1). Afterwards the results of the evaluation are presented and discussed (see section 8.2). This chapter concludes by describing how cFRAM and its handbook have been improved based on the findings of the evaluation study (see section 8.3 and Appendix E for the final version of the cFRAM handbook).

### **8.1 Methodology**

The evaluation study applies a cFRAM analysis to two types of companies: (1) software vendors that plan to migrate their software products into the cloud and (2) companies that plan to use cloud-based software products. A case study approach was chosen to evaluate cFRAM and its handbook (Eisenhardt 1989; Pan & Tan 2011). A case study approach allows the application of a cFRAM analysis with each participant individually to understand if cFRAM is applicable in different situations and environments. Understanding the applicability of cFRAM in different situations and environments is a key objective as the multi-stage study showed that all companies are different and that there is no rigid approach that can be followed to adopt cloud computing. Thus, companies need to be provided with a method that they can adapt to their situation and individual challenges. Furthermore, a case study approach allows the interviewer to engage in deeper discussions with the participants to evaluate the individual steps and elements of a cFRAM analysis. Other types of studies, e.g. surveys, are inappropriate to test cFRAM, as it does not allow the application of a cFRAM analysis in different situations and environments. It would only allow the hypothetical questioning of how useful participants think cFRAM might be. Thus, a survey would neither produce strong results nor suggest areas for improvement. An

ethnographic study, in principal, would have been desirable, as it would allow the researcher to embed with a company to apply a cFRAM analysis over a longer period of time (instead of applying it once during the interview). Thus, it would have been possible to collect more in-depth data and understand how cFRAM can inform individual adoption steps of cloud computing. In reality, however, an ethnographic study is infeasible due to reasons already stated for the multi-stage study (see section 3.2). Ethnographic studies are very time and resource intensive. Furthermore, it requires a strong commitment from the participating companies. It is also not possible at the outset of the study to estimate how long it will take a company to move to the cloud, and therefore how long cFRAM needs to be used. Overall, a case study approach provides the best compromise between collecting data for different situations and environments and getting an in-depth understanding of the strength and weaknesses of each step of a cFRAM analysis to conclude if cFRAM achieves its goal. For the future, however, an evaluation of cFRAM in the form of ethnographic studies is desirable.

Between May 2015 and August 2015 cFRAM and its handbook were evaluated with 14 companies (in the following referred to as Evaluation Partners, or EPs,) which are introduced in Table 10.

*Table 10 - Description of participants that were part of evaluating cFRAM*

<b>Participant</b>	<b>Description</b>	<b>Function of participant</b>	<b>Already in the cloud?</b>	<b>Role in the cloud</b>
EP1	PP1 from the multi-stage study (the cFRAM has been developed in part with their help and examples of PP1 are used in the handbook to illustrate the steps of the cFRAM).	Managing Director	Yes	Software vendor
EP2	PP3 from the multi-stage study, as at the time of the evaluation they were still in the process of migrating software products into the cloud. Therefore, the company was deemed appropriate, as they would be able to use the results produced by the cFRAM immediately.	Leader of product development	No	Software vendor
EP3	The same company with which some of the questions of the multi-stage study were validated (see chapter 3). This company was chosen again because the participant went through the entire process of cloud adoption and was thus able to judge retrospectively how useful the cFRAM might have been, what issues would not have been covered by the cFRAM, etc.	Leader of product development	Yes	Software vendor
EP4	A software vendor start-up that develops products to analyse sports matches. As a start-up, they are a	Managing Director	Yes	Software vendor

	rapidly developing company and might be able to use the results of the cFRAM to plan future steps.			
EP5	The IT department of a faculty of a University that is already offering virtual machines to researchers on request. They are, however, thinking about expanding this service by building their own private cloud to enable the use of experimental software products more easily.	Head of IT	No	Cloud provider
EP6	The IT department of a medium sized company that is currently planning to introduce Office 365 and replace the client based installations of various Office versions.	Head of IT	No	End-customer
EP7	A consulting company that has specialised on IT outsourcing, private clouds, and IT project management. The company currently migrates their email services into the cloud.	Managing Director	Yes	End-customer
EP8	A software vendor that has specialised on developing accounting products. They are thinking about migrating some of their software products into the cloud but are concerned about data security.	Deputy leader of product development	No	Software vendor
EP9	A software vendor that has specialised on developing ERP solutions and other IT related services. They are currently investigating how they can migrate their ERP solution into the cloud.	Head of IT	No	Software vendor
EP10	A consulting company that has specialised on IT infrastructure management and virtualisation. The company is currently developing a private cloud to host software applications on behalf of customers.	System engineer	No	Cloud provider
EP11	A medium sized manufacturing company that is in an early stage of planning to migrate their ERP solution into the cloud.	Head of IT	No	End-customer
EP12	The IT department of a medium sized company that is currently planning to migrate their data warehouse and analytics software into the cloud.	Managing Director	No	End-customer
EP13	The IT department of a medium sized company that is in the process of supporting their sales process through a cloud-based application that runs on tablets so that sales people always have up to date information.	Application manager	No	Software vendor
EP14	A software vendor that has specialised in the development of HR (Human Resources) management applications. The company is an early stage of planning to migrate their software application into the cloud.	Several people from different departments	No	Software vendor

(Please note, EP4-EP14 were selected randomly and were outside of the Oil & Gas industry.)

cFRAM and the handbook were evaluated with every EP in a two to four hour interview. Each interview had three steps. Before the interview, the handbook was sent to the EP with the request to read it. In the second step, the interviewer visited the EP to carry out a cFRAM analysis with the participant by going through the handbook step by step. By doing that, some of the steps could not be applied to their full extent,

e.g. the group discussion in step 1. However, this was necessary as group discussions would have been difficult to organise and would likely have resulted in fewer companies willing to participate. The steps were applied with pen and paper and not with the FRAM Model Visualiser. Pen and paper were used for two reasons. First, it saved time during the interview. Second, it was easier to include the participant in creating the different FRAM models as they themselves could draw the functions, aspects, etc. In the third and last step, the collected data was processed by the interviewer, entered into the FRAM Model Visualiser, and sent to the participant for verification and further feedback.

To conclude the interview, the EPs were asked a series of questions to discuss the strengths and weaknesses of cFRAM and to identify areas for improvement. In addition to discussing the strength and weaknesses of every step of cFRAM, the following high-level questions were asked to engage in deeper discussions with the EPs:

- Do you see a need for cFRAM to address organisational changes cloud computing require?
- How useful is cFRAM in addressing organisational challenges you face with cloud computing? How useful is the analysis of capabilities within cFRAM to structure organisational and technical challenges?
- Does categorising the resources with the four viewpoints help in identifying core capabilities? Are the viewpoints the right ones?
- Does cFRAM help you with thinking about resilience and how to increase it?
- How do you think can the method be improved to increase its validity?
- Is the handbook written in a concise and understandable way?
- Are the examples used in the handbook understandable and sufficient?
- How do you think can the handbook be improved to increase its usefulness?
- Do you think you would have been able to carry out a cFRAM analysis on your own only by reading the handbook, without an expert being present?

## 8.2 Results and discussion

Overall, the results of the evaluation study are unequivocal and lead to the conclusion that cFRAM enables companies to adapt their capabilities to cloud computing in a way that increases the company's resilience. By presenting the results of the evaluation study in detail, the following sections provide the evidence that lead to the conclusion. After presenting the EPs answers to the questions asked after every interview, the results are structured similar to the steps of a cFRAM analysis. Every section will explain the ease with which the steps could be applied by the EPs and highlight EPs that provided noteworthy results.

Every EP of the evaluation study stated that they see a need for a method like cFRAM to address and plan organisational changes required by cloud computing. EP12 summarised the strengths of cFRAM by stating that cFRAM helps in relating the technical and organisational challenges of cloud computing. EP5 and EP12 noted that cFRAM is a good method to show developers and technicians what functions in a FRAM model they are responsible for and how these functions are connected to other functions. EP12 believes that it will make it easier for developers and technicians to understand dependabilities. Understanding dependabilities is necessary in order to increase resilience by moving to the cloud (see section 2.3)

Every EP stated that they find cFRAM helpful in addressing the organisational challenges of cloud computing on different levels of organisation. The strength of cFRAM is that it allows the definition of functions on different levels of detail. EP14 stated that in the first instance they would create a high level FRAM model with the most important functions of their company. Afterwards they would use the high level FRAM model to dive deeper into single functions by creating a separate FRAM model. FRAM models on different levels of detail would allow them to address specific organisational challenges without losing sight of the bigger picture.

Further proving the strengths of cFRAM in addressing the organisational challenges of cloud computing on different levels of organisation to increase resilience is the fact that throughout the analysis the EPs were often reminded of tasks they still have to do

or colleagues they have to contact. This indicates that a cFRAM analysis is addressing the right kind of organisational areas necessary to adopt cloud computing.

Every EP (except EP6) stated that categorising the resources with the four viewpoints (cultural, management, application, and governance) helped them in understanding how their resources are influenced and constrained. Assigning viewpoints to the resources also enabled them to identify capabilities that they think might be affected by a move to the cloud. Thus, the EPs appreciated that cFRAM enables them to focus their resources on capabilities that need to adapt in order to move to the cloud successfully (resources in a general sense, not necessarily those of functions).

EP4 pointed out that the results of their cFRAM analysis could be useful in presenting their company to potential investors. As EP4 is a start-up, investors will be able to get a good overview of the company and its strength and weaknesses. By making investors aware of the capabilities, they will be able to decide how competitive these capabilities are and if it is worth investing in the company. Although using cFRAM as an investment decision tool was not a goal at the outset, it fits into the general scope of the method.

Although not all EPs were familiar with the term ‘resilience’ or ‘Resilience Engineering’, they were familiar with the concepts that form the basis of resilience, i.e. creating processes that are robust yet flexible enough to succeed under varying conditions (see section 2.2). All EPs stated that increasing resilience is becoming more important within their company. They appreciated cFRAM for helping them to get more familiar with the concept of resilience. Some EPs explicitly stated that they appreciated cFRAM for allowing them to visualise resilience and their company’s weaknesses, and suggesting ways to increase resilience.

All EPs stated that the cFRAM handbook is written in a concise and understandable way (a point also appreciated by Erik Hollnagel, the lead developer of FRAM, who read the handbook too). Although some EPs alluded to the fact that they did not read the complete handbook before the interview, all except EP13 read at least the first few sections in which the basic elements of cFRAM are explained. Thus it allowed the

interviews to go through the steps of the cFRAM analysis together right away, without elaborate explanations of the method. Applying the steps of a cFRAM analysis together then showed how well the EPs understood the method and its goals only by reading the handbook.

The opinions about the examples described in the handbook were mixed. All EPs appreciated that the handbook used examples to illustrate a cFRAM analysis in practice. They stated that the examples made it easier to understand the reasons for carrying out the steps. Some EPs (in particular EP2, EP8 and EP11), however, would like to see more examples, in particular, for different kinds of situations. They believe more examples will further enable them to carry out a cFRAM analysis without an expert being present. EP8 stated that the examples used in the handbook are too generic. At the same time EP8 admitted that examples for such methods are often generic. Indeed, it was discussed with EP8 that the examples have to be generic (to some extent) so that people with different backgrounds and different reasons for using cFRAM can understand them.

The majority of EPs admitted that they think they would not have been able to carry out a cFRAM analysis on their own only by reading the handbook or that it would have taken them a long time. Despite all EPs stating that the handbook is a step in the right direction, they believe more examples would significantly help them carry out a cFRAM analysis on their own. After having used cFRAM once with an expert being present, all EPs stated that they think they would be able to carry out a cFRAM analysis again in the future on their own.

Overall, the feedback from the EPs has been very consistent. In other words, they identified the same strengths and weaknesses of cFRAM and had similar suggestions for improvement.

## 8.2.1 Collecting data to identify functions and describing them

### 8.2.1.1 Subject of analysis of the cFRAM evaluation

The subject of analysis with each EP for the cFRAM evaluation has been very diverse. The overall focus was on software vendors and the migration of their software products into the cloud. The focus was on software vendors as the focus of this thesis has been on software vendors. That is also why the cFRAM handbook has been written specifically for software vendors.

At the same time, however, the goal was to understand if cFRAM has a wider applicability. In other words, the evaluation study was designed to find out if cFRAM can be applied to companies that use/plan to use cloud-based software products. A wider applicability would significantly increase the usefulness of cFRAM as companies could use it for different purposes. In other words, after learning the method once they can use it regularly. Thus, the return on investment is higher. Table 11 shows the subject of the cFRAM analysis for all 14 EPs.

Table 11 - Subject of cFRAM analysis

Participant	Company type	Subject of cFRAM analysis
EP1	Software vendor	Migration of software product into the cloud
EP2	Software vendor	Migration of software product into the cloud
EP3	Software vendor	Migration of software product into the cloud
EP4	Software vendor	Development of new software application in the cloud
EP5	IT department	Development of private cloud
EP6	IT department	Adoption of Office 365
EP7	IT consulting company	Migration of email system into the cloud
EP8	Software vendor	Migration of software product into the cloud
EP9	Software vendor	Migration of software product into the cloud
EP10	IT consulting company	Development of private cloud
EP11	IT department	Migration of ERP system into the cloud
EP12	IT department	Integration of cloud-based data analysis services
EP13	Software vendor	Development of new software application in the cloud
EP14	Software vendor	Migration of software product into the cloud

### *8.2.1.2 Usefulness of the generic customer lifecycle*

The handbook proposes a generic customer lifecycle to guide the data collection that is necessary to identify functions and describe aspects (see section 6.3.2). The generic customer lifecycle has already been successfully tested in chapter 6. The evaluation study of cFRAM aimed at further validating the usefulness of the generic customer lifecycle.

The generic customer lifecycle was appropriate for all software vendors that were part of the evaluation study. It has been useful to introduce and explain cFRAM as a method, as it allowed explaining the nature of functions and aspects through examples. In all cases, the functions in the generic customer lifecycle could be adapted or extended (by introducing additional functions) to represent the situation of each individual software vendor accurately.

For all non-software vendors that were part of the evaluation study the generic customer lifecycle could be easily adapted. More specifically, it was useful to explain cFRAM and illustrate the level of detail the identification of functions should aim for (some EPs were not sure at first what level of detail would be appropriate for the goal of the analysis and the scope of the interview).

### *8.2.1.3 Usefulness of the proposed questions*

Due to the generic customer lifecycle and the accompanying questions the identification of functions and the description of their aspects was straightforward and efficient for all EPs (the efficiency is particularly noteworthy as it allowed an effective cFRAM analysis in a limited time interview).

The ease with which cFRAM and its handbook allow the identification of functions became particularly apparent with EP13, who was not able to read the handbook before the interview. With the help of the generic customer lifecycle and the questions it was still possible to identify functions and go through the other steps of a cFRAM analysis in the same amount of time it took for other interviews (for some functions it

was not possible to go into as much detail as would have been desirable, but it was still possible to successfully test cFRAM with EP13).

The handbook proposes the form of a group discussion with employees from different departments for a cFRAM analysis. The form of a group discussion was only possible with EP14. It provided noteworthy results. Three employees were part of the evaluation: the leader of product development, the leader of software development, and the head of IT. The discussions between the employees during the cFRAM analysis revealed that cFRAM is an excellent method to be applied in groups. Every employee of EP14 was able to explain their understanding of how things work within the software vendor to agree on functions that are important. Throughout the latter parts of the cFRAM analysis, in particular for step 4 (Planning the move to the cloud), the form of a group discussion enabled every employee to offer their ideas. Furthermore, the employees were able to comment on each other's ideas to derive at the best plan for moving to the cloud.

#### *8.2.1.4 Relevance of aspects*

The aspects of functions (Input, Output, Time, Control, Precondition, and Resource) were of different importance for the cFRAM analyses. Chapter 6 argued that Time was not important for PP1 and that this might be due to the IT industry (where it is often not critical if a function starts at 14:00 or 14:01). Although Time was by far not as relevant as the other aspects (3 descriptions across the EPs for the *before* cloud FRAM model, compared with Control, 25 descriptions, Precondition, 11 descriptions, and Resource, 10 descriptions), it was important for EP8 and EP13. Time was important for EP8 and EP13 as they have functions that should be carried out in parallel to save time. If, however, these functions are not carried out in parallel, no other function will fail. They would only start later. EP8, for example, has to create new user accounts for their software product in different places. They have to create a user account in their database and Active Directory. Ideally, these two functions should be carried out in parallel so as to save time. EP13 used the Time aspect for similar reasons.

When leaving Input and Output aside (as they are essentially part of every function) Control was the most dominant aspect. The reason for the dominance of Control could be that when the EPs move to the cloud they bring functions previously held by customers or users into the EP where the EP is responsible for them. Thus, they need to pay more attention to control aspects, as they need to ensure that the products or services run as expected by the customer and user, and that they run in a dependable manner. This finding is very similar to findings from the multi-stage study (see chapters 4 & 5).

The analysis of aspects should, however, be treated carefully as every interview was limited in time. Thus, the focus was on identifying and describing the most important functions and aspects (not all aspects possible). If cFRAM would be used in more extensive interviews with the EPs (e.g. a day instead of 2-4 hours) the numbers above would probably change.

## **8.2.2 Identifying performance variabilities**

After identifying, describing and connecting functions every EP was asked to identify performance variabilities. While explaining the theory of performance variabilities it was pointed out that performance variabilities have effects on the Output of functions in terms of time and precision. In the majority of cases this explanation has been sufficient in order to allow the EPs to identify performance variabilities. The EPs went through every function to think about negative events that happened in the past or almost happened but could be avoided. In addition, based on the knowledge gained about performance variabilities in previous interviews, the interviewer asked hypothetical questions about what could go wrong for functions skipped by the interviewee. Often these hypothetical questions enabled the EPs to identify further performance variabilities.

### *8.2.2.1 Ease of identifying performance variabilities*

Chapters 2 and 6 suggested that it could be helpful to develop common conditions (CCs) specifically for the IT industry, similar to those that already exist for the health care industry. The evaluation study has shown that the development of CCs is not a

pressing issue but can enable companies to be more thorough. The CCs could replace the hypothetical questions the interviewer asked during the evaluation study, further strengthening cFRAM when applied without an expert being present. Table 12 presents a list of CCs that have been specifically designed for the IT industry. The CCs have been identified by comparing performance variabilities across the EPs, to derive at the underlying conditions of performance variabilities that appear most often. In order to be included in Table 12 the condition for a performance variability needs to appear across at least three EPs, in order to eliminate those performance variabilities that appear more than once by coincidence. Although this list is not exhaustive it provides a start and can be extended by future research.

*Table 12 - Common Conditions for the IT industry to identify performance variabilities*

<b>CC/name</b>	<b>Explanation</b>
CC1: No user support available	Users have problems using the provided IT services but IT support is not available, e.g. because its the weekend or night or IT support is tailored to UK times.
CC2: Request from user to IT department or software developer is opaque	Users request a new product feature or IT service but do not express the requirements clearly enough, which can lead to the development of product features users were not requesting. That requires the developers to invest more time as might be necessary to fix the developed features.
CC3: Set up of software/hardware takes too long due to insufficient manpower	Users cannot start working as the IT department or software vendor has insufficient manpower to install the requested software/hardware. This CC is often closely related to CC4.
CC4: Request for software/hardware reaches IT department or software vendor too late	The IT department or software vendor is put under pressure because users request software/hardware at short notice and expect the IT department or software vendor to react instantly.
CC5: User has problems using provided IT services	Users might put the wrong types of data into the provided IT services causing the IT service to fail, or they do not understand how to use an IT service which results in a support call.
CC6: No capacity to develop requested product features/IT services	Users request new IT services or product features but the IT department or software vendor has insufficient manpower or monetary resources to develop the requested IT service or product feature. This can result in users taking their own actions or moving to a competitor.

The findings of chapter 2 and the multi-stage study (chapters 4 & 5) have to be kept in mind when using the CCs in Table 12. Chapter 2 and the multi-stage study suggested that companies are too different from each other as to all experience similar performance variabilities so that there could be a general list of principles companies

have to follow in order to become more resilient. Therefore, the CCs are kept abstract to give companies an idea what areas they should investigate for potential performance variabilities. Companies are encouraged, however, to identify additional areas that might be relevant to them.

It is noteworthy that some of the CCs in Table 12 feature elements of use and technology uncertainty (see section 2.3). CC2, for example, captures the case when users or customers do not express their requests clearly enough. Use uncertainty explains what happens when the IT or software vendor does not answer requests from users or customers promptly: users and customers take their own actions. IT departments and software vendors cannot answer requests promptly, however, if these requests are opaque. Thus, it might not only be a responsibility of the IT department or software vendors to deal with use and technology uncertainty. Instead it might be a wider problem that requires better education of users about how IT works. CC4 provides further evidence for the need to educate users better. CC4 captures the case when requests reach the IT department or software vendor too late. Users and customers then expect the IT department or software vendor to deal with the request immediately, without considering that they have other tasks to do. Further studies are necessary to investigate the effects of use and technology in different situations and for different technologies (see section 9.5).

#### 8.2.2.2 *Number of performance variabilities*

Table 13 shows the number of performance variabilities that were identified for each EP. Although the EPs are from different industries, produce different products and move to the cloud for different reasons, it is noteworthy that the majority of EPs experience 2 or 3 performance variabilities. There is no explanation why the majority of EPs experience 2 or 3 performance variabilities.

*Table 13 - Number of performance variabilities before the move to the cloud*

<b>Partici- pant</b>	<b># of Performance variabilities before the cloud</b>
EP1	4
EP2	5

EP3	/(No data because cFRAM was only discussed)
EP4	/(No data because EP only has a cloud product)
EP5	2
EP6	4
EP7	3
EP8	3
EP9	2
EP10	3
EP11	3
EP12	1
EP13	2
EP14	3

The most common performance variability experienced, in some form, across the EPs was best described by EP8. EP8 is responsible for developing and distributing software products that are being used by the company's employees. If an employee has the need for one of EP8's software products, the leader of the department of the employee has to send EP8 an official request. This should happen well in advance of the time the employee needs the software product. Too often, however, the department leaders send those requests too late or incomplete. Late or incomplete requests put pressure on EP8, as they need to suspend other tasks in order to enable the employee to use the software product immediately. EP5, EP6, EP11, and EP13 make similar experiences on a regular basis.

The number of performance variabilities presented in Table 13 should be treated with care, as the significance or severity of performance variabilities can vary between companies. In other words, a performance variability of company A could do more damage than two performance variabilities of company B. The following examples of EP9 and EP13 illustrate the differences in the severity of performance variabilities best.

EP9 develops ERP solutions. As part of their sales pitch consultants of EP9 visit the potential customer to present the product. If the customer chooses to buy the product the same consultant visits the customer to analyse the customer's business processes in order to tailor the product to them. For both functions EP9 sees performance variabilities. Both functions produce their Output too late if EP9 acquires too many customers in a short period of time. In that case, the consultants are too busy to visit new customers. In the majority of cases, EP9 stated that the performance variabilities lead to the customer buying or starting to use the product a few days later than planned.

EP13 is an SME fashion company and towards the end of every season EP13 has left-over stock that they sell for a discount to wholesale customers. Sales people of EP13 visit those customers personally to sell the left-over stock. EP13 is developing a software solution to support the sales people in the process. Currently, however, the sales people receive a paper-based list at the beginning of each week with the number of items left per model, size, colour, etc. The form of a paper-based list has led to many problems. Most noteworthy is the fact that the paper-based lists are quickly out of date as EP13 has many sales people that visit customers. It occurred regularly that sales people sold stock to a customer only to find out later, when the buy was handed over to the finance department for completion, that the stock was not actually available anymore because it was sold to another customer. Over time the sales people adapted to that situation and now call headquarters shortly before meetings and before closing a deal to confirm the numbers they have are still correct. This 'workaround' is, however, not without mistakes and mix-ups of numbers happen regularly. With the help of tablet computers and a cloud-based software application EP13 hopes to provide their sales people with up-to-date numbers that are easily accessible during sales pitches (in addition to other advantages such as showing 3D models of the stock, different colours, etc.). Thus, the tablet computer will enable EP13 avoid selling stock they do not have anymore.

Comparing the performance variabilities of EP9 and EP13 reveals that both do the respected company harm but the performance variabilities experienced by EP13 are much more serious as they regularly cost sales, whereas in the case of EP9 they are

only delayed by a few days. Furthermore, EP13 stated that many customers that thought they made a good bargain for the left over stock only to be disappointed later do not buy from EP13 again. Thus, the long-term consequences for EP 13 are much more severe. Overall, performance variabilities should be considered in the light of the subject of analysis.

In addition to identifying performance variabilities, cFRAM also helped companies identify entire areas within their company that require attention in order to dampen potential performance variabilities. EP4, for example, helped cFRAM and the identification of performance variabilities, realised that by concentrating all their resources on developing their product, they have neglected the relationship with their customers. The majority of their functions are concentrated around the early stages of the customer lifecycle, i.e. customer adoption and their set up, and the later stages of the customer lifecycle have received less or no attention at all, i.e. servicing customer and increasing customer satisfaction. cFRAM helped the start-up to become aware of this situation by showing the implications of performance variabilities on coupled functions, e.g. if the Output of a servicing customer function is imprecise it can affect a function responsible for increasing customer satisfaction negatively. The start-up decided to invest more resources into the latter parts of the customer lifecycle soon in order to dampen the performance variabilities.

### **8.2.3 Identifying existing capabilities**

After identifying the functions, describing their aspects and identifying performance variabilities the EPs were asked to identify existing capabilities that reside within the *before* cloud migration FRAM. To identify existing capabilities, the three most important functions had to be identified by the EPs. After listing the resources of the three most important functions it was investigated how the resources are influenced or constrained by assigning them to one of four viewpoints. (cultural, management, application, and governance). To reveal capabilities that are going to be affected by cloud computing the influences and constraints across the resources of the three most important functions had to be identified.

After identifying existing capabilities for the three most important functions the handbook suggests to carry out the same steps for the remaining functions to identify additional capabilities or find out if the previously identified capabilities are applicable to those functions too. During all interviews there was only time to identify the capability (or capabilities) for the three most important functions. Although it would have been desirable to identify all capabilities that reside within functions it is not a drawback for the evaluation study. As it was possible to identify the capabilities of the three most important functions it shows that the steps necessary to identify capabilities achieve their goal.

#### *8.2.3.1 Usefulness of the identification of capabilities*

When identifying capabilities with EP1-EP10 (except EP9, for which there was no time to identify capabilities), it became apparent that this step goes into a lot of detail. It was often the case that this step goes into more detail than might be desirable for a two to four hour interview. EP1-EP10 stated that they found the identification of capabilities useful to think about their functions and the interdependencies more deeply. Furthermore, they stated that they were positively surprised about the level of detail cFRAM allowed them to go into in a two to four hour interview. However, there was not always time to discuss the results of the analysis in great detail. It was discussed with the EPs that they would get more value out of cFRAM if they were able to investigate the impact of cloud computing on all their capabilities and interdependencies that might exist between capabilities. Doing this, however, would require longer interviews (possibly over several days in order to create FRAM models that are described in sufficient detail to identify all capabilities).

As it could be concluded after the interviews with EP1-EP10 that the steps for identifying and adapting capabilities in cFRAM achieve their goal but go into too much detail, the decision was made to skip this step and the adaptation of capabilities in step 4 of cFRAM for the remaining interviews with EP11-EP14 as these EPs had only two hours for the interview. The interviews with EP1-EP10 revealed that analysing capabilities with cFRAM takes at least 2.5 hours.

### 8.2.3.2 *Identifying the three most important functions and their resources*

Almost all EPs (excluding EP9 and EP11-EP14) stated the focus on the three most important functions to identify capabilities was appropriate and indeed necessary in order to focus on the most pressing issues and not lose sight of the bigger picture. All understood that the same steps would have to be carried out for the remaining functions if it were a real world cFRAM analysis and not an evaluation study.

In order to identify the three most important functions it had to be made clear to the EPs that this does not necessarily mean that three functions that rely on each other have to be the most important ones. EP4, for example, argued along the lines of their customer lifecycle, that if they do not have a function like <Set up customer> they do not need a function like <Service customer> because customers would not be able to use their product in the first place. It was explained to the EPs that the identification of the three most important functions should be more about those functions that, for example, generate the most revenue or higher customer satisfaction. The reason for this is that all EPs, as software vendors or IT departments, aim to deliver products to customers or users. Thus, revenue and customer satisfaction are good indicators of how well the EPs perform.

EP10 provides a noteworthy example for the identification of the three most important functions. For EP10 the three most important functions all work towards the same goal (similar to EP2, EP4, and EP5). EP10 is currently in the process of developing a private cloud to host applications on behalf of their customers. At the moment, EP10 buys hardware for customers and places it in the customer's data centre. Afterwards EP10 installs the desired applications on the hardware. EP10 is aiming to make this process more efficient and faster by developing a private cloud. The three most important functions EP10 identified in their *before* cloud migration FRAM model are all aiming towards finding out what the customer needs and tries to achieve with the applications EP10 offers. Hence, the three most important functions aim to deliver the right application to the customer.

EP7 provides another noteworthy example for the identification of the three most important functions. EP7 identified three functions that were not expected to be the

most important ones by the interviewer (based on experiences from previous interviews). EP7 relies on email to communicate with and support their customers. Customers get an email address for support requests. If EP7 receives an email it is analysed and sorted automatically to forward it to an appropriate technician or consultant. EP7 is currently in the process of migrating their email system into the cloud. EP7 identified functions in the FRAM model as the three most important that are responsible for (1) analysing and forwarding the email to an appropriate technician or consultant, (2) giving the customer feedback and (2) billing the customer. The interviewer expected the three most important functions to be related to actually dealing with customer emails. EP7 explained, however, that the functions they have for dealing with customer emails are necessary functions to have. The three functions EP7 identified to be the most important are those aiming at making the communication with customers as efficient as possible.

#### 8.2.3.3 *Assigning the resources to the viewpoints*

All EPs (except EP6) stated that listing the resources of the three most important functions and assigning them to one of the four viewpoints (cultural, management, application, and governance) allowed them to analyse their resources more deeply to identify factors that influence or constrain them. Furthermore, the EPs stated that they think the four viewpoints are appropriately addressing pressing issues when planning a move to the cloud. Thus, the evaluation study of cFRAM successfully collected more data to confirm that the four viewpoints capture the underlying processes of adapting capabilities to cloud computing (i.e. augmenting cycle of theory development, whereas chapters 4 & 5 carried out the framing cycle of theory development, according to Pan & Tan, 2011).

Some EPs struggled with distinguishing between the viewpoints. This led them to think that more than one viewpoint might be appropriate for a particular resource. Together with the EPs the viewpoints were then discussed and described in more detail by providing examples from previous interviews. EP8, for example, struggled with the distinction between *Management* and *Application*. With regard to resources around managing the development of software applications, e.g. software development platforms such as Visual Studio, EP8 was not sure whether it should be

assigned to *Management* or *Application*. It was recommended to think about how Visual Studio was chosen as the software development platform. Was it chosen because EP8 uses programming languages that are only supported by Visual Studio or was it chosen for its managerial capabilities, e.g. coordinating the work of various software developers, version control, libraries, etc. In the end, EP8 decided that it is more accurate to assign Visual Studio to *Application* rather than *Management*. Afterwards it was also pointed out to EP8 that (1) the decision does not have to be final and that the viewpoint can change over time or in other situations and (2) that the process of analysing what viewpoint is appropriate is more valuable than the final decision.

During the evaluation study it became clear that there is not always a clear distinction between the viewpoints and that it depends on the company and what they are trying to achieve. In other words, some might assign a certain resource to *Application* whereas others would assign it to *Management*. This is not a drawback of cFRAM. Rather it increases its flexibility, specifically when different departments are involved in a cFRAM analysis. Discussing what viewpoint is most appropriate for a resource can provide valuable insights into how resources are used and how their use might be constrained. Indeed, EP5 and EP10 pointed out that this is the reason why they think assigning the resources to one of the four viewpoints is important, as it helps them to think about resources more deeply.

Table 14 - Number of appearances of the four viewpoints for the categorisation of resources

Participant	# of cultural	# of management	# of application	# of governance
EP1	3	1	6	3
EP2	3	1	4	2
EP3	/	/	/	/
EP4	9	10	4	4
EP5	5	6	1	3
EP6	1	6	2	3
EP7	1	3	2	6
EP8	1	5	3	2
EP9	/	/	/	/
EP10	2	5	1	4

(Please note that the data for EP4 shows the resources for the *after* cloud migration FRAM model as EP4 developed their product in the cloud right away)

In the majority of cases the EPs were able to assign the resources to one of the four viewpoints without the help of the interviewer. Table 14 shows the number of resources that were assigned to each of the four viewpoints. The data shows that the total number of resources, regardless of the four viewpoints, differs significantly between the EPs. A possible explanation for this is the level of abstraction. Some EPs preferred to go into more detail and included resources that were secondary or over which they have no influence (e.g. a telephone was a resource for EP8). Being able to define resources on different levels of abstraction is in line with FRAM, which allows users to define functions on different levels of abstraction depending on the goal of the investigation.

Table 14 also shows that none of the four viewpoints is dominant. With the data from the evaluation study it can be concluded that the representation of viewpoints depends on the company and the goal the company tries to achieve with cloud computing. Hence, all viewpoints are equally relevant for a move to the cloud. This finding is in line with the findings from the multi-stage study (chapter 4 & 5). Furthermore, when moving to the cloud and adapting functions, the resources of these functions adapt accordingly. Companies, when moving to the cloud, should not aim to balance the four viewpoints. Balancing the four viewpoints is not possible as they can sometimes be in conflict with each other (e.g. making a decision when addressing management issues can constrain the options available for application issues). Rather, companies should accommodate the four viewpoints by deciding which viewpoint is most important or presents the most pressing issues for their particular situation.

Accommodating the viewpoints appropriately is important as they influence the adaptation of capabilities (see chapter 5). The viewpoint which is most important can change over time or during the adoption of cloud computing. For example, software vendors that migrate one of their products to the cloud might decide that they want to migrate the product without making substantial changes to it, similar to what EP8 decided. In this case, *Cultural* might appear particularly dominant because the

company wants to find out how customers react to a cloud product. Over time, the focus might shift towards *Management* in order to make the move to the cloud as efficient as possible. A section has been added to the end of the cFRAM handbook that explains how companies can accommodate the viewpoints to help them use cFRAM as a long-term planning method during the move to the cloud.

#### *8.2.3.4 Summarising the findings in the definition of capabilities*

Naming and defining capabilities after identifying the three most important functions with their resources and assignment to the four viewpoints is the last step in analysing a company before the move to the cloud is planned. The capabilities are, therefore, a way to summarise previous analysis efforts (similar to the multi-stage study where the capabilities summarised the steps the software vendors took to migrate their software products into the cloud). When defining and describing the capabilities many EPs had problems. Three problems stand out.

First, some EPs struggled naming their capability (or capabilities). Too often the names they gave their capabilities sounded more like the goal the capability tries to achieve, rather than how it tries to achieve it. The goal of EP6, for example, is the timely provision of tools such as Office and Adobe Photoshop. Therefore, they named their capability 'In-time tool provision'.

Second, some EPs were unhappy with the definition and names of their capabilities as they complained that they were sounding too generic. EP5, for example, is currently providing virtual machines on request on dedicated servers. They are planning to expand this service and act more like a cloud provider. To provide virtual machines on request, they defined one of their capabilities as 'Provision of managed computing resources'. It was discussed with EP5 that capabilities that sound generic are not necessarily a drawback as the aim of the definition of capabilities is to summarise what the EP can and cannot do. Therefore, if everyone within EP5 understands what the capability means in detail (e.g. which functions are part of it) they can sound generic. Generic sounding capabilities, however, require documentation of the capabilities so that they can be shared among employees.

Third, some EPs tried to identify their overarching capability, i.e. those that reside in all functions and not just within the three most important ones. EP10, for example, explained that their core capability is the inherent knowledge about the IT market they have among their technicians and consultants. The inherent knowledge helps them offer the right solutions to customers at the right point in time. This sort of capability, however, applies to many functions within EP10. The focus of the evaluation study with EP10 was on one particular software application and the aim of EP10 to act more like a cloud provider. Therefore, the identification of a capability that is directly related to the delivery of software applications could have provided results with which EP10 would have been able to plan their move to the cloud in more detail.

The three points discussed above are not necessarily a drawback of cFRAM, as the definition of capabilities is only a way to summarise a cFRAM analysis. Going through the steps of a cFRAM analysis is more important and reveals more insights than the name and definition of a capability itself. Furthermore, naming the capability intends to help employees refer to known issues among each other and to avoid misunderstandings. Table 15 shows some of the capabilities and their definition for EP1-EP10.

*Table 15 - Name and definition of capabilities before the cloud*

<b>Participant</b>	<b>Capability before the cloud</b>	<b>Definition of capability before the cloud</b>
EP1	Service delivery management	The capability is responsible for developing the right software solutions for the right customer.
EP2	Software service management	The capability is responsible for supporting customers in the use of EP2's software products.
EP3	/	/
EP4	/	/
EP5	Provide managed computing resources	The capability is responsible for offering and enabling users of EP5 to use virtual machines for a variety of purposes.
EP6	In-time tool provision	The capability is responsible for enabling users of EP6 to use a variety of tools such as Office that EP6 offers as the IT department of the company. The capability also applies to other services the IT department offers, e.g. Adobe Photoshop

EP7	React to and anticipate failures of customers in a structured but flexible manner	The capability is responsible for dealing with customer failures and other sorts of customer requests in an efficient manner. The capability is optimised towards anticipating failures so that customers do not experience outages of services.
EP8	Provision, support and development of user applications	The capability is responsible for developing the right software solutions for the right customer.
EP9	/	/
EP10	Inherent knowledge of IT market	The capability captures the knowledge about the IT market that EP10's technicians and consultants have acquired over the years. The inherent knowledge helps them to offer their customers the right software solution at the right time by anticipating market developments.

#### **8.2.4 Planning the move to the cloud**

The data collected in steps 1-3 of a cFRAM analysis form the basis for the fourth step of cFRAM: Planning the move to the cloud. In this step, the functions of the *before* cloud migration FRAM model are adapted to accommodate cloud computing and, if necessary, new functions are introduced. Afterwards, the effects on existing performance variabilities are analysed to inform further adaptations of functions that might be necessary to dampen performance variabilities. In addition, the cloud might introduce new performance variabilities. In the last step, the effects of the adapted functions and performance variabilities on the existing capabilities are analysed by listing the resources of functions in the cloud and assigning them to one of the four viewpoints. By comparing the list with the list of step 3 of cFRAM the adaptation of capabilities is informed. At the end of step 4 of cFRAM companies have created the *after* cloud migration FRAM.

##### *8.2.4.1 Adapting functions to accommodate cloud computing*

To adapt functions to accommodate cloud computing the changes in responsibilities cloud computing triggers were discussed with each EP. Afterwards, the EPs went through every function of their *before* cloud FRAM model to discuss the effects of

cloud computing on individual functions. Two differing results could be observed as the EPs went through the process of adapting functions.

First, with companies in an early stage of moving to the cloud, e.g. in a stage where they are still discussing the usefulness of cloud computing, the discussions with the EPs were engaging and stimulating. These EPs were discussing the underlying principles of cloud computing in order to adapt their functions in the best way possible. EP11, for example, is currently discussing to replace their on-site ERP solution with a cloud-based solution. They only had an initial brainstorming session to discuss general advantages and disadvantages of cloud computing. They stated that the cFRAM analysis helped them to see how many functions would become obsolete by a move to the cloud. Furthermore, they identified a major area that a potential ERP cloud vendor would need to address. Their ERP solution is highly customised with customisations developed by EP11. At the moment, when they receive updates from their ERP provider, they can decide themselves when to install the update so that there is sufficient time to test the impact of the update on the customisations. With a cloud-based solution this would be more difficult as the ERP cloud vendor provides updates automatically. With the help of the cFRAM analysis, EP11 was able to clearly define this potential challenge by investigating how they could adapt functions to decrease the impact of updates on customisations. Furthermore, it allowed them to plan the next steps in their move to a cloud-based ERP solution.

Second, companies that are in an advanced stage of cloud adoption do not appear to benefit as much from a cFRAM analysis as those in an early stage. These EPs have started to adapt functions and thus stick to what they already know about cloud computing and how their company is planning to adopt it. They did not seem to be interested in exploring alternative, possibly better, options. The problem of sticking to existing knowledge was overly pronounced for EP9. EP9 is currently in the process of migrating their ERP solution into the cloud to offer it as a service (see above). When investigating the changes to functions of the *before* cloud and *after* cloud migration FRAM model it becomes clear, however, that they will in fact not offer their ERP solution as a service but simply as a remote hosting solution. The interview with EP13 provided similar results in terms of relying on the knowledge already acquired during

the migration. Thus, the aim of enabling users of cFRAM to make unbiased decisions about organisational changes (see section 7.2) could only be achieved for EPs that were in an early stage of moving to the cloud.

Analysing how functions need to adapt to accommodate cloud computing did not always provide positive results that convinced or enabled the EPs to move further in their move to the cloud. EP8, for example, was able to conclude that migrating their software product into the cloud provides no additional value. Instead, it would create more work for them. The number of functions that have to adapt to cloud computing and the number of functions that need to be introduced illustrate this fact very well. Before the cloud EP8 had 11 functions. In order to move to the cloud EP8 would have to adapt 2 functions and introduce 2 new functions. These 4 functions, however, would not yet enable EP8 to address weaknesses that currently exist, i.e. for the on-site version of their software products. Furthermore, the cloud would expose EP8 to additional performance variabilities (see further below). EP8, however, still appreciated cFRAM for clarifying the situation and allowing them to identify weakness in their current approach that they will aim to address before moving to the cloud.

When adapting functions to cloud computing and introducing new functions the *after* cloud migration FRAM models often appeared more complex, a point raised by the EPs, in particular EP2, EP11, and EP13. When discussing the changes to the functions in detail it became clear that the FRAM models appear more complex because the EPs take over responsibilities and tasks previously held by their customers or users. In other words, the EPs become more complex but the solutions for their customers and users become simpler and easier to manage. The multi-stage study produced similar findings. Table 16 shows a list with the numbers of functions each EP had to adapt or introduce and how many functions cloud computing made obsolete. In Table 16 a function was counted as adapted if its purpose changed or one of its six aspects changed.

When analysing Table 16 it is important to note that the number of changes to functions is not necessarily reflecting the complexity of changes. Changing just one

function can be a significant and time intensive task for any EP, e.g. due its dependencies and resources. Furthermore, it is questionable how useful a comparison of the numbers in Table 16 between the EPs is, as functions can be defined on different levels of detail (chosen by the EPs). Hence, the more detail the EP went into, the higher the number of functions that changed to accommodate cloud computing. Therefore, the numbers in Table 16 should be used with caution and considered in light of the subject of analysis.

The case for EP6 illustrates the above point very well. After adapting functions to accommodate cloud computing EP6 concluded that not many functions would need to change. The cFRAM analysis was, however, able to show the wider implications when adapting just a few functions. EP6 stated that they were surprised by how a seemingly simple change in the licensing model of a software product can ripple through the whole company. EP6 is planning to adopt Office 365, with monthly payments per Office user, to replace their current model of buying an Office product per user. The use of Office 365 will enable them to migrate every user to the same version of Office as, at the moment, some use Office 2007, others use Office 2010 and again others use Office 2013. Although this process appeared straightforward to EP6 at first, the cFRAM analysis revealed that adopting Office 365 would put a greater pressure on the IT department of EP6. The reason for this is that all users will automatically get the latest updates. If an update should contain bugs or cause difficulties during the installation, all users will be affected at once. EP6 decided to react to this by not making the move to Office 365 public and only migrating a few selected users at first, to gain experiences with the cloud. Similar experiences were made by EP11.

Chapters 1 and 2 explained that many companies neglect complementary organisational changes when adapting to technological discontinuities. The example of EP6 as described above provides an explanation for why companies neglect these changes: In their understanding only one function appears to change when moving to the cloud, without realising that changing one function can have an effect on coupled functions. cFRAM could prove to be a viable method to help companies understand how the effects of adapting one function can ripple through the entire company.

Table 16 - Changes to functions required by the move to the cloud

Participant	# of functions that had to be adapted	# of functions that had to be introduced	# of functions that became obsolete
EP1	1	2	1
EP2	1	6	3
EP3	/	/	/
EP4	/	/	/
EP5	2	3	1
EP6	1	1	1
EP7	1	1	0
EP8	2	2	1
EP9	4	5	3
EP10	3	3	2
EP11	2	0	3
EP12	2	3	0
EP13	3	8	5
EP14	3	5	2

#### 8.2.4.2 Dampening performance variabilities through the move to the cloud

After adapting functions to accommodate cloud computing the EPs were asked to investigate the impact of these adaptations on existing performance variabilities and to conclude if new performance variabilities are introduced by being in the cloud. The EPs stated that cFRAM allowed them to visualise performance variabilities, which would make it easier to plan steps for dampening or even eliminating them, thus increasing the resilience of their company. At the same time, the cFRAM analysis raised awareness among the EPs that a new technology, such as cloud computing, is not only helping them to dampen or eliminate performance variabilities but also introduces new performance variabilities. The results of the cFRAM analysis allowed the EPs to identify areas in which they have to take steps after the move to the cloud to address new performance variabilities. Beyond that, the results of this step of the cFRAM analysis produced mixed results.

Many EPs were able to dampen or even eliminate performance variabilities by moving to the cloud. The performance variabilities these EPs were able to dampen were, however, often the result of technical changes. In other words, the performance variabilities could be dampened due to the effects of cloud computing and less due to actions taken by the EPs. In the cloud EP2, EP10, and EP14, for example, take over the installation of the software product, updating the software product, and supporting the software product. By dampening the performance variabilities for these tasks the EPs are able to work more efficiently and increase customer satisfaction. The FRAM analysis with PP1 (chapter 6) produced similar results.

Although many EPs were able to dampen the performance variabilities that are related to technical challenges, the majority of EPs were not able to dampen performance variabilities that are related to organisational challenges. EP6 and EP8, for example, have performance variabilities that exist between them and other departments. Both rely on the leader of a department to contact them if an employee requires a software application. It was already discussed above that the leaders of departments sometimes forward the requests for a software application too late. The EPs that experienced these kinds of performance variabilities were not able to dampen them. They were not even able to inform steps that could be taken in the future to dampen them. The reason the EPs struggled with these kinds of performance variabilities might be that the interviews for the evaluation study were with the leaders of IT or product development and not with the departments that were causing the performance variabilities. When performance variabilities span across multiple departments it might be necessary to work with all affected departments to adapt functions appropriately.

Table 17 shows the total number of performance variabilities the EPs experienced before the cloud, the number of new performance variabilities cloud computing introduces, and the total number of performance variabilities in the cloud. What becomes apparent by investigating the data in Table 17 is that all EPs except EP8 and EP12 were able to decrease the total number of performance variabilities by moving to the cloud. This in itself is, however, not a good indicator for concluding that these companies also became more resilient. Differing severity of performance variabilities

was already discussed above. It is less a drawback of cFRAM and more a drawback of the theory of performance variabilities. Due to the large number of participants in the cFRAM evaluation study (the first study of its kinds that uses FRAM to investigate similar questions within 14 different companies) it is possible to conclude that performance variabilities should be applied with care and that it is important to document the underlying causes of performance variabilities. In other words, it is not sufficient to conclude whether performance variabilities affect the Output of a function in terms of time or precision. To provide a step into this direction, the remainder of this section will focus on analysing the underlying causes of performance variabilities, rather than the number of performance variabilities, in order to conclude if a company becomes more resilient by moving to the cloud.

Table 17 - Dampening of performance variabilities by moving to the cloud

Participant	# of Performance variabilities before the cloud	# of new performance variabilities by being in the cloud	# of Performance variabilities in the cloud
EP1	4	1	1
EP2	5	0	2
EP3	/	/	/
EP4	/	/	2
EP5	2	0	1
EP6	4	1	3
EP7	3	0	2
EP8	3	1	4
EP9	2	0	1
EP10	3	0	2
EP11	3	0	2
EP12	1	0	1
EP13	2	2	2
EP14	3	0	2

By investigating the cause of the performance variabilities in Table 17 it becomes clear that cloud computing does indeed make *individual* EPs more resilient. The reason for the increase in resilience is due to the dampening of performance variabilities caused by technical factors (see above). In the majority of cases,

technical performance variabilities affected the company as a whole, whereas other performance variabilities of the evaluation study affected individual employees or departments. With the examples of EP6 and EP8 the difference in technical and other types of performance variabilities can be illustrated. As explained above, EP6 and EP8 stated that sometimes other departments request software products for employees too late, which puts pressure on EP6 and EP8 because the new employee needs to be able to work. At the same time, however, this performance variability only affects individual employees and not the company as a whole (as it would be in the case of a cloud outage, which would be a technical performance variability).

With the data in Table 17 it is not possible to argue if cloud computing makes companies more resilient *in general*. This is partly due to the drawback in the definition of performance variabilities explained above, due to the differences in performance variabilities the EPs experience, and the steps they take to dampen them. Different kinds of performance variabilities can have different implications on resilience overall. If all EPs experienced the same performance variabilities and were able to dampen or eliminate these through cloud computing, it would be possible to argue that cloud computing makes companies more resilient. The fact that performance variabilities differ significantly between companies strengthens the need for methods such as cFRAM to allow companies to investigate their performance variabilities and find ways to eliminate or dampen them.

#### 8.2.4.3 *Adapting capabilities*

The final step in a cFRAM analysis is the investigation of the changes to functions and performance variabilities on existing capabilities to inform their adaptation or the development of new capabilities. In order to inform the adaptation of capabilities or the development of new capabilities the EPs (this section only refers to EP1 – EP10, except EP9, for the reasons explained in section 8.2.3) were asked to focus again on the three most important functions that were identified for the *before* cloud migration FRAM model, i.e. step 3 of a cFRAM analysis (Identifying existing capabilities). Step 3 was essentially repeated for the *after* cloud migration FRAM model. The resources of the three most important functions are listed and assigned to one of the four viewpoints (cultural, management, application, and governance). By comparing the

list with the list created in step 3 the EPs could compare the resources and their viewpoints to understand if resources became obsolete (or if new resources are available) and why viewpoints might have changed. A change in resources or viewpoints can have an impact on existing capabilities.

The EPs stated that investigating the impact of cloud computing on their existing capabilities to inform the adaptation or development of new ones has been very helpful in understanding the organisational changes cloud computing requires. It showed them, for example, what the effects are if a viewpoint of a resource changes and how they can react to it. Furthermore, the EPs appreciated structuring the organisational changes with the help of capabilities as they stated it makes it easier to communicate organisational changes between different groups of employees. EP5, for example, stated that it could help them discuss the organisational changes with their technicians (who would otherwise focus only on technical changes).

The cFRAM evaluation study is only able to indicate how the four viewpoints and capabilities can be used for planning a move to the cloud due to the time constraints of every interview (as already explained in section 8.2.3.4). Thus, the analysis of resources with the four viewpoints to identify factors that influence and inform the adaptation of capabilities for cloud computing did not produce results in the level of detail of the multi-stage study. The discussion with the EPs during the evaluation study was still able to prove, however, that cFRAM enables companies to anticipate some effects of cloud computing while enabling them to react to others in the way the underlying model of cFRAM was designed (see section 7.3).

Table 18 shows the capabilities of the EPs both for the *before* cloud migration FRAM model and the *after* cloud migration FRAM model. The comparison of capabilities before the move to the cloud and after the move to the cloud provided interesting results. Table 18 shows that for many EPs their capabilities did not have to change significantly. In fact, in the majority of these cases the companies were required to add responsibilities to their existing capabilities rather than having to replace them because cloud computing makes them obsolete. What becomes clear through the data presented in Table 18 is that *software vendors* were most often required to

significantly change their capabilities in contrast to companies that adopted cloud computing for other reasons, e.g. to outsource their infrastructure such as EP7. With the data collected during the evaluation study it is not possible to conclude if the results can be explained by either of the following two suggestions.

*Table 18 - Capabilities of the EP' before and after the move to the cloud*

<b>Participant</b>	<b>Capability before the cloud</b>	<b>Capability in the cloud</b>
EP1	Service delivery management	Service delivery management + Cloud service management
EP2	Software service management	Software service management + Software service development
EP3	/	Educating users of our philosophy
EP4	/	/
EP5	Provide managed computing resources	Provide ad-hoc computing resources
EP6	In-time tool provision	In-time tool provision & maintenance
EP7	React to and anticipate failures in a structured but flexible manner	React to and anticipate failures in a structured but flexible manner
EP8	Provision, support and development of user applications	Provision, support and development of user applications and cloud environment
EP9	/	/
EP10	Inherent knowledge of IT market	Inherent knowledge of IT market + Cloud service management

First, it could be the case that software vendors that migrate (some of) their products into the cloud are affected by cloud computing to a greater extent than other companies. Software vendors might be affected by cloud computing to a greater extent as they link their revenue directly to cloud computing by relying on it to provide their products to customers (e.g. EP2 and EP4). If the cloud has an outage, the software vendor's products are not available and their revenue is affected negatively. Thus, software vendors integrate cloud computing more deeply into their company and are required to adapt their capabilities appropriately. Hence, the effects of technological discontinuities discussed in section 1.1 apply to a large extent as cloud computing renders some of the software vendors' existing capabilities obsolete and cloud computing can be considered more as capability-destroying.

Second, it could be the case that those companies other than software vendors that adopted cloud computing, for example, to outsource parts of their infrastructure, were generally prepared for a technological discontinuity such as cloud computing. These companies appear to be prepared for a technological discontinuity because many of the functions of these companies did not have to change when applying step 4 of the cFRAM analysis (Planning the move to the cloud). In other words, these companies continually question their functions and adapt these to new information. EP7, for example, as an IT consulting company, is currently in the process of migrating their email service into the cloud. Naturally, this company has a deep expertise in cloud computing and the underlying technologies as they have consulted some of their customers during the migration process into the cloud. The cFRAM analysis showed that EP7 only has to introduce one new function in order to migrate their email services into the cloud successfully. The discussion with EP7 revealed that this is due to the fact that EP7 optimised their email services over the years based on their own expertise and experiences they made with customers. In other words, their email services have probably been influenced by the development of cloud computing, although they are only now moving to the cloud, as EP7 stated. Hence, the effects of technological discontinuities discussed in section 1.1 do not apply to these kinds of companies as they only have to adapt their capabilities slightly. In the majority of cases it was necessary for these companies to extend the responsibilities of their capabilities and cloud computing can be considered more as capability-enhancing.

In reality, both cases apply depending on the company and their history. Indeed, the multi-stage study was able to conclude similar findings. PP2 of that study (see section 3.2) had extensive experience with a remote hosting solution similar to cloud computing. They developed this solution on their own in 2001, five years before the commercialisation of cloud computing. When they started to adopt cloud computing they still had to change some of their functions, albeit to a lesser extent than the majority of other PPs.

The cFRAM evaluation study leads to a refinement of the theory of technological discontinuities. Chapter 1 argued that technological discontinuities can be classified

as either capability-enhancing or capability-destroying and that cloud computing was capability-destroying. The evaluation study (and to some extent the multi-stage study) has shown that a clear distinction into capability-enhancing and capability-destroying is not always possible. Instead it depends on the history of the company that is adopting the technological discontinuity as the history determines what functions exist within the company, what skills employees have, etc. Depending on the functions and skills, technological discontinuities have varying effects and are not always clearly capability-enhancing or capability-destroying. Instead, the notions of capability-enhancing and capability-destroying should be considered as the ends of a continuum where technological discontinuities, depending on the history of the company, are either more capability-enhancing or more capability-destroying. By showing what impact cloud computing has on existing capabilities, cFRAM can help companies understand what end of the continuum they are placed at. Understanding where along the continuum they are placed will enable them to move to the cloud efficiently as they can focus their resources on adapting those capabilities that require adaptation.

### **8.3 Improving cFRAM**

#### **8.3.1 Improvements to cFRAM and its handbook**

Evaluating cFRAM and its handbook in the field was useful to identify the strength and weaknesses and to identify areas for further improvement. After having evaluated cFRAM with 10 of the 14 participants, several minor improvements to cFRAM and its handbook were made, e.g. simplifying the language or reorganising paragraphs. In addition, four more substantial changes to cFRAM and its handbook were made that are explained below. The improvements were made based on the feedback from participants. They were made after 10 interviews in order to have enough participants available to test the improved cFRAM and its handbook. The feedback of the group including the first 10 participants was compared with the feedback of the group including the last four participants to understand if the improvements have indeed improved cFRAM and its handbook. Based on this comparison it is possible to conclude that the improvements helped the participants apply cFRAM. Therefore, the improvements have been successful.

The first improvement that has been made to cFRAM regards the steps of a cFRAM analysis. They were reorganised to make it easier for users to focus on the two main objectives of cFRAM: investigating performance variabilities and investigating capabilities while moving to the cloud. More specifically, the first step has been extended to include more detailed questions so that the first step is sufficient to create a FRAM model accurately describing the company. The second step now focuses solely on the identification and description of performance variabilities. The goal of this change is a stronger focus on performance variabilities. Furthermore, by putting all the questions to create the *before* cloud migration FRAM model into one step, companies can decide themselves into how much detail they want to go. The ability to be able to create FRAM models more quickly albeit on a more abstract level was frequently requested by companies. cFRAM recommends that it is perfectly acceptable to create a high level FRAM model at first in order to get a general idea of the impact cloud computing will have on the company but that in order to plan the move to the cloud it is necessary to create more detailed FRAM models, i.e. it is necessary to answer the majority of the questions the handbook proposes. The creation of high-level FRAM models will likely create more success stories among users of cFRAM as it generates usable results more quickly. In addition, the CCs developed in section 8.2.2.1 have been added to step 2 of cFRAM (Identifying performance variabilities) with the note that the CCs are there to assist the companies in identifying performance variabilities but that they should also think about additional areas where performance variabilities might exist.

Second, with the data collected during the evaluation it was possible to link the organisational changes cFRAM is helping companies to identify to the technical adoption challenges of cloud computing. Linking the organisational changes to technical challenges is an essential step in order to ensure that a socio-technical perspective is maintained. Furthermore, the feedback from the participants has revealed that it is also important for their technical people so that they understand why it is necessary to carry out organisational changes. Several participants pointed out that, otherwise, technical people often forget about the organisational changes or ignore them. Linking the organisational changes to technical challenges is done in

step 4 of cFRAM (Planning the move to the cloud). When companies discuss how existing functions need to change for cloud computing and what new functions need to be introduced, companies are encouraged to ask themselves two questions at the same time. First, how do we want to carry out this function from an organisational perspective? Second, how do we want to carry out this function from a technical perspective? It is important to discuss both questions together in order to retain a socio-technical perspective and ensure technical solutions do not prescribe organisational changes and *vice versa*.

Third, the guidance on the viewpoints has been extended. In the original handbook the guidance on the viewpoints and how resources should be assigned was kept too abstract. It was assumed that the explanation of the viewpoints would be sufficient to allow companies to assign the resources to them. The cFRAM evaluation has shown, however, that the participants sometimes struggled categorising the resources with the viewpoints. EP1-EP10 (except EP9) concluded that the four viewpoints are appropriately addressing their most pressing issues. Thus, there is no problem with the viewpoints itself but rather with their description and application. In the improved version of the handbook the description of the viewpoints has been reworked with a particular focus on distinguishing the differences between the viewpoints so that users of cFRAM find it easier to decide which viewpoint is most appropriate. In addition, more examples have been added to illustrate which viewpoint might be most appropriate for which resource (see Table 19 for the table that has been added to the handbook and replaced Table 6 from section 7.4.3). The examples come from the evaluation study and the multi-stage study (see chapters 4 & 5).

Table 19 – Examples of resources assigned to the viewpoints

Viewpoint	Examples of resources
Cultural	Internalised rules and norms of behaviour that employees follow, e.g. sales people
	Preferences or behaviours of customers and users
	Knowledge base to inform about past customer interactions and preferences
	Market situation and behaviour of competitors
Management	Roadmap for product development, e.g. documents ensuring product features are feasible and developers have appropriate skills

	Coordinating feature development and the lifecycle of technologies generally
	Supporting the customer in using the product or increasing customer satisfaction, e.g. ticket system for support
	Software development environments, e.g. Eclipse or Visual Studio
	Communication tools, e.g. email
Application	Development of product features and the approaches used, e.g. SCRUM
	Dependencies that can enhance or stifle the development of a product, e.g. APIs or libraries
	Programming languages that are being used, e.g. C# or Java
	Set up of the product at the customer site, e.g. installation procedures
Governance	Formal rules and norms of behaviour, e.g. for interacting with customers
	Governmental or institutional laws and regulations
	Corporate policies, e.g. documents to ensure corporate identity is achieved
	Handbooks or product documentations that specify what the software vendor's product can and cannot do
	FAQs created for customer support

Fourth, the handbook has been optimised to make it easier for people to carry out a cFRAM analysis in practice and without an expert being present. The participants of the evaluation study provided diverse feedback to optimise the applicability of cFRAM. The feedback ranged from the wish to create a software tool to a one-page sheet with the most important information on cFRAM. Some of the feedback from the participants is infeasible for the scope of this thesis, such as the software tool, and may only be carried out in the long-term. In addition to creating a one-page sheet with the most important information on cFRAM, the handbook now also refers to a website<sup>4</sup> that contains examples that aim to show how cFRAM can be used in different situations. At the same time, however, it was important to keep the information and examples on the website abstract and only show high-level functions as otherwise people who want to carry out a cFRAM analysis might let their models be influenced too much by the examples. The one-page sheet is attached to the handbook and was identified by several participants as important for one reason.

---

<sup>4</sup> <http://thecfram.wordpress.com>

cFRAM is designed for application in groups. It would be too distracting to provide every participant with the handbook. It is still important for the participants, however, to understand the steps of a cFRAM analysis. The one-page sheet provides every participant with information on functions, the aspects, and the main steps so that they have an easy way to refer to it in case they get lost during the group application of cFRAM.

### **8.3.2 Steps taken to increase the usefulness of cFRAM in addition to the evaluation study**

Two steps have been taken in order to disseminate cFRAM and its handbook. First, a website has been developed that introduces the idea of cFRAM and provides a platform to download the latest version of the handbook and access examples of cFRAM analyses that come from the evaluation study. Depending on the success of the website it will be extended to include more examples or different versions of the handbook aimed at different types of companies, e.g. end-customers of cloud computing or cloud providers.

Second, cFRAM and the cFRAM example from the handbook have been presented at two conferences to help spread the ideas of cFRAM among resilience experts. The feedback provided by participants of the conference was very positive and has helped to further improve cFRAM and its handbook. In addition, cFRAM was also discussed in person with the lead developer of FRAM, Erik Hollnagel. His feedback further helped to improve the handbook and increase the overall usefulness of cFRAM. In particular, his extensive practical experience helped in deciding what information users of cFRAM need to be provided with in order to be able to apply it on their own and without an expert being present. The cFRAM handbook inspired the development of a new FRAM handbook (which is written by Hollnagel).

## **8.4 Conclusions**

This chapter evaluated and improved cFRAM and its accompanying handbook by testing them with 14 companies from industry. The evaluation study has been critical to conclude that cFRAM achieves its goal of enabling companies to adapt their

capabilities to cloud computing in a way that increases the company's resilience. Furthermore, the evaluation allowed cFRAM to be improved in ways that make it easier for people to use cFRAM without an expert being present. Improvements that were made are, for example, the development of a one-page note sheet, the development of a website to provide examples of cFRAM applications, the reorganisation of the cFRAM steps, and the editing of explanations and descriptions. Based on the feedback from the EPs it can be concluded that cFRAM is ready to be used in practice.

The evaluation study was also able to confirm that cFRAM is the answer to the overarching research question of this thesis: *how can companies adapt their capabilities to technological discontinuities to increase the company's resilience?* By assisting companies in adapting their capabilities to cloud computing, cFRAM enables them to increase their resilience on two levels. First, it enables companies to inform steps to dampen or eliminate performance variabilities. Dampening or eliminating performance variabilities is necessary to increase the resilience of companies, as chapter 6 discussed in detail. Second, it enables companies to identify existing capabilities and investigate the impact of cloud computing on these capabilities to inform their adaptation. Appropriately adapting capabilities is necessary as resources of capabilities can become obsolete in the cloud or new resources can be introduced (as chapters 4 & 5 discussed in detail). cFRAM enables companies to discuss what resources became obsolete in the cloud and what resources are newly introduced to ensure that capabilities are adapted appropriately.

## 9 Conclusions

The overall conclusion of this thesis is that companies can adapt their capabilities to technological discontinuities to increase the company's resilience with the help of cFRAM. Thus, cFRAM provides the answer to the overarching research question: *How can companies adapt their capabilities to technological discontinuities to increase the company's resilience?*

To inform the development of cFRAM, the answers to RQ1 and RQ2 were combined. To answer RQ1, chapters 4 & 5 defined four viewpoints that capture the underlying processes of adapting capabilities to cloud computing. To answer RQ2, chapter 6 tested FRAM (Functional Resonance Analysis Method) successfully as a method that informs steps to increase and measure resilience. Chapter 7 extended FRAM through the four viewpoints to enable the analysis of capabilities when moving to the cloud (thus the name cFRAM). The evaluation study of cFRAM with 14 companies from industry in chapter 8 proved that cFRAM achieves its goal of enabling companies to adapt their capabilities to cloud computing in a way that increases the company's resilience.

In addition to the overall conclusion, this thesis also provides three minor conclusions where cFRAM helps. First, chapter 8 explained the differences in the approaches to cloud computing taken by the Evaluation Partners to conclude that there is no rigorous process to adapt capabilities to cloud computing that companies can follow. Thus, there is a need for a method like cFRAM. Second, increasing resilience is a continuous process as the internal and external environment of complex systems changes. More specifically, performance variabilities need to be continuously monitored and capabilities need to be continuously adapted. cFRAM assists companies throughout the migration and beyond. Third, the four viewpoints developed in chapter 5 capture the underlying processes of adapting capabilities to cloud computing (i.e. framing cycle of theory development). Chapter 8 tested the viewpoints in different situations and environments to conclude that they appropriately influence and inform the adaptation of capabilities (i.e. augmenting cycle of theory development). By carrying out similar studies for other technological discontinuities it should be possible to develop similar viewpoints for these

technological discontinuities. This would allow an adaptation of cFRAM to other technological discontinuities (see section 9.5 for details).

### **9.1 Lessons learned from the multi-stage study and FRAM analysis**

The multi-stage study was informed by adopting an adaptive Socio-Technical Systems (STSs) perspective. The theory of adaptive STSs largely reflects the software vendor's situation. However, there are elements of adaptive STSs that did not apply to the software vendors. The software vendors from the study have to interact more extensively with their customers (and cloud provider) after the migration into the cloud. The software vendors find that important and necessary because the customers rely on the software vendor for the dependable operation of the product and the software vendor relies on the cloud provider to do their part. Hence, it is necessary for software vendors to be aware of the boundaries of systems in the cloud, which can be: (1) the software vendor, (2) the software vendor and their customers, (3) the software vendor and the cloud provider, and (4) the software vendor, their customers and the cloud provider (there are potentially more systems depending on the number of cloud providers the software vendor is working with, the number of products they sell, etc.). Depending on the system, the software vendor has to be aware of different priorities and consider different types of decision makers. Furthermore, being aware which of the above four systems the software vendor is currently dealing with helps in reacting to and anticipating changes. For example, both PP1 and PP2 had to react to an increase in support requests after migrating their software products into the cloud. PP1 decided to provide support with the help of their cloud provider, i.e. system no. 4, whereas PP2 decided to build support capabilities in-house, i.e. system no. 2.

While all software vendors had a clear goal for migrating their software products into the cloud, how to move to the cloud in order to achieve the goal was unclear for most of them. The majority of software vendors adopted an ad-hoc approach to plan from step to step. Adopting an ad-hoc approach and making adjustments during the migration is somewhat similar to making local adaptations and gradually rolling them out, the way adaptive STSs suggests. However, it is not the same. Neglecting local adaptations for internal decisions can be due to the fact that they are SMEs, where

even minor decisions can have an impact on the majority of employees (e.g. PP1 has around 30 employees).

From a customer perspective, some of the software vendors are planning to, or already are, utilising the concept of local adaptations to a greater extent. As it is easier to roll out software updates and new features in the cloud, PP5 tries to anticipate feature requests of customers and tests these with a limited number of customers before rolling them out to all customers. PP3 acts in a similar way and has a small group of core customers with whom they evaluate new products and features. These beta customers have been with PP3 for a long time and a great amount of trust has been established between the parties. PP2 has the potential to utilise local adaptations in the future. By developing a control panel where they can enable and disable features for certain customers, they could also use the control panel for evaluating new features with a small group of customers, say 5% of the customers. If the feature proves to be valued by customers it can be gradually rolled out to all customers (first 10% of customers, then 30%, and so on). The concept of local adaptations, however, needs to be tested further before its usefulness can be evaluated.

The FRAM analysis with PP1 in chapter 6 was also able to further illustrate concepts of adaptive STSs that the multi-stage study alluded to. The changes PP1 carried out while migrating their software product into the cloud were described as immediate actions the software vendor had to take. The immediate actions were concentrated on a few functions, thus the changes were locally. After having adopted cloud computing successfully the software vendor now gradually rolls out these adaptations to other parts of the company. Hence, the theory of adaptive STSs captures how companies adapt to new technologies appropriately. The exemplification of local adaptations in part refines the findings from the multi-stage study. The FRAM analysis with PP1 has shown that local adaptations can be useful in carrying out organisational changes that are necessary to adopt a new technology successfully.

## **9.2 Similarities of multi-stage study to packaged software development**

The analysis of the first and second stage of the multi-stage study has revealed that the majority of PPs underwent a similar transformation as early companies who

moved from developing bespoke software solutions to offering more generic ERP solutions in the 1990's (Buxmann et al. 2013; Brown 2013; Pollock & Williams 2008).

By migrating their software products into the cloud, the software vendors moved from developing bespoke software solutions towards offering a more generic product in similar ways as ERP development companies. When developing bespoke software solutions, the software vendor was clear about the requirements the customer had (at least in theory, in practice this is often not the case due to communication issues). A more generic product requires the capturing of a wider range of customers. In addition, the software vendor becomes responsible for deciding which requirements to include in the product. Furthermore, the software vendor needs to simultaneously include requirements for current customers *and* future customers. The challenge between accommodating current and future customers, however, has caused many companies that adopted technological discontinuities to fail, as current customers are sometimes reluctant to adopt a new technology and therefore provide a false picture of requirements (Christensen 2013).

The different approaches taken by the PPs shows how they find the best solution to this challenge. PP5 is trying to anticipate product features instead of just relying on their customers to tell them what they need. PP1 and PP2, in contrast, collect requirements over a longer period of time to decide which will be implemented. PP2 implemented a control panel, where they activate and deactivate features for specific customers. The control panel helps them to accommodate as many customers as possible. PP1, on the other hand, made the conscious decision to have only one version of their product where all features are available to all customers. Concentrating on one version could mean that they lose some customers but gain in efficiencies due to easier support, version control, etc. PP3 and PP4 are working closely with existing customers, that used their products before the cloud, to help them migrate the product into the cloud. PP3 is heavily relying on three customers that evaluate beta versions and provide feedback. In return, these customers will get a discount once the cloud version is ready. PP4 is working with other software vendors and their cloud provider to migrate their product. PP3 and PP4 face the potential

problem of being too closely attached to current customers that might distort the requirements that are important for a cloud product.

### **9.3 Similarities of cloud computing to IT Outsourcing**

There are similar lessons from IT Outsourcing that are potentially applicable to cloud computing. Surveys by Computer Economics (2014) and CA Technologies (2014) show that companies increasingly bring IT functions back in-house after a time of trying to outsource as much as possible (some talked about a ratio of 80:20 in favour of outsourcing). The reasons for retaining IT functions in-house are diverse. Two lessons, however, stand out that are potentially applicable to cloud computing. First, companies and IT departments are required to become more flexible and agile in their application and service development. Many companies made the experience that agility and outsourcing do not mix. If the outsourcing partner is not as agile as the outsourcer the whole approach falls apart because the outsourcer will always have to wait for the outsourcing partner to deliver. This lesson is applicable to cloud computing as cloud users only get what the cloud provider is offering. If they require additional functionality (e.g. a specific type of database) it can take a long time until the cloud provider has developed this functionality (especially if it is a niche cloud provider as in the case of PP1 who is planning to use a major cloud provider in addition to their niche provider as the major provider supports more types of database systems). In addition, it can be expensive as most cloud computing cost models are only cheaper than traditional data centres as long as the rented computing resources are within the estimates.

Second, today the development of new applications and services is more holistic. Developers are not just involved in developing and testing new applications and services. They are also responsible for tailoring applications to business processes. Furthermore, in today's complex and dynamic environment it is increasingly important not just to understand what something does (e.g. a technology or application), it is also important to understand how it does it. Understanding how a certain technology operates becomes particularly important in case something fails. When IT departments outsource functions they also outsource knowledge about related technologies and applications. This lesson is applicable to cloud computing as

companies using cloud computing need to have less detailed knowledge about IT infrastructure management because the cloud provider is taking care of most of the tasks associated with it (e.g. take PP2 whose developers are relieved that they do not have to look after hardware anymore). In case companies should decide at a later stage to bring IT functions back into the company it can be associated with high costs as not only hardware needs to be acquired but also people with appropriate skills. It remains to be seen, however, if cloud computing takes a similar path as IT Outsourcing.

#### **9.4 Threats to validity**

In addition to the limitations of the specific studies that were carried out as part of this thesis, there are general drawbacks that future work needs to address. First, when carrying out case studies it is difficult to generalise from individual cases. The multi-stage study tried to address this by talking to companies and industry experts outside of the study and sharing the results with a large group of people to gather feedback, e.g. through presenting the results at conferences. The evaluation study tested cFRAM with 14 companies, which is a relatively large number for such studies. In addition, the ideas of cFRAM and the handbook were shared in similar ways to the multi-stage study.

Throughout this thesis all parties of the cloud environment (cloud provider, software vendor, and end-customers) were investigated to some extent. In all cases, however, only one of the parties of a particular environment was interviewed. In other words, a cloud provider was interviewed which was not the cloud provider of one of the software vendors or end-customers that were interviewed. It would have been desirable to interview all parties in a particular environment to investigate the relationships between the parties in more detail, to explore how they affect each other, and the organisational and technical changes each party carries out. For example, during the multi-stage study all software vendors stated that the majority of their customers were happy with adopting a cloud product. The timeframe of the multi-stage study was too short, however, to conclude if this is actually the case or if the PPs only assume that. It was requested to interview some customers of the PPs. Interviewing customers was not possible for the majority of PPs (in fact, only one PP

agreed to it) as they all operate in a competitive environment and many customers do not want the use of the software vendor's product to be public knowledge.

## **9.5 Future work**

There are a number of areas that can help to further validate the findings from this work and expand them into other domains and settings.

Chapter 2 made a structured argument for the need for Resilience Engineering in IT. It defined and described use and technology uncertainty to provide a way of capturing issues that approaches designed to increase resilience in IT need to overcome. The multi-stage study (chapters 4 and 5) and FRAM analysis (chapter 6) were able to illustrate some of the aspects of use and technology uncertainty in practice and how companies dealt with them. Further studies are necessary, particularly in different domains and for different technological discontinuities, e.g. smartphones, to understand use and technology uncertainty in more detail. Understanding the effects of use and technology uncertainty in more detail could help extend the CCs for the IT industry developed in section 8.2.1.

The methodology for the multi-stage study allowed the collection of data that explained in great detail how the PPs migrated (some of) their software products into the cloud. Thus, the methodology might be useful for similar studies, possibly in relation to other technologies such as smartphones. Before concluding the usefulness of the methodology, however, it is necessary to validate it as it was only used within one study and five companies, a relatively small sample. Validating the methodology is difficult as, in contrast to other domains, it is not possible to carry out controlled experiments. The PPs adopted cloud computing successfully but it is not possible to conclude if it was due to the actions they took while adopting cloud computing or because they developed a feature many customers have been waiting for. As the data collected by this study is not conclusive in this regard, the methodology does not provide a mechanism for capturing it. It would be necessary to further extend the multi-stage study and interview many more companies that plan to migrate their software products into the cloud. The higher the number of participants, the less likely it is that all effects are due to feature development. If the methodology of the multi-

stage study still produces accurate results it should be applied to similar technologies, e.g. smartphones.

cFRAM has been specifically developed for cloud computing due to its viewpoints. An area for future work would be the testing of cFRAM for other technologies. It is reasonable to assume that the steps of a cFRAM analysis work for other technologies, as they are based on FRAM which has been tested in various settings. The viewpoints, however, probably need to be adapted. As the viewpoints are a result of the multi-stage study, this area of future work could go in tandem with the suggestions on validating the methodology. Regarding the adaptation of cFRAM to a different technology or setting, only one page in the cFRAM handbook would need to be changed (the page where the four viewpoints are described).

This work related capabilities and the theory of skill-, rule-, and knowledge-based behaviour (SRK) in a general sense by arguing that adapting capabilities requires employees to adopt knowledge-based behaviour. Therefore, capabilities can be used a communication tool to inform employees about the organisational changes technological discontinuities require. Chapter 8 was able to confirm that companies think that capabilities are an appropriate communication tool and can summarise organisational changes. In future work it might be worth relating capabilities and SRK more closely by investigating the impact of new technologies on individual employees. Investigating the impact on individual employees could enable a deeper understanding of how actions by employees influence the adaptation of capabilities. Similar studies to the multi-stage study could be carried out. Instead of interviewing high-level employees such as the Managing Director it would be necessary to interview employees in other roles, e.g. software developers, technicians, etc.

A final suggestion relates to the identification of performance variabilities and ways to dampen them. Learning new methods like FRAM require a lot of effort from companies. In addition, creating accurate FRAM models can require many people within a company to participate. Hence, it might be worth exploring if FRAM can also be used for metrics other than resilience. It is reasonable to assume that this is a possibility as the FRAM's main element is functions, which are very generic. One

metric that could provide a start for enhancing the usefulness of a FRAM analysis is *fifo*—fan-in/fan-out. *Fifo* is commonly used in software engineering to understand the complexity of software programs. The fan-in number of a module represents the number of immediate parent modules. The fan-out number of a module represents the number of immediate subordinate modules. Implementing *fifo* into FRAM would mean making use of the couplings between functions to define the fan-in and fan-out numbers. Using *fifo* could, for example, allow companies to understand how complex adapting one function to a new technology might be by analysing the couplings of that function in detail (i.e. defining the fan-in and fan-out numbers). Thus, *fifo* could also help understanding and communicating how adaptations will ripple through the entire company. It is important to note, however, that all metrics in FRAM, including performance variabilities, only apply to instantiations of a FRAM model.

## Bibliography

- Afuah, A., 2000. How much do your co-opetitors' capabilities matter in the face of technological change? *Strategic Management Journal*, 21(3), pp.387–404.
- Alshamaila, Y., Papagiannidis, S. & Li, F., 2013. Cloud computing adoption by SMEs in the north east of England. *Journal of Enterprise Information Management*, 26(3), pp.250–275.
- Anderson, S. & Felici, M., 2012a. Risks, Technologies and Societies. In S. Anderson & M. Felici, eds. *Emerging Technological Risk - Underpinning the Risk of Technology Innovation*. London: Springer London, pp. 3–27.
- Anderson, S. & Felici, M., 2012b. Unbound Technology. In S. Anderson & M. Felici, eds. *Emerging Technological Risk - Underpinning the Risk of Technology Innovation*. London: Springer London, pp. 31–51.
- Åström, K.J. & Wittenmark, B., 2011. *Computer-controlled systems: theory and design*, Courier Dover Publications.
- Badham, R., Clegg, C. & Wall, T., 2000. Socio-technical theory. In W. Karwowski, ed. *Handbook of Ergonomics*. New York, NY, USA: John Wiley & Sons.
- Baker, J., 2012. The Technology–Organization–Environment Framework. In Y. K. Dwivedi, M. R. Wade, & S. L. Schneberger, eds. *Information Systems Theory SE - 12*. Integrated Series in Information Systems. Springer New York, pp. 231–245.
- Baldwin, H., 2013. Chasing Shadows: What's Really Going On With Rogue IT. *Forbes*.
- Barney, J., 1991. Firm resources and sustained competitive advantage. *Journal of Management*, 17(1), pp.99–120.
- Baskerville, R. & Wood-Harper, A.T., 1998. Diversity in information systems action research methods. *European Journal of Information Systems*, 7(2), pp.90–107.
- Baxter, G. et al., 2012. The ironies of automation: still going strong at 30? In *ECCE'12 Proceedings of the 30th European Conference on Cognitive Ergonomics*. pp. 65–71.
- Baxter, G. & Sommerville, I., 2011. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23, pp.4–17.
- Belmonte, F. et al., 2011. Interdisciplinary safety analysis of complex socio-technological systems based on the functional resonance accident model: An application to railway traffic supervision. *Reliability Engineering and System Safety*, (96), pp.237–249.

- Benson, R.J., Ribbers, P.M. & Blitstein, R.B., 2014. *Trust and Partnership: Strategic IT Management for Turbulent Times*, John Wiley & Sons.
- Bertalanffy, L. von, 1969. *General system theory: foundations, development, applications*, New York: Braziller.
- Bharadwaj, A.S., 2000. A resource-based perspective on information technology and firm performance: an empirical investigation. *MIS Quarter*, 24(1), pp.169–196.
- Bharadwaj, S.S. & Lal, P., 2012. Exploring the impact of cloud computing adoption on organizational flexibility: A client perspective. In *Proceedings of 2012 International Conference on Cloud Computing Technologies, Applications and Management, ICCCTAM 2012*. pp. 121–131.
- Bigdoli, H., 2011. Successful Introduction of Cloud Computing into Your Organization: A Six-Step Conceptual Mode. *Journal of International Technology & Information Management*, 20(1), p.21.
- Borghoff, U.M. & Schlichter, J.H., 2000. *Computer-Supported Cooperative Work*, Springer Berlin Heidelberg.
- Bower, J.L. & Christensen, C.M., 1995. Disruptive technologies: catching the wave. *Harvard Business Review*, 73(1), pp.43–53.
- Bradley, C. & Matson, E., 2011. Putting strategies to the test. *McKinsey Quarterly*, (January), pp.1–8.
- Brandenburger, A.M. & Nalebuff, B.J., 1995. The Right Game: Use Game Theory to Shape Strategy. *Harvard Business Review*, 28(5), pp.57–71.
- Brender, N. & Markov, I., 2013. Risk perception and risk management in cloud computing: Results from a case study of Swiss companies. *International Journal of Information Management*, 33(5), pp.726–733.
- Brown, A.W., 2013. *Enterprise Software Delivery: Bringing Agility and Efficiency to the Global Software Supply Chain*, Addison Wesley.
- Brynjolfsson, E. & McAfee, A., 2014. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*, W. W. Norton & Company.
- Brynjolfsson, E. & Saunders, A., 2013. *Wired for Innovation: How Information Technology Is Reshaping the Economy*, MIT Press.
- Bughin, J., 2012. Wiring the open-source enterprise. *McKinsey Quarterly*, (January), pp.1–4.
- Burton, B. & Willis, D.A., 2014. *Gartner's Hype Cycle Special Report for 2014*. Gartner.

- Buxmann, P., Diefenbach, H. & Hess, T., 2013. *The Software Industry - Economic Principles, Strategies, Perspectives*, Springer.
- CA Technologies, 2014. *How to survive and thrive in the application economy*. CA Technologies.
- Cabrerizo, F.J. et al., 2010. Analyzing consensus approaches in fuzzy group decision making: advantages and drawbacks. *Soft Computing*, 14(5), pp.451–463.
- Calinescu, R. et al., 2010. *The UK Large-Scale Complex IT Systems (LSCITS) Initiative*. The UK Research & Training Initiative.
- De Carvalho, P.V.R., 2011. The use of Functional Resonance Analysis Method (FRAM) in mid-air collision to understand some characteristics of the air traffic management system resilience. *Reliability Engineering and System Safety*, (96), pp.1482–1498.
- Checkland, P., 1999. *Systems Thinking, Systems Practice: Includes a 30 Year Retrospective* New editio., John Wiley & Sons.
- Chosokabe, M. et al., 2015. Effects of Small Group Discussion: Case Study of Community Disaster Risk Management in Japan. In B. Kamiński, G. E. Kersten, & T. Szapiro, eds. *Outlooks and Insights on Group Decision and Negotiation SE - 1*. Lecture Notes in Business Information Processing. Springer International Publishing, pp. 3–12.
- Christensen, C.M., 2013. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail (Management of Innovation and Change)* Reprint ed., Harvard Business Review Press.
- Christensen, C.M. & Overdorf, M., 2000. Meeting the Challenge of Disruptive Change. *Harvard Business Review*, 78(2), pp.66–76.
- Clegg, C. et al., 1996. Tools to incorporate some psychological and organizational issues during the development of computer-based systems. *Ergonomics*, 39(3), pp.482–511.
- Computer Economics, 2014. *IT Outsourcing Statistics 2014/2015: Outsourcing Trends and Cost/Service Experiences for 11 Key IT Functions*. Computer Economics.
- Cox, M.W. & Alm, R., 2008. You Are What You Spend. *The New York Times*, p.1.
- Craig, D., Kanakamedala, K. & Tinaikar, R., 2007. The next frontier in IT strategy: A McKinsey Survey. *McKinsey Quarterly*, (Spring), pp.1–3.
- Crowston, K. & Myers, M.D., 2004. Information technology and the transformation of industries: Three research perspectives. *Journal of Strategic Information Systems*, 13, pp.5–28.

- Dalpiaz, F., Giorgini, P. & Mulopoulos, J., 2013. Adaptive socio- technical systems: a requirements-based approach. *Requirements Engineering*, 18, pp.1–14.
- Dobbs, R. et al., 2014. Management intuition for the next 50 years. *McKinsey Quarterly*, (September), pp.1–13.
- Downes, L. & Nunes, P., 2014. *Big Bang Disruption: Strategy in the Age of Devastating Innovation*, Penguin Publishing Group.
- Dreyfus, H.L., 1987. *What Computers Can't Do: The Limits of Artificial Intelligence Revised.*, Harpercollins.
- Dreyfus, H.L., 1992. *What Computers Still Can't Do: A Critique of Artificial Reason*, MIT Press.
- Drucker, P.F., 1967. The manager and the moron. *McKinsey Quarterly*, (December).
- Duck, J.D., 1993. Managing change: the art of balancing. *Harvard Business Review*, (November-December), pp.109–118.
- Eason, K., 2007. Local sociotechnical systems development in the NHS National Programme for Information Technology. *Journal of Information Technology*, 22, pp.257–264.
- Eisenhardt, K.M., 1989. Building Theories from Case Study Research. *The Academy of Management Review*, 14(4), pp.532–550.
- Eisenhardt, K.M. & Martin, J.A., 2000. Dynamic capabilities: What are they? *Strategic Management Journal*, 21(10), pp.1105–1121.
- El-Gazzar, R., 2014. A Literature Review on Cloud Computing Adoption Issues in Enterprises. In B. Bergvall-Kåreborn & P. Nielsen, eds. *Creating Value for All Through IT SE - 14*. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 214–242.
- Engelhardt, M. & Bain, L.J., 1986. On the Mean Time between Failures for Repairable Systems. *IEEE Transactions on Reliability*, 35.
- ENISA, 2014. *Annual Incident Reports 2013 - Analysis of Article 13a annual incident reports*. European Union Agency for Network and Information Security.
- Feeney, D.F. & Willcocks, L.P., 1998. Core IS capabilities for exploiting information technology. *Sloan Management Review*, 39(3), pp.9–21.
- Frost, B. & Mo, J.P.T., 2014. System Hazard Analysis of a Complex Socio-Technical System: The Functional Resonance Analysis Method in Hazard Identification. In *Australian System Safety Conference*. Melbourne, Australia.
- Galunic, C. & Eisenhardt, K.M., 2001. Architectural Innovation and Modular Corporate Forms. *The Academy of Management Journal*, 44(6), pp.1229–1249.

- Garrison, G., Wakefield, R.L. & Kim, S.H., 2015. The effects of IT capabilities and delivery model on cloud computing success and firm performance for cloud supported processes and operations. *Int J. Information Management*, 35(4), pp.377–393.
- Girard, J. & Girard, J.A., 2009. *A Leader's Guide to Knowledge Management: Drawing on the Past to Enhance Future Performance*, Business Expert Press.
- Glaser, B.G. & Holton, J., 2004. Remodeling Grounded Theory [80 paragraphs]. *Forum: Qualitative Social Research*, 5(2).
- Glaser, B.G. & Strauss, A.L., 2009. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Transaction.
- Grant, R.M., 1996. Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration. *Organization Science*, 7(4), pp.375–387.
- Grant, R.M., 1991. The Resource-Based-Theory of Competitive Advantage: Implications for Strategy Formulation. *California Management Review*, (Spring), pp.114–135.
- Van Grembergen, W., De Haes, S. & Guldentops, E., 2004. Structures, Processes and Relational Mechanisms for IT Governance. *W. Van Grembergen Strategies for Information Technology Governance*, pp.1–36.
- Gupta, P., Seetharaman, A. & Raj, J.R., 2013. The usage and adoption of cloud computing by small and medium businesses. *International Journal of Information Management*, 33(5), pp.861–874.
- Guvendiren, K., Brinkkemper, S. & Jansen, S., 2014. Productization of an IT Service Firm. In C. Lassenius & K. Smolander, eds. *Software Business. Towards Continuous Value Delivery SE - 9*. Lecture Notes in Business Information Processing. Springer International Publishing, pp. 115–131.
- Handel, M.J. & Poltrock, S., 2011. Working around official applications. *Proceedings of the ACM 2011 conference on Computer supported cooperative work - CSCW '11*, pp.309–312.
- Herrera, I.A. & Woltjer, R., 2010. Comparing a multi-linear (STEP) and systemic (FRAM) method for accident analysis. *Reliability Engineering and System Safety*, (95), pp.1269–1275.
- Herritt, R., 2014. When Technology Ceases to Amaze. *The New Atlantis*, (Winter), pp.121–131.
- Holling, C.S., 1973. Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics*, 4(1), pp.1–23.

- Hollnagel, E., 2012a. *An Application of the Functional Resonance Analysis Method (FRAM) to Risk Assessment of Organisational Change*. Swedish Radiation Safety Authority.
- Hollnagel, E., 2007. Flight decks and free flight: Where are the system boundaries? *Applied Ergonomics*, 38, pp.409–416.
- Hollnagel, E., 2012b. *FRAM: The Functional Resonance Analysis Method*, Ashgate.
- Hollnagel, E. et al., 2011. *Resilience Engineering in Practice: A Guidebook*, Ashgate.
- Hollnagel, E., 2009. *The ETTO Principle: Efficiency-Thoroughness Trade-Off*, Ashgate.
- Hollnagel, E., Hounsgaard, J. & Colligan, L., 2014. *FRAM - a handbook for the practical use of the method* First edit., Centre for Quality, Region of Southern Denmark.
- Hollnagel, E. & Speziali, J., 2008. *Study on Developments in Accident Investigation Methods: A Survey of the “State-of-the-Art.”* SKI.
- Hollnagel, E. & Woods, D.D., 2005. *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering: An Introduction to Cognitive Systems Engineering*, CRC Press.
- Hollnagel, E., Woods, D.D. & Levenson, N., 2006. *Resilience Engineering: Concepts And Precepts*, Ashgate.
- Hopkins, R. & Jenkins, K., 2008. *Eating the IT Elephant: Moving from Greenfield Development to Brownfield*, IBM Press; 1 edition.
- HP, 2013. *HP 2012 Cyber Risk Report*. HP.
- Ignatiadis, I. & Nandhakumar, J., 2007. The impact of enterprise systems on organizational resilience. *Journal of Information Technology*, 22, pp.36–43.
- ISACA, 2014. *Cloud Governance: Questions Boards of Directors Need to Ask*. Information Systems Audit and Control Association, Inc.
- ISACA, 2012. *Guiding Principles for Cloud Computing Adoption and Use*. Information Systems Audit and Control Association, Inc.
- Kaplan, R.S. & Norton, D.P., 2007. Using the balanced scorecard as a strategic management system. *Harvard Business Review*, 85.
- Khajeh-Hosseini, A. et al., 2012. The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise. *Softw. Pract. Exper.*, 42(4), pp.447–465.

- Khanagha, S. et al., 2013. Management Innovation and Adoption of Emerging Technologies: The Case of Cloud Computing. *European Management Review*, 10, pp.51–67.
- Klein, G.A., Calderwood, R. & MacGregor, D., 1989. Critical decision method for eliciting knowledge. *IEEE Transactions On Systems Man And Cybernetics*, 19(3), pp.462–472.
- Klein, K.H. & Myers, M.D., 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1), pp.67–93.
- Koller, T., Goedhart, M. & Wessels, D., 2010. *Valuation: Measuring and Managing the Value of Companies* 5th ed., John Wiley & Sons.
- Kotter, J.P., 2009. Leading change: why transformation efforts fail. *IEEE Engineering Management Review*, 37(3).
- Laprie, J.C., 2008. From Dependability to Resilience. In *International Conference on Dependable Systems & Networks (DSN 2008)*. pp. G8–G9.
- Lazar, J., Feng, J.H. & Hochheiser, H., 2010. *Research Methods in Human-Computer Interaction*, John Wiley & Sons.
- Leedy, P.D. & Ormrod, J.E., 2005. *Practical research: Planning and design* 8th ed., Upper Saddle River: NJ: Prentice Hall.
- Leonard-Barton, D., 1992. Core capabilities and core rigidities: A paradox in managing new product development. *Strategic Management Journal*, 13(Special issue), pp.111–125.
- Leonard-Barton, D. & Kraus, W.A., 1985. Implementing new technology. *Harvard Business Review*, 63(6), pp.102–110.
- Leveson, N., 2004. A new accident model for engineering safer systems. *Safety Science*, 42(4), pp.237–270.
- Leveson, N., 2012. *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press.
- Liu, H. & Orban, D., 2008. GridBatch: Cloud Computing for Large-Scale Data-Intensive Batch Applications. In *8th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2008)*. Lyon, France.
- Lu, Y. & Ramamurthy, K. (Ram), 2011. Understanding the link between information technology capability and organizational agility: an empirical examination. *MIS Quarterly*, 35(4), pp.931–954.

- Ludwig, G. & Pemberton, J., 2011. A managerial perspective of dynamic capabilities in emerging markets: The case of the Russian steel industry. *Journal for East European Management Studies*, 16(3), pp.215–236.
- Lundberg, J., Rollenhagen, C. & Hollnagel, E., 2009. What-You-Look-For-Is-What-You-Find - The consequences of underlying accident models in eight accident investigation manuals. *Safety Science*, 47(10), pp.1297–1311.
- Madni, A.M. & Jackson, S., 2009. Towards a Conceptual Framework for Resilience Engineering. *IEEE Systems Journal*, 3(2), pp.181–191.
- Maglyas, A., Nikula, U. & Smolander, K., 2012. Comparison of Software Product Management Practices in SMEs and Large Enterprises. In M. Cusumano, B. Iyer, & N. Venkatraman, eds. *Software Business SE - 2*. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 15–26.
- Manyika, J., Chui, M. & Bughin, J., 2013. Disruptive technologies: Advances that will transform life, business, and the global economy. *McKinsey Global Institute*, (May), p.163.
- Martin, M. & Ching, R., 1999. Information Technology (IT) Change Management. In *America Conference on Information Systems (AMCIS)*. pp. 103–105.
- McDonald, M. & Aron, D., 2013. *Hunting and Harvesting in a Digital World: The 2013 CIO Agenda - Executive Summary*. Gartner.
- Mell, P. & Grance, T., 2009. *The NIST Definition of Cloud Computing*. National Institute of Standards and Technology.
- Mercuri, M. et al., 2014. *Failsafe: Guidance for Resilient Cloud Architectures*. Microsoft.
- Miles, M.B. & Huberman, A.M., 1994. *Qualitative Data Analysis: An Expanded Sourcebook* Second Edi., Sage Publications, Inc.
- Murray-Webster, R. (Office of G.C., 2010. *The Management of Risk: Guidance for Practitioners* 3rd Editio., Stationary Office.
- Myers, M., 2004. Hermeneutics in Information Systems Research. In J. Mingers & L. P. Willcocks, eds. *Social Theory and Philosophy for Information Systems*. Chicester: John Wiley & Sons, pp. 103–128.
- Nemeth, C.P. & Herrera, I., 2015. Building change: Resilience Engineering after ten years. *Reliability Engineering & System Safety*, 141(0), pp.1–4.
- Northrop, L. et al., 2006. *Ultra-large-scale systems - The Software Challenge of the Future*. Software Engineering Institute.
- Nuseibeh, H., 2011. Adoption of Cloud Computing in Organizations. *Information Systems Journal*, pp.1–8.

- Orlikowski, W.J., 1993. CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development. *MIS Quarterly*, 17(3), pp.309–340.
- Pan, S.L. et al., 2007. The dynamics of implementing and managing modularity of organizational routines during capability development: insights from a process model. *IEEE Transactions on Engineering Management*, 54(4), pp.800–813.
- Pan, S.L. & Tan, B., 2011. Demystifying case research: A structured-pragmatic-situational (SPS) approach to conducting case studies. *Information and Organization*, 21, pp.161–176.
- Paulk, M.C. et al., 1993. *Capability Maturity Model for Software, Version 1.1*. Carnegie Mellon University.
- Pearlson, K.E. & Saunders, C.S., 2010. *Managing and using information systems: a strategic approach* 4th ed., Hoboken, N.J.: Wiley.
- Perrow, C., 1984. *Normal Accidents*, New York, NY, USA: Basic Books.
- Peterson, R.R., 2004. Integration Strategies and Tactics for Information Technology Governance. In *Strategies for Information Technology Governance*. Idea Group Publishing, pp. 37–80.
- Pollock, N. & Williams, R., 2008. *Software and Organisations: The Biography of the Enterprise-Wide System Or How SAP Conquered the World*, Taylor & Francis.
- Porter, M.E., 2004. *Competitive Advantage: Creating and Sustaining Superior Performance*, Free Press.
- Porter, M.E., 2008. The five competitive forces that shape strategy. *Harvard Business Review*, 86(1), pp.78–93, 137.
- Rasmussen, J., 1997. Risk management in a dynamic society: A modelling problem. *Safety Science*, 27(2-3), pp.183–213.
- Rasmussen, J., 1983. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13, pp.257–266.
- Ravichandran, T. & Lertwongsatien, C., 2005. Effect of Information Systems Resources and Capabilities on Firm Performance: A Resource-Based Perspective. *Journal of Management Information Systems*, 21(4), pp.237–276.
- Reason, J., Hollnagel, E. & Paries, J., 2006. Revisiting the «Swiss Cheese» Model of Accidents. *Journal of Clinical Engineering*, (13).
- Righi, A.W., Saurin, T.A. & Wachs, P., 2015. A systematic literature review of resilience engineering: Research areas and a research agenda proposal. *Reliability Engineering & System Safety*, 141(0), pp.142–152.

- Rimal, B.P. et al., 2011. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. *Journal of Grid Computing*, 9(1), pp.3–26.
- Roberts, D., 2013. *Unleashing the Power of IT: Bringing People, Business, and Technology Together* 2nd ed., John Wiley & Sons.
- Roberts, K.H., 1990. Some Characteristics of One Type of High Reliability Organization. *Organization Science*, (1), pp.160–176.
- Rochwerger, B. et al., 2009. The Reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4), pp.1–11.
- Rogers, E.M., 2003. *Diffusion of Innovations* 5th ed., Simon & Schuster International.
- Rosenbloom, R.S. & Christensen, C.M., 1994. Technological Discontinuities, Organizational Capabilities, and Strategic Commitments. *Industrial and Corporate Change*, 3(3), pp.655–685.
- Rosenzweig, P., 2007. Misunderstanding the Nature of Company Performance: The Halo Effect and Other Business Delusions. *California Management Review*, 49(4), pp.6–20.
- Ross, J.W., Beath, C.M. & Goodhue, D.L., 1996. Developing long-term competitiveness through IT assets. *Sloan Management Review*, 38(1), pp.31–42.
- Skok, M., 2013. 2013 Future of Cloud Computing Survey Reveals Business Driving Cloud Adoption in Everything as a Service Era. *North Bridge*.
- Sommerville, I. & Sawyer, P., 1997. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, 3, pp.101–130.
- Sommerville, I., Sawyer, P. & Viller, S., 1998. Viewpoints for requirements elicitation: a practical approach. *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on*, pp.74–81.
- Spedale, S., 2003. Technological Discontinuities: Is Co-operation an Option? *Long Range Planning*, 36(3), pp.253–268.
- Strauss, A. & Corbin, J., 1994. Grounded Theory Methodology - An Overview. In N. K. Denzin & Y. S. Lincoln, eds. *Handbook of Qualitative Research*. Sage Publications, Inc, pp. 273–285.
- Suchman, L. a., 1983. Office procedure as practical action: models of work and system design. *ACM Transactions on Information Systems*, 1(4), pp.320–328.
- Sultan, N.A., 2011. Reaching for the “cloud”: How SMEs can manage. *International Journal of Information Management*, 31(3), pp.272–278.

- Susman, G.I. & Evered, R.D., 1978. An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly*, (23), pp.582–603.
- Teece, D.J., 2011. *Dynamic Capabilities and Strategic Management: Organizing for Innovation and Growth* Reprint Ed., OUP Oxford.
- Teece, D.J., Pisano, G. & Shuen, A., 1997. Dynamic Capabilities and Strategic Management. *Strategic Management Journal*, 18(7), pp.509–533.
- Timmons, S., 2003. A failed panopticon: surveillance in nursing practice via new technology. *New Technology, Work and Employment*, 18(2), pp.143–153.
- Tolerance, F., 1992. *Dependability: Basic Concepts and Terminology* J.-C. Laprie, ed., Springer-Verlag New York, Inc.
- Trist, E., 1981. The evolution of socio-technical systems - a conceptual framework and an action research program. In *Conference on Organizational Design and Performance*. p. 67.
- Tushman, M.L. & Anderson, P., 1986. Technological Discontinuities and Organizational Environments M. Pettigrew Andrew, ed. *Administrative Science Quarterly*, 31(3), pp.439–465.
- Voß, A. et al., 2006. Understanding and Supporting Dependability as Ordinary Action. In K. Clarke et al., eds. *Trust in Technology: A Socio-Technical Perspective SE - 9*. Computer Supported Cooperative Work. Springer Netherlands, pp. 195–216.
- Wade, M. & Hulland, J., 2004. The Resource-Based View and Information Systems Research: Review, Extension and Suggestions for Future Research. *MIS Quarterly*, 28(1), pp.107–142.
- Walker, G.H. et al., 2008. A review of sociotechnical systems theory: A classic concept for new command and control paradigms. *Theoretical Issues in Ergonomics Science*, (9), pp.479–499.
- Walsham, G., 2006. Doing interpretive research. *European Journal of Information Systems*, 15, pp.320–330.
- Werfs, M. et al., 2013. Migrating Software Products to the Cloud: An Adaptive STS Perspective. *Journal of International Technology & Information Management*, 22, pp.37–54.
- Werfs, M., 2012. *Services to the Cloud - Usability Study*. University of St Andrews.
- Werfs, M. & Baxter, G., 2013. Towards resilient adaptive socio-technical systems. *Proceedings of the 31st European Conference on Cognitive Ergonomics - ECCE '13*.

- Wernerfelt, B., 1984. A resource-based view of the firm. *Strategic Management Journal*, 5(2), pp.171–180.
- Winter, S.G., 2003. Understanding dynamic capabilities. *Strategic Management Journal*, 24(10), pp.991–995.
- Woods, D. & Dekker, S., 2000. Anticipating the effects of technological change: a new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science*, 1(3), pp.272–282.
- Zardari, S. & Bahsoon, R., 2011. Cloud Adoption: A Goal-oriented Requirements Engineering Approach. In *Proceedings of the 2Nd International Workshop on Software Engineering for Cloud Computing*. SECCLOUD '11. New York, NY, USA: ACM, pp. 29–35.

## Appendix A – List of multi-stage study first stage questions

Questions related to the plan phase of Figure 6:

- Why did the company want to use cloud computing?
  - Who proposed the use of cloud computing?
  - Can it be considered a Market-Pull or Technology-Push?
  - How does the company define cloud computing?
  - What will cloud computing be used for?
  - What did the decision process look like?
  - Was it clear what capabilities are/would needed (to be) developed?
  - Did the company go through distinct steps, consciously? (e.g. lifecycle)
- Did the company develop a road map/strategy? (for current and future features)

Questions related to the migrate phase of Figure 6:

- Were there differences in designing a product for the cloud compared to non-cloud environments?
  - Did the company incorporate functionalities of the cloud into requirements engineering to alter your product? E.g. offer functionalities that were not possible before
  - Did the work of the software engineers itself change? (processes, methods, skill set)
  - Did the IT perceive a loss of control to the cloud provider? If yes, how did the company deal with it?
  - Did the company specifically address SLAs, availability, or security in the design process or any other phase?
- Were there differences in developing a product for the cloud compared to non-cloud environments?
  - Did this affect the use of cloud computing in other company areas?

Questions related to the run phase of Figure 6:

- Are customers willing to adopt cloud computing?
  - Did all the customers accept the use of the cloud? Or did the company have two groups of customers in the end?
  - Was there a tension between the two groups of customers (cloud and non-cloud)?
  - Did the company support customers with migrating to the cloud?
  - Did the adoption of cloud computing adversely affect any part of the company? How did the company identify and deal with it?

- Did cloud computing affect the response to changing customer needs or business requirements?

## **Appendix B – List of multi-stage study second stage questions**

Questions related to *Customers* of the Balanced Scorecard:

- How has the response to customer requests changed?
  - How is it influencing internal business processes (e.g. customer acquisition costs)?
  - How is it influencing learning & growth (e.g. loyalty)?
  - How is it influencing financials (e.g. average revenue per customer)?

Questions related to *Internal business processes* of the Balanced Scorecard:

- Were overall agility and flexibility of the company affected by cloud computing?
- Were overall effectiveness and efficiency of the company affected by cloud computing?
- What company areas were affected?
- Did the external use of cloud computing trigger an internal use of it?

Questions related to *Learning & Growth* of the Balanced Scorecard:

- How is cloud computing helping your company to grow?
- Does your company have a learning culture?
- Is cloud computing affecting the innovation ability of your company?
- Did employees learn new skills on their own or was teaching necessary?

Questions related to *Financials* of the Balanced Scorecard:

- How is cloud computing affecting your financial planning?
- Are there financial planning benefits of using the cloud?
- Are there financial planning downsides of using the cloud?

## Appendix C – Grouping of micro themes

Macro theme	Micro theme	Part of tension:	Explanation
Planning	Gain competitive advantages	Tension 1	The aim is to use cloud computing to expand the customer base and thus make the business model more robust and attractive to customers
	Keep mission of company	Tension 1	Although cloud computing offers many opportunities the PPs explore these carefully by keeping the mission of their company; in the majority of cases this meant continuing to offer one product to all customers
	Steep learning curve	Tension 1 and 3	As cloud computing is a new technology all PPs had limited knowledge about what would be necessary to adopt it and use it to offer products to customers
Migrating	Increase efficiency of product development	Tension 2	Through the migration into the cloud the PPs acquired more customers that have differing demands; that makes it necessary to make design product development more efficiently; it depends on individual PPs if that makes operations leaner or more complex
	More responsibilities	Tension 2	In the cloud the PPs are required to have better support capabilities because they have more customers and host the product for them; this can make operations more complex, at least in the short term
Running	Build trust with providers and customers	Tension 3	Some PPs expressed that a few of their customers are still wary of adopting a cloud product. The PPs try to convince these customers by building trust between them and the cloud provider(s)
	In control of provision and updates	Tension 3	As the PPs maintain and update the product for the customers, the customers have to trust the PPs and the cloud provider(s)
Customer demands	Obstacle to sale reduced	Tension 1	Cloud computing can make it easier to expand the customer base in existing markets and move to new markets, nationally as well as internationally, thus making the progression of the business model easier
	Offer trials	Tension 1	By offering trials customers can be convinced to buy the product more easily, thus allowing the PPs to expand their customer base
	Standard set up process	Tension 2	Cloud computing allows the PPs to treat the majority of customers in the same way, at least during the set up process; at the same time, cloud computing has made it necessary to create a standard set up process due to the larger number of customers
	Robust backend processes	Tension 2	As the PPs are responsible for the dependable operation of the product, they have to have appropriate internal processes which can make them more complex
Internal processes	Focus on core tasks	Tension 2	By focusing on core tasks through the outsourcing of secondary tasks, the PPs are able to design internal

			processes in a more leaner way
	Less physical space	None	/
	Internal use of the cloud	None	/
Human resources management	Technology savvy employees	Tension 1	Not only the company as a whole has to understand cloud computing, but individual employees too; depending on how the employees appreciate cloud computing it can have an effect on the business model as the employees use cloud computing to develop new products, for example
	Learning environment	Tension 1 and 3	See above: Steep learning curve
	Prepare for disruptive events	Tension 2	See above: Robust backend processes

## Appendix D – Generic product lifecycle for FRAM analysis

The generic product lifecycle shown in Figure 28 represents an alternative to the generic customer lifecycle explained in section 6.3.2. In contrast to the generic customer lifecycle, the generic product lifecycle takes a product centric view and assumes that the decision a software vendor makes revolve around the impact on their product (not the customer as for the generic customer lifecycle). The generic product lifecycle and customer lifecycle were both presented to PP1. PP1 rejected the generic product lifecycle and decided the generic customer lifecycle was more accurately representing their approach. However, the generic product lifecycle might be appropriate for other software vendors. Therefore, the following paragraphs will briefly describe the four main functions of the generic product lifecycle.

The function *Instantiate product environment* can be considered as the function that starts the product lifecycle. Before the move to the cloud, the customer carries out this function. Thus the software has only a limited amount of control over this function. Once the function has been completed and its Output has been produced (Product environment ready) the function *Set up product* starts.

The function Set up product is responsible for installing the product and tailoring the product installation to the needs of the customer. When the Output of this function has

been produced (Product is set up) and the function is completed, the customer is able to use the product.

The function *Service product* is responsible for fixing bugs and enabling customers to use the product in the way they expect to. In other words, this function takes a short term view and is deals with day to day tasks that revolve around the products of the software vendor.

The function *Enhance product functionality* is responsible for taking a long-term view of the software vendor’s products. This function is about identifying opportunities to add product functionality or to develop entirely new products that customers might desire.

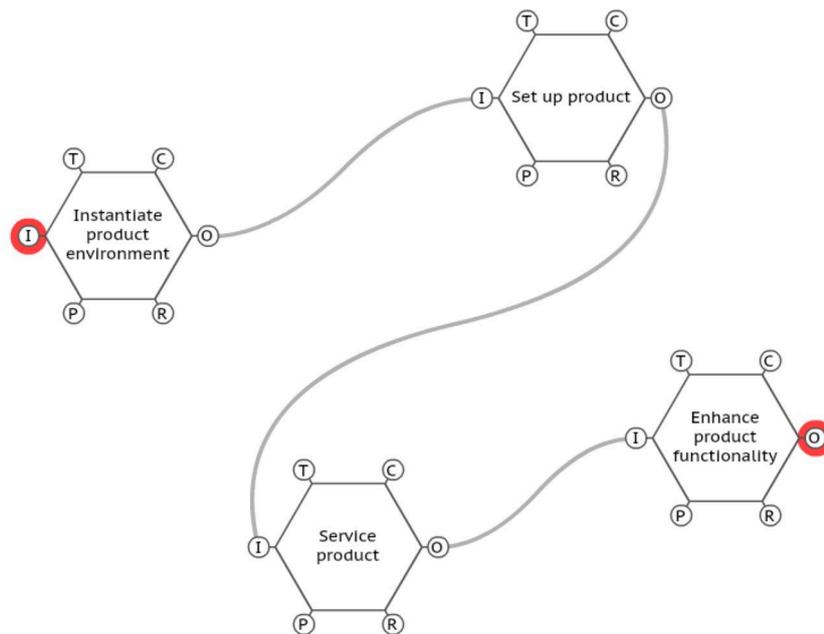


Figure 28 - Product lifecycle to structure a FRAM analysis (as an alternative to the customer lifecycle that was used with PPI)

## Appendix E – Final cFRAM handbook

If you have read chapters 7 & 8 there is no need to read the cFRAM handbook below. If you have only read chapter 7 you might want to read the cFRAM handbook below to see the improvements through the evaluation study presented in chapter 8. Please note that the cFRAM handbook refers to its own figure and table numbers. Due to formatting requirement of this thesis the one-page note sheet is longer than one page.

cFRAM

—

Using the Functional Resonance  
Analysis Method to plan  
organisational changes of software  
vendors that move to the cloud

—

Handbook

Version: 2

Marc Werfs  
University of St Andrews



## Table of contents

<b>Introduction.....</b>	<b>235</b>
<b>FRAM – A short introduction.....</b>	<b>236</b>
<b>cFRAM: A generic customer lifecycle to assist the data collection stages....</b>	<b>239</b>
<b>cFRAM: Collecting data (Step 1) .....</b>	<b>241</b>
<b>Example: Collecting data .....</b>	<b>242</b>
<b>cFRAM: Identifying performance variabilities (Step 2) .....</b>	<b>245</b>
<b>Example: Identifying performance variabilities .....</b>	<b>247</b>
<b>cFRAM: Identifying existing capabilities (Step 3) .....</b>	<b>249</b>
<b>Example: Identifying existing capabilities.....</b>	<b>251</b>
<b>cFRAM: Planning the move to the cloud (Step 4) .....</b>	<b>254</b>
<b>Example: Planning the move to the cloud.....</b>	<b>256</b>
<b>cFRAM: Using the cFRAM as a long-term planning tool .....</b>	<b>261</b>
<b>Applying the cFRAM throughout the migration.....</b>	<b>261</b>
<b>Accommodating the viewpoints during the migration.....</b>	<b>261</b>
<b>Preparing employees and customers for the cloud .....</b>	<b>262</b>
<b>Appendix A – Questions to stimulate discussions .....</b>	<b>263</b>
<b>cFRAM notes.....</b>	<b>265</b>

## Introduction

Cloud computing offers many opportunities to software vendors. It enables them, for example, to compete on a global level and to save costs by outsourcing secondary tasks. Cloud computing is, however, a technological discontinuity which means that companies are required to have complementary organisational changes in order to fully exploit the opportunities presented by cloud computing. The organisational changes are often overlooked, as they are intangible and sometimes invisible.

Cloud computing introduces software vendors to new kinds of risks. Software vendors take over responsibilities previously held by the customer, such as maintaining the product or installing updates. At the same time, software vendors outsource tasks to the cloud provider over which they have only a limited amount of control. The software vendor has to rely on the cloud provider to deliver the computing resources that were promised.

This handbook proposes that the notion of capabilities can help software vendors plan the organisational changes cloud computing requires and relate these to the technical adoption challenges. This will enable software vendors to exploit the opportunities presented by cloud computing while minimising the risks associated with a move to the cloud. As all software vendors are different and thus have different capabilities, there is no rigid model that can be followed to adopt cloud computing. *Capabilities* use resources in a structured way to carry out a specific task. In other words, they are organisational routines (IT infrastructure management is an example for a capability).

In this handbook, the Functional Resonance Analysis Method (FRAM) is extended to enable software vendors to understand how their existing capabilities suit cloud computing and, if necessary, what new capabilities need to be developed. The method's main elements are functions and aspects (that connect functions with each other). Functions within FRAM contain everything to carry out a task. Similar to capabilities, functions can contain resources, such as machines, documents, personnel, etc. These characteristics of FRAM make it a suitable method to investigate capabilities.

The FRAM's advantage over other methods is its ability to analyse the *resilience* of the software vendor. Resilience can be defined as the dependability of functions during changing conditions. Understanding the resilience of the software vendor and *how it can be increased through the adaptation of capabilities* can enable the company to react to and anticipate changes faster. In light of today's complex and connected systems, such as cloud computing, companies have no choice but to learn how to react to and anticipate changes faster.

FRAM is also used as the basic model because it is flexible, does not require a lot of training, and lets companies focus on those issues that are relevant to their particular circumstances. Thus, this handbook enables software vendors to find a solution best suited to their needs.

The extension of FRAM as described in this handbook addresses specifically capabilities and thus will be referred to as cFRAM in the sequel. cFRAM targets Managing Directors or leaders of product development of SME software vendors that want to explore the possibility of distributing their software products via the cloud. To understand the complementary organisational changes cloud computing requires it is important to consider the viewpoints of different departments. Hence, the method is designed to be applied during group discussions with people from different departments where the Managing Director or leader of product development takes over the role as moderator and navigates through the steps of a cFRAM analysis.

cFRAM has four steps. In the first step, data is collected to create a FRAM model that shows the functions of the software vendor necessary for the on-site version of the software product. In the second step, performance variabilities are identified that inform how the resilience of the software vendor can be increased. In the third step, the FRAM model is analysed to identify the capabilities that reside in the software vendor and what functions they reside in. In the fourth step, the move to the cloud is planned. While planning the move the software vendor has to investigate how it should be done, i.e. how the functions in the FRAM model need to be adapted, how existing capabilities can enhance or inhibit a move to the cloud, and, if necessary, what new capabilities need to be developed.

The next section provides a short introduction to the FRAM to explain its key concepts. The third section introduces a generic customer lifecycle that can be used to guide the analysis. The following four sections explain each of the steps of a cFRAM analysis as outlined above. At the end of each section, an example is shown to illustrate how the steps can be applied in practice. The example was created by applying the cFRAM to a software vendor that migrated their software product into the cloud retrospectively. The final section explains how the cFRAM can be used for planning the move to the cloud in the long-term. The final page of this handbook provides an overview of the cFRAM that can be handed out to participants.

## FRAM – A short introduction

The FRAM consists of *functions* that are connected with each other through *aspects*. The *functions* are abstractions to capture work routines and related resources, tangible and intangible. The six aspects are **I**nput, **O**utput, **T**ime, **C**ontrol, **P**recondition, and **R**esources. Figure 1 shows an example of a FRAM visualisation.

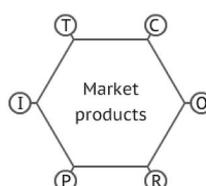


Figure 1 - Example of a single FRAM function and its six aspects responsible for marketing

It is possible to define functions at different levels of detail. The level of detail and the number of functions depend on the purpose for which the FRAM is being used. <Market products>, for example, could be a function, like in Figure 1, but it is also possible to go into more detail, like in Figure 3 at the end of this section. The first step in a FRAM analysis should therefore always be identifying the functions that are of importance or interest (e.g. Market products in Figure 1). It is, of course, possible and sometimes advantageous to concentrate on high-level functions at first and go into more detail in later stages of the analysis. It is not important which function is identified first. In a FRAM analysis there are not always clear start and end functions. Furthermore, the aspects of the functions ensure that all necessary functions are identified, regardless of the first function that is being identified.

Once an initial set of functions has been identified, the functions need to be described in more detail by defining (some of) their aspects:

- **Input:** is used or transformed by the function to produce the Output. Input can be anything like material, information, etc. The Input starts a function.
- **Output:** the result of what the function does. The Output can be anything like material, information, etc. When the Output has been produced, the function is completed.
- **Precondition:** has to be true or verified in order for a function to start. It does not in itself, however, constitute a signal to start a function. The Input starts a function. This distinction can be used to decide whether something should be an Input or a Precondition.
- **Control:** regulates or supervises a function so that the desired (or planned) Output is produced. Control can be a plan, a set of guidelines or rules, a schedule, etc. Control can also be social expectations, e.g. those by management or supervisors.
- **Resource:** is consumed when the function is executed. A Resource can be anything like matter, information, a machine, a software tool, etc. When something is a Resource, less is available after the function as was at the beginning.
- **Time:** captures the different ways in which time can affect a function. Time can be considered another form of Control. For example, a function may need to be carried out before, after, or in parallel to another function. Time can also relate to a single function that needs to start at a certain point in time. The Time aspect is generally of less importance for the purposes of this handbook

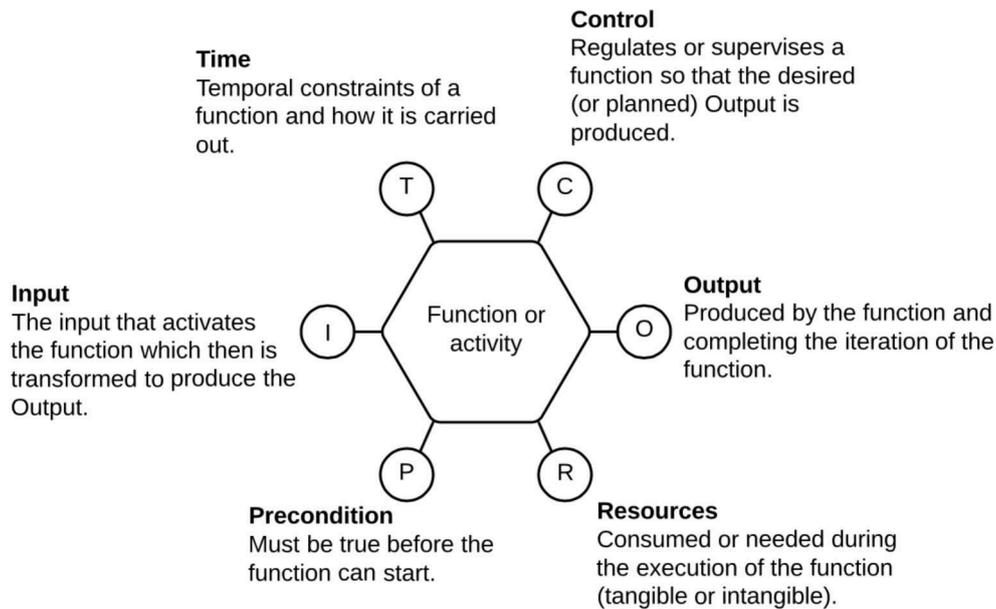


Figure 2 - The FRAM hexagon showing the six aspects and their descriptions

Functions need to have at least an Input or Output. Only the Output of a function can be connected to other aspects of other functions, i.e. connecting Precondition to Control is not allowed. It is often useful not to describe all aspects of a function at first, as this can make the analysis complicated and it is easy to lose sight of the bigger picture. It is recommended to describe only those aspects that are deemed appropriate for the analysis and for which information is available. For all other cases *background functions* can be defined. Background functions only have an Input *or* Output and are assumed to be stable during the execution of the functions (background functions are grey in FRAM). Figure 2 summarises the six aspects.

The FRAM example from Figure 1 has been extended to show <Market products> in more detail in Figure 3. To connect the first three functions only their Input and Output aspects have been described. The description of the aspects is shown as text boxes on the lines between the hexagons. <Provide advertising budget> is different from the rest as its Output is a Resource for <Develop under 20s commercial> (it is a Resource because Budget is being consumed during the execution of the function). Therefore, <Develop under 20s commercial> can only start (and be continued) when a budget is available.

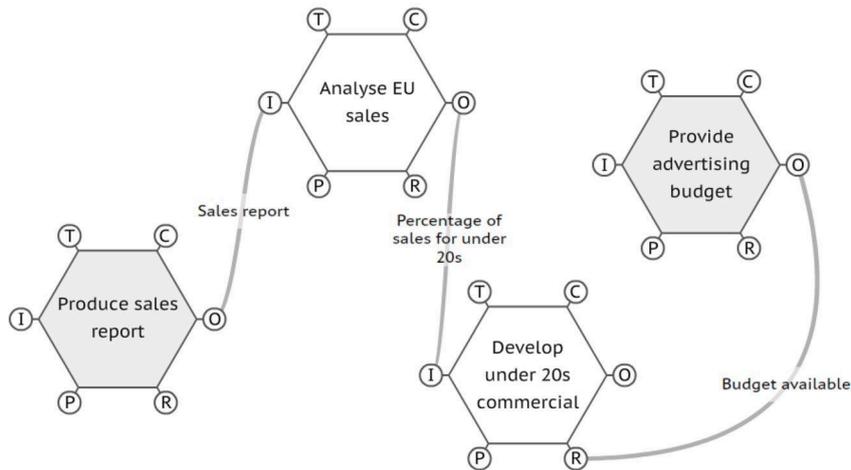


Figure 3 - FRAM example of Marketing functions in more detail. Background functions are shown in grey.

### cFRAM: A generic customer lifecycle to assist the data collection stages

A generic customer lifecycle has been developed that may assist software vendors in collecting data to identify functions and describe their aspects (see Figure 4). It should be considered as a model to guide the analysis and not one that prescribes it. Software vendors might find that the generic customer lifecycle does not apply to them, depending on their business model and the kind of products or services they distribute. Therefore, and to keep an open mind for the subsequent analysis, only high-level functions and aspects have been defined in the lifecycle. In the following, the generic customer lifecycle will be explained by using the FRAM notion.

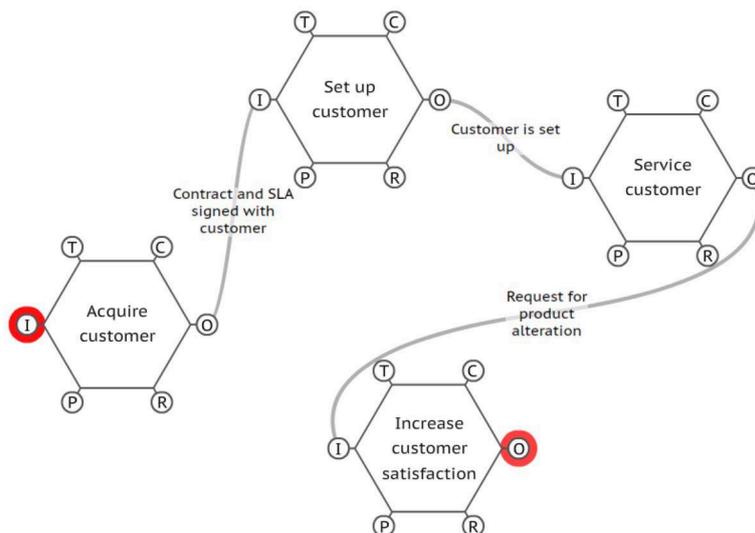
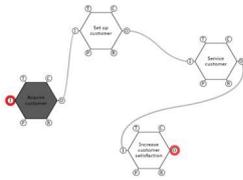
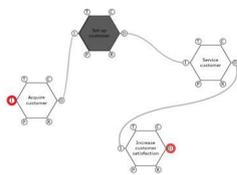


Figure 4 – Generic Customer lifecycle. The I (for Input) and O (for Output) in <Acquire customer> and <Increase customer satisfaction> respectively are circled red because they do not lead to a function.

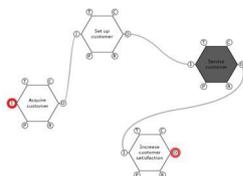
The customer lifecycle has four functions: (1) <Acquire customer>, (2) <Set up customer>, (3) <Service customer>, and (4) <Increase customer satisfaction>.



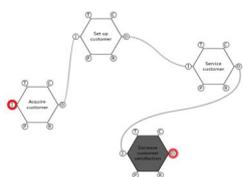
<Acquire customer> contains everything that is necessary to acquire new customers. This can, for example, include elements of Sales & Marketing such as advertisement, or calling potential customers. The function Acquire customer has two Inputs that can start the function: (1) [Project bid] where the software vendor bids on a publicly advertised project of a potential customer, (2) [Customer request] where a customer contacts the software vendor directly, e.g. because they came across an ad or searched for the product on the Internet. Within the function Acquire customer, a potential customer is converted to a customer. The Output of the function is, therefore [Contract and SLA (Service Level Agreement) signed] between the customer and the software vendor.



<Set up customer> contains everything necessary to enable customers to use the product or service the software vendor provides. This can include, for example, installing the product in the customers data centre, train the customer's employees, etc. The Output of the function is [Customer is set up] at which point they are able to use the product or service distributed by the software vendor.



<Service customer> deals with everyday problems customers might have. In other words, it contains everything necessary to support the users of the software vendor's product. This can mean responding to technical questions users might have, e.g. if something is not working, or assisting the customer in updating the products, or capturing feature requests. The Output of this function is [Request for product alteration], because only the development of new product features is not covered by <Service customer>. That is the purpose of the last function in the lifecycle.



<Increase customer satisfaction> contains everything necessary to retain customers as long as possible. In other words, this function deals with long-term customer issues, such as developing product features, whereas the previous function deals with short-term user issues, such as bugs or technical difficulties. The Output is <Enhanced product functionality>. With this function, the lifecycle comes back to the start as a new customer has been acquired and their long-term satisfaction is ensured. It could be possible to connect <Increase customer satisfaction> to <Acquire customer>, as the Output [Enhanced product functionality] can function as a Control aspect for <Acquire customer> (because new features, for example, can make it easier to expand the customer base).

During a cFRAM analysis, a graphical presentation of the cFRAM can be produced by using the software tool FRAM Model Visualiser<sup>1</sup>. A table presentation for one function is shown in Table 1. They can be useful to provide additional information, e.g. description of the function (the FRAM Model Visualiser provides the option to produce a report that contains the graphical presentation and tables). Depending on the number and couplings of the functions the visualisation of the FRAM can look complicated and referring to the tables might be more practical.

**Table 1 - Table presentation of the second function of the customer lifecycle FRAM**

<b>Name of function</b>	<b>Set up customer</b>
Description	Containing all actions and tasks to enable the customer to use the software vendor's products and services.
<b>Aspect</b>	<b>Description of aspect</b>
Input	Contract and SLA signed with customer
Output	Customer is set up
Precondition	
Resources	
Control	
Time	

## **cFRAM: Collecting data (Step 1)**

The data collection for a cFRAM analysis is very flexible. The guidance and processes described in this section are only a suggestion of what has worked well in practice and what the original handbook on the FRAM suggests<sup>2</sup>.

Within SME software vendors everyone is somehow part of product-development, -distribution or -support. For that purpose, a group discussion is suggested to start the data collection. In a group discussion people from various departments can take part and offer their views. Different views are necessary to identify the complementary organisational changes cloud computing requires, e.g. in the areas of HR, Sales & Marketing or Finances. It is recommended that the form of a group discussion be used for all stages of the cFRAM (the final page in this handbook should be handed out to all participants so that they can follow the steps of the analysis).

The following list is only a suggestion of departments that could offer valuable input to the data collection.

- Sales & Marketing
- Software development

<sup>1</sup> The FRAM Model Visualiser can be downloaded here: <http://functionalresonance.com/tools-visualisation/fram-visualisation.html>

<sup>2</sup> The handbook of the original FRAM can be found here: <http://functionalresonance.com/how-to-build-a-fram-model/fram-handbook.html>

- Technical infrastructure
- Support
- Other roles or departments depending on the kind of software product or services being sold

There should be a moderator to guide the discussion and to ask opening questions. The job of the moderator is also to encourage everyone to speak openly, not only about successes but also failures. The moderator should ideally be someone in a high-level position who has a good overview of the different departments and product areas. It could be, for example, the Managing Director or the leader of product development.

The following high-level questions are suggested to start a discussion to identify the functions for the FRAM that represents the on-site version of the software product:

1. How do we acquire new customers?
2. How are new customers set up to use our products?
3. How are users supported in their everyday use of our products?
4. How do we achieve long-term customer satisfaction?

*These questions are not exhaustive and the reader should think about other questions that might be appropriate for their particular circumstances or products. Appendix A proposes more questions that can guide the collection.*

While the proposed questions are being discussed, it is important to focus on the identification of functions that are necessary for everyday activities, and order all other information around these functions. It should be agreed on: the name of a function (should be a verb or verb phrase), the description of the function, and the description of aspects, i.e. Input, Output, etc.

### **Example: Collecting data**

The SME software vendor used for the examples, in the following referred to as Project Partner or PP, has one software product that was subject to the migration into the cloud. The product was primarily designed for the Oil & Gas industry. The product was migrated into the cloud to enable customers to use the product more quickly. Before the cloud, the PP had to have extensive discussions with the customer's IT department for configuring the hardware. Furthermore, the PP hopes to develop new competitive advantages by being in the cloud and expand into new international markets.

The steps proposed in this section were applied to the PP. The functions from the customer lifecycle of the previous section were explained to the Managing Director of the PP and their applicability discussed by asking the proposed high-level questions (a group discussion was not done since existing data collected from this software vendor over a 1 ½ year period was also used in which various departments were represented). Figure 5 shows the resulting FRAM. In the following each function of the FRAM will be explained in detail.

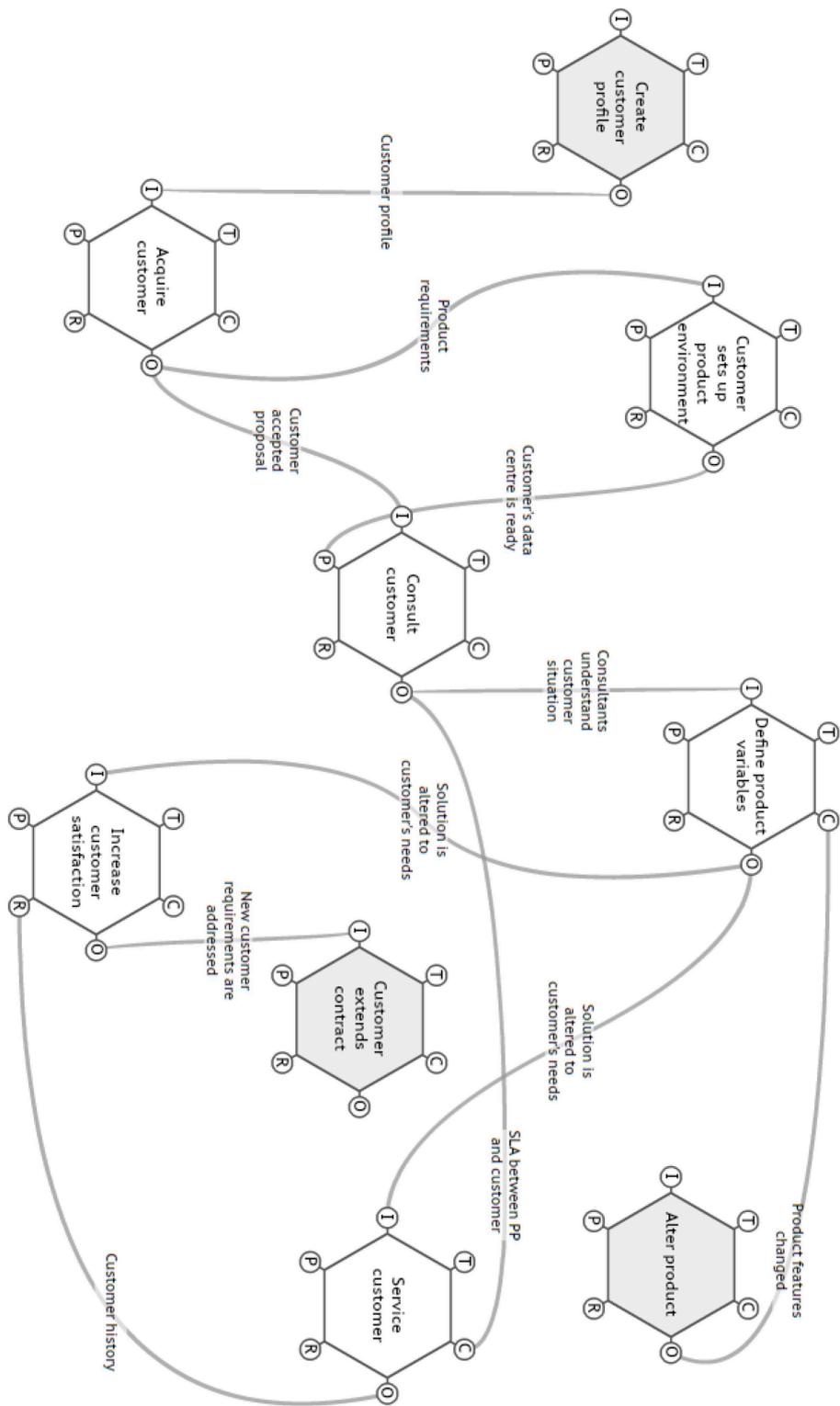
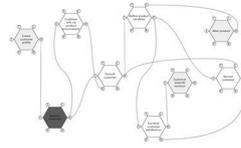


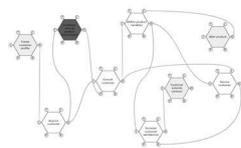
Figure 5 – FRAM of the PP before the move to the cloud. The text boxes on the lines are the aspects



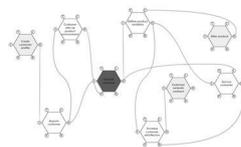
<Create customer profile> is a background function starts the FRAM. The PP is working together with another company that creates profiles of potential customers that the PP uses when contacting them. [Customer profile] is therefore the Input for <Acquire customer>.



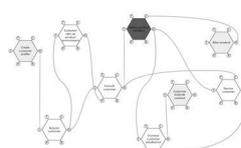
<Acquire customer> converts potential customers into actual customers. At the end of this function, the customer has accepted the proposal from the PP and receives the product requirements. The product requirements describe what kind of hardware and access to databases the PP needs from the customer so that the product can be installed in the customer's data centre.



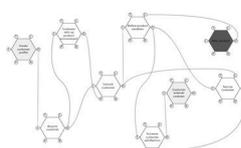
<Customer sets up product environment> represents the tasks related to setting up the hardware and access to databases. This function is outside of the PP's control. The PP can provide guidance to the customer but the customer retains the final responsibility. Only when this function is completed and the Output [Customer's data centre is ready] has been produced can <Consult customer> start. Therefore, the Output of <Customer sets up product environment> is a Control aspect of <Consult customer> (since the data centre needs to remain ready during the execution of <Consult customer>).



<Consult customer> is responsible for tailoring the initial installation of the PP's product to the specific needs of the customer, e.g. reflecting the customer's business processes. The function produces two Outputs that are necessary towards enabling the customer to start using the product. The first Output is [Consultants understand customer situation] that is the Input for <Define product variables>. The second Output is [SLA between PP and customer].

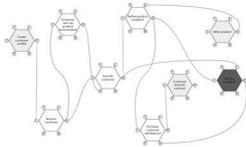


<Define product variables> uses the information collected in <Consult customer> to tailor the product to the customer's needs. The generic solution that is installed initially contains variables, such as the number of items in a report, which have been defined very broadly. The consultants can define the range of these variables more strictly to represent the customer's situation accurately. Therefore, the Output is [Solution is altered to customer's needs]. At this point, the customer can start using the product, which is shown by the fact that the Output is the Input for <Service customer> and <Increase customer satisfaction>. If, however, the product is not able to capture the customer's situation appropriately <Alter product> becomes important.

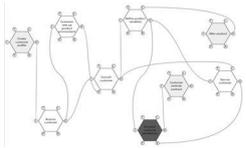


<Alter product> is a background function that captures the event if a software developer from the PP has to alter the

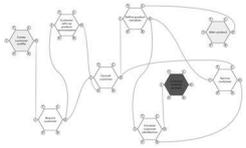
backend of the solution to satisfy customer requirements. Because this function is not relevant to every customer, its Output [Product features changed] is a Control aspect of <Define product variables>.



<Service customer> is carried out by PP's support staff and deals with everyday problems users might encounter, e.g. a report is not produced as expected. Customers have a phone number and FAQ sheets they can consult in these cases.



<Increase customer satisfaction> is handled by the sales department. The sales people who carry out this function are, however, different from those that carry out <Acquire customer>. The function's goal is to retain customers by convincing them to buy upgrades or new products. To achieve the goal, the function uses [Customer history] that is the Output of <Service customer> to know what kinds of issues the customer's users struggle with, for example.



<Customer extends contract> is a background function and end the FRAM. <Increase customer satisfaction> produces the Output [New customer requirements addressed] which is the Input for <Customer extends contract> and is used as a means to show that customers will extend the contract if their requirements are addressed.

## cFRAM: Identifying performance variabilities (Step 2)

After creating an initial version of the FRAM to capture the functions that show how software vendors operate for the on-site version of their product, this section will go into more detail to analyse the resilience of a software vendor through the identification of performance variabilities. The identified performance variabilities will be important for the fourth step (Plan the move to the cloud). The aim is to dampen the identified performance variabilities through the move to the cloud to increase the overall resilience of the software vendor.

The way in which the analysis of resilience is integrated into the FRAM is partly explained by its name. The method focuses on the analysis of functional resonance (hence the name Functional Resonance Analysis Method). Failures in today's systems emerge because the performance of functions vary (due to technological, human or organisational elements) and sometimes the variabilities reinforce each other causing the variability of one function to be higher than expected and affecting coupled functions (negatively or positively). To understand how the potential performance variability of functions can be identified it is useful to distinguish between three types of performance variabilities.

First, the function itself can experience performance variabilities, so called *internal performance variability*. In this case a function can fail, e.g. due to

organisational pressures that affect human performance or because equipment has not been maintained properly (wear and tear). Second, the function can fail due to a change in the working environment, so called *external performance variability*. In this case, a function fails because the function operates outside its designed parameters, e.g. extreme weather conditions. Third, the output from other functions can affect *downstream functions*. Downstream functions are those functions that use the Output of other functions as Input, Precondition, etc. If an upstream function varies in performance (or fails) and the Output is not as it should be (or not available at all) it can affect downstream functions. For example, if the Output of an upstream function is the Control for a downstream function, the downstream function will not start as not all conditions are fulfilled (like in the example in Figure 3 below). Table 2 provides a list of performance variabilities IT companies commonly experience. They can be used to start the discussion on performance variabilities.

The Output of functions can be affected by the internal, external, and downstream performance variability in terms of time and precision. Performance variability can cause functions to produce Output (1) *too early*, (2) *on time*, (3) *too late*, or (4) *not at all*. With regard to precision, performance variability can cause functions to produce Output that is (1) *precise*, (2) *acceptable*, or (3) *imprecise*.

The FRAM example from the second section of this handbook in which the basics of the FRAM were explained has been analysed for performance variabilities (see Figure 6). The function *Develop under 20s commercial* is likely to experience performance variabilities (shown by the wave symbol in the hexagon) due to the Resource *Budget available* from *Provide advertising budget*. If the budget is not sufficient, the function *Develop under 20s commercial* is likely to produce an Output that is imprecise. Thus, the performance of an upstream function affects the performance of a downstream function.

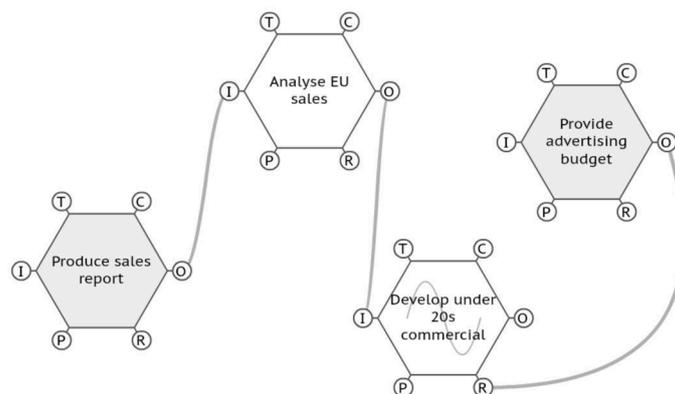


Figure 6 - Identification of performance variability (shown by a wave symbol in the function)

**Table 2 - Common Conditions to help with the identification of performance variabilities**

CC/name	Explanation
CC1: No user support available	Users have problems using the provided IT services but IT support is not available, e.g. because its the weekend or night or IT support is tailored to UK times.
CC2: Request from user to IT department or software developer is opaque	Users request a new product feature or IT service but do not express the requirements clearly enough, which can lead to the development of product features users were not requesting. That requires the developers to invest more time as might be necessary to fix the developed features.
CC3: Set up of user software/hardware takes too long due to insufficient manpower	Users cannot start working as the IT department or software vendor has insufficient manpower to install the requested software/hardware. This CC is often closely related to CC4.
CC4: Request for software/hardware reaches IT department or software vendor too late	The IT department or software vendor is put under pressure because users request software/hardware with too short notice and expect the IT department or software vendor to react instantly.
CC5: User has problems using provided IT services	Users might put the wrong types of data into the provided IT services causing the IT service to fail, or they do not understand how to use an IT service which results in a support call.
CC6: No capacity to develop requested product features/IT services	Users request new IT services or product features but the IT department or software vendor has insufficient manpower or monetary resources to develop the requested IT service or product feature. This can result in users taking their own actions or moving to a competitor.

**Example: Identifying performance variabilities**

Together with the Managing Director it was discussed in which functions they either see a potential performance variability or have experienced performance variabilities in the past (in terms of Time and Precision). Figure 7 shows which functions of the PP experience performance variabilities. They are elaborated in the following.

The diagrams consist of three identical hexagonal shapes arranged vertically. Each hexagon has a central text label and six vertices labeled with letters: T (top), C (top-right), O (right), R (bottom-right), P (bottom-left), and I (left). A wavy line is drawn across each hexagon, passing through the central text.

- Acquire customer:** Customers sometimes have trouble signing off on buying the software as the sales price is high which requires people higher in the hierarchy of the customer to sign off on the investment.
- Customer sets up product environment:** The customer often fails to accomplish this task on time as the customer’s IT department needs a lot of time to buy new hardware and install it.
- Service customer:** The customer sometimes fails to install updates at all or properly. Customers try to save time by installing updates directly to their live environment, without testing them first.

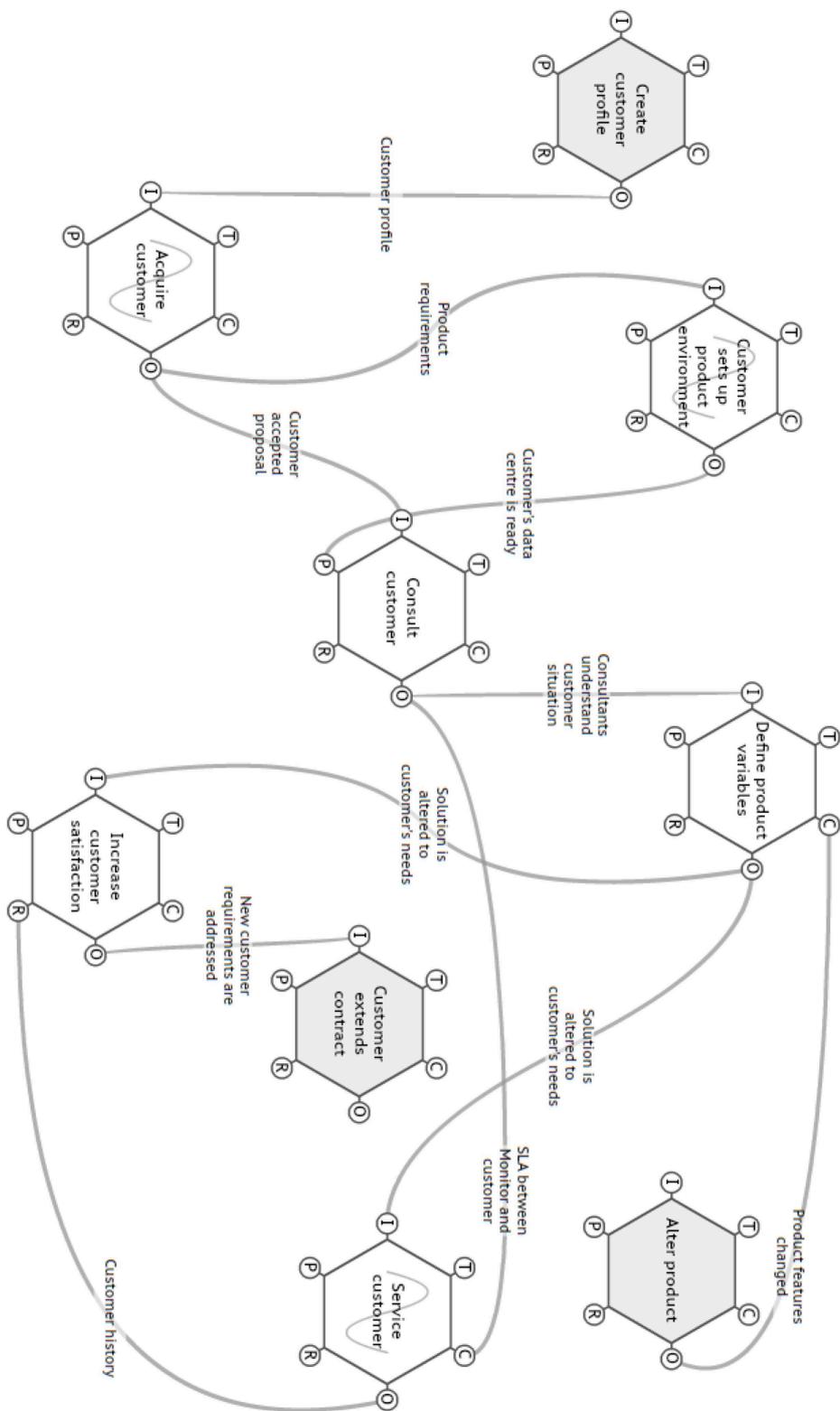


Figure 7 - Continuation of the example in which all necessary aspects and background functions (in grey) have been identified. Performance variabilities are shown by a wave symbol.

### cFRAM: Identifying existing capabilities (Step 3)

Existing capabilities of a software vendor are identified with the help of the FRAM created in the previous two sections and four viewpoints that are introduced in this section. The four viewpoints (cultural complexity, management complexity, application complexity, and governance complexity) will assist companies in identifying the underlying enablers or inhibitors of the functions in their FRAM. Table 3 provides a short overview of the sub-steps carried out in step 3 of the cFRAM.

Table 3 - Overview of the sub-steps to identify existing capabilities

Sub-step	Purpose
1. Identify the three most important functions	cFRAM is interested in core capabilities from which the company derives competitive advantages. Core capabilities reside within several functions.
2. Define the resources of the functions and assign them to one of the four viewpoints	Define the purpose of the three most important functions and list their resources. Then investigate how the resources are influenced or constrained by assigning them to one of the four viewpoints.
3. Identify capability (or capabilities)	Identify the influences and constraints across resources of the three most important functions to reveal the capabilities that are going to be affected by cloud computing.
4. Extrapolate capabilities to other functions (or identify additional capabilities)	Investigate if identified capabilities are appropriate for other functions in the FRAM model. Otherwise, repeat the above steps.

The three most important functions of the FRAM model need to be identified in order to start the identification of capabilities. The reason for focusing on three functions is that a cFRAM analysis is about adapting core capabilities and not all capabilities that might exist. Furthermore, it is important to investigate the impact of cloud computing on those capabilities from which the software vendor derives competitive advantages. Trying to identify the capabilities of all functions can become complicated depending on the number of functions in a FRAM model. The capabilities of other functions, i.e. those not part of the three most important ones, can be identified in later stages by repeating the sub-steps explained in this section. When identifying the three most important functions, background functions should be excluded. They should be excluded because they generally do not have a large impact on the FRAM model as a whole and thus are unlikely to contain core capabilities.

The resources can either come from other functions that are connected through the Resource aspect, or they can be internal resources such as people, documents, machines, etc. After listing the resources of a function, they need to be assigned to one of the four viewpoints. The viewpoints indicate how the use of resources is influenced and in which cases their use might be constrained.

The reason for doing this can be explained with the help of the definition of capabilities: capabilities use *different resources*, tangible and intangible, in a specific way to carry out a task. By categorising the resources with the viewpoints, software vendors also get a deeper understanding of how the use of one resource might influence or constrain the use of another resource. The way software vendors exploit the influences and minimise the constraints will reveal their capabilities.

The four viewpoints are *cultural complexity*, *management complexity*, *governance complexity*, and *application complexity* (they have been identified through a 12 month multi-stage with five software vendors that were followed during their migration to the cloud):

- Cultural complexity: Often plays a role for resources where people are involved. People can be the resource (or part of the resource) or people are affected by the use of a resource.
- Management complexity: Often plays a role when the use of software needs to be coordinated with other resources, e.g. people or business processes.
- Application complexity: Is likely to be at the centre of many resource constraints. It is likely to be found in connection with resources of the products the software vendor develops or the use of them.
- Governance complexity: Often plays a role where documents are involved, which describe how a task needs to be carried out. Difficult to deal with, as it cannot be easily changed and some of it might be out of the control of the software vendor or the customer. That is why Governance complexity, in contrast to the other viewpoints, often only constrains what can and cannot be done.

**Table 4 – Examples of resources assigned to the viewpoints**

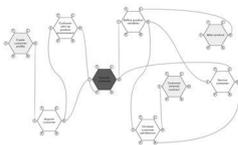
<b>Viewpoint</b>	<b>Examples of resources</b>
Cultural complexity	Internalised rules and norms of behaviour that employees follow, e.g. Sales people
	Preferences or behaviours of customers and users
	Knowledge base to inform about past customer interactions and preferences
	Market situation and behaviour of competitors
Management complexity	Road map for product development, e.g. documents ensuring product features are feasible and developers have appropriate skills
	Coordinating feature development and the lifecycle of technologies generally
	Supporting the customer in using the product or increasing customer satisfaction, e.g. Ticket system for support
	Software development environments, e.g. Eclipse or Visual Studio
	Communication tools, e.g. Email

Application complexity	Development of product features and the approaches used, e.g. SCRUM
	Dependencies that can enhance or stifle the development of a product, e.g. APIs or libraries
	Programming languages that are being used, e.g. C# or Java
	Set up of the product at the customer site, e.g. installation procedures
Governance complexity	Formal rules and norms of behaviour, e.g. for interacting with customers
	Governmental or institutional laws and regulations
	Corporate policies, e.g. documents to ensure corporate identity is achieved
	Handbooks or product documentations that specify what the software vendor's product can and cannot do
	FAQs created for customer support

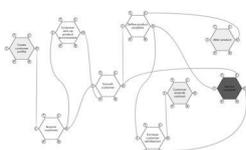
### Example: Identifying existing capabilities

The three most important functions of the FRAM shown in Figure 7 (see page 12) are (Sub-step 1):

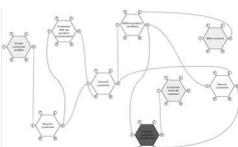
1. Consult customer
2. Service customer
3. Increase customer satisfaction



The function *Consult customer* is the most important function as the PP makes a large share of their revenue from consulting services that go together with their sale of software products. When customers buy the PP's product, it comes in a very generic form and needs to be tailored to the customers' needs. That process is carried out by consultants from the PP that go to the customer and examine their business processes and other information.



*Service customer* is the second most important function as it is responsible for supporting the users of the customer during everyday activities. In other words, this function is responsible for ensuring that customers can use the product in the way they need to. If customers are unhappy with the support, bugs are not being fixed fast enough, or requested features are not introduced, the customer might acquire a different product.



*Increase customer satisfaction* is the third most important function as it is responsible for ensuring long-term customer satisfaction. This function works in tandem with *Service customer*. Whereas *Service customer* is more about short-term satisfaction of users, *Increase customer satisfaction* is responsible for renewals of licenses and the sale of new

products or upgrades, for example. Thus, this function tries to ensure on-going revenue from customers.

From the description of the three most important functions, it becomes clear that all of them work towards the same overall goal. The overall goal is enabling the users of a customer to use the software product in the way they require it on a daily basis.

To further understand the overall goal, it is worth listing the resources these functions require and how these are influenced or constrained by the four viewpoints (Sub-step 2). Table 5 shows a listing of the functions, their resources, and categorisation into viewpoints.

**Table 5 - Showing the functions with their resources and viewpoints**

<b>Function</b>	<b>Resource</b>	<b>Viewpoint</b>	<b>Explanation</b>
Consult customer	Consultants	Cultural complexity	Consultants are confronted with the culture of the customer, which influences their proposed solutions. Furthermore, they are also influenced by the culture of the PP.
	The software product	Application complexity	The product is influenced by application complexity as it determines what the product can and cannot do and thus what the consultants can offer the customer. It might also not be technically feasible or desirable to develop every feature a customer requests.
	Procedure documentation	Governance complexity	Procedure documentation is influenced by governance complexity as the documents inform about corporate policies and formal rules of behaviour that must be followed.
	Product documentation	Governance complexity	See "Procedure documentation"
Service customer	Support personnel	Cultural complexity	See "Consultants"
	Help desk software	Management complexity	Help desk software is influenced by management complexity, as it depends on how the help desk software was chosen, how it is integrated into the company, and how it is being used.
	Software developer	Application complexity	Software developers are mainly influenced by application complexity as it depends on how the software has been developed in the past, how it utilises other software products, e.g. through APIs, and what skills the developers have.
	The software product	Application complexity	See above
	Software	Application	See "The software product"

	product updates	complexity	
Increase customer satisfaction	Sales people	Cultural complexity	See “Consultants” and “Support personnel”
	Software developer	Application complexity	See above
	The software product	Application complexity	See above
	Customer history	Management complexity	Customer history captures issues that have been subject of customer interactions. If it is appropriately managed it can be used to inform the development of product features or new products.

All four complexities are represented in Table 5. Application complexity (6 resources) and cultural complexity (3 resources), however, appear more often than Management complexity (2 resources) and Governance complexity (2 resources). Indeed, the if considering the overall goal the three functions appear to achieve, the number of resources for each viewpoint seems plausible. Application complexity is by far the most represented viewpoint as PP1 aims to enable customers to use their product. Hence, application influences what can be offered to the customer in terms of functionality and how the customer can use the product. Cultural complexity is the second most represented viewpoint as PP1 might have to adapt to the customer’s needs and behaviour. That will influence the actions and behaviour of PP1. Management complexity is less often represented but still of importance as the three functions need to be managed and coordinated (since they are all working towards the same goal and <Service customer> and <Increase customer satisfaction> can only start if <Consult customer> has produced its Output. Governance complexity, as the final complexity, is also of importance as a control instrument since the functions (and employees of PP1) need to be aware of they can offer their customers. In other words, the offers need to be in line with what the software product of PP1 can actually do.

With the categorisation of the resources into viewpoints it is possible to identify a core capability the PP possesses: service delivery management (Sub-step 3). The PP has to deal with three overall issues that make up the core capability. First, all three functions’ objective is providing the product the customer needs. Initially one might therefore call the capability product delivery management. The description of the functions and their resources show, however, that the PP provides more than the software product, i.e. consulting services and support—therefore *service* delivery management capability. Second, all three functions are organised around delivering and enabling the customer to use the product. Furthermore, the functions customise the software product for the customer and continually update it—therefore *service delivery* management capability. Third, all three functions have to be managed as they need to be coordinated and procedures need to be followed. Furthermore, the continued communication with the customer needs to be managed—therefore *service delivery management*.

The identified core capability has some applicability to other functions of the FRAM, e.g. <Define product variables> (Sub-step 4). The remaining two functions are not appropriately covered by the capability. <Customer sets up product environment> contains, in fact, no capability as this function is carried out by the customer and the PP has only a very limited amount of control over it. <Acquire customer> is a sales function and some of it is outsourced to a third party that creates potential customer profiles (see description of function above). As it is the only function for sales it is more appropriate to keep the function instead of converting it into a capability.

## cFRAM: Planning the move to the cloud (Step 4)

The previous section investigated the functions and their resources to identify existing capabilities. At this point, the software vendor should have a good understanding about what they can and cannot do. In this section, this knowledge is used to plan the move to the cloud. It is investigated how the functions in the FRAM need to adapt to accommodate cloud computing. Furthermore, it will be concluded what steps need to be taken to dampen performance variabilities, if the existing capabilities are likely to enhance or stifle the changes to the functions and, if necessary, what new capabilities need to be developed. Table 6 provides an overview of the sub-steps carried out in this section.

Table 6 - Overview of the steps for planning the move to the cloud

Sub-step	Purpose
1. Adapt functions to the cloud	Go through every function of the FRAM model and investigate the impact of cloud computing by discussing changes in responsibilities. If necessary, introduce new functions to respond.
2. Investigate impact of adaptations on performance variabilities	Conclude if the performance variabilities can be dampened through the adaptations of sub-step 1 or if further adaptations are necessary.
3. Investigate impact of adaptations on existing capabilities and inform development of new capabilities	List resources of functions in the cloud and assign to one of the four viewpoints. Afterwards compare the list with the list of step 3 of cFRAM to inform the adaptation of capabilities.

To adapt functions to the cloud and create the *after* cloud migration FRAM model, software vendors need to investigate how functions in the *before* cloud migration FRAM model need to adapt to accommodate cloud computing (sub-step 1). When investigating if functions have to change, can change or do not need to change, the resources of functions can assist. If, for example, a resource becomes unavailable in the cloud or resources are added, the related function(s) have to change. If resources are not affected, but software vendors see a potential to exploit existing resources differently, the function could

change (now or at a later stage) to enhance the capabilities in the cloud. Otherwise, there is no immediate need to change a function for moving to the cloud.

When the decision is made to introduce new functions or to adapt existing ones, companies need to answer two questions in the process. First, how do we want to carry out this function from a technical perspective? Second, how do we want to carry out this function from an organisation perspective? It is important to discuss both questions together as decisions made for one questions can constrain the options available for the other. In addition, it needs to be ensured that the technical and organisational aspects do not inhibit each other.

The software vendors should also investigate how the adaptations will affect the performance variabilities that were identified in the *before* cloud migration FRAM (sub-step 2). Cloud computing can be used as an opportunity to eliminate or dampen performance variabilities. The options software vendors have in doing so depend on the nature of the performance variability. Further adaptations might be necessary to eliminate or dampen performance variabilities or to avoid new performance variabilities. When carrying out sub-step 2 it is important to consider that changing a function to dampen performance variabilities can have an impact on coupled functions. Thus, changes to functions can ripple through the FRAM model. When adapting functions, whether it is necessary or deemed appropriate to dampen performance variabilities, software vendors need to investigate the wider consequences. Otherwise they risk decreasing their resilience, instead of increasing it.

When all functions have been adapted to the cloud software vendors need to investigate how the adaptations will enhance or stifle existing capabilities (sub-step 3). Software vendors should ask themselves if the capability is able to work without the functions and resources that are being eliminated through a move to the cloud or if the capability is able to integrate any new functions and resources. Furthermore, software vendors are encouraged to ask themselves how capabilities need to be adapted in order to dampen performance variabilities further. To answer these questions, it is necessary to list the resources of the functions in the *after* cloud migration FRAM model and assign these to the four viewpoints in the same way as in step 3 of cFRAM. Software vendors can start with the three most important functions identified in step 3 of cFRAM. If one of the three most important functions becomes obsolete through the cloud it is sufficient to start the analysis of capabilities with the two remaining functions. By comparing the list with the list that was created in step 3 of cFRAM for the *before* cloud migration FRAM model software vendors can inform the adaptation of their capabilities (see below). Afterwards the step should be repeated for other functions that changed (or functions that are new) by investigating if the adapted capability is appropriate for them or if new capabilities need to be developed. It is, therefore, ensured that all functions in the cloud FRAM model are investigated to inform the adaptation of capabilities.

To guide the adaptation of capabilities, software vendors moving to the cloud should investigate in particular changes to responsibilities. Does the software

vendor outsource any responsibilities to the cloud provider? If yes, related resources are likely to be outsourced too, which can make existing functions and capabilities obsolete. Furthermore, does the software vendor take over responsibilities from the customer? If yes, new resources are likely to be introduced which may require the company to introduce new functions and develop new capabilities. Existing capabilities might need to be adapted for either of (or all) of the following three reasons:

1. Functions have been adapted that included the introduction of new resources or elimination of existing resources
2. New functions have been introduced or existing ones have been eliminated
3. The viewpoint of a resource changes, i.e. other factors now influence or constrain the use of a resource, e.g. from cultural complexity to management complexity

### **Example: Planning the move to the cloud**

Cloud computing changes the responsibilities of the PP. The PP takes over responsibilities previously held by the customer and, at the same time, passes on some responsibilities over the infrastructure to the cloud provider. Figure 8 shows what functions of the PP have changed. They are elaborated in detail below.

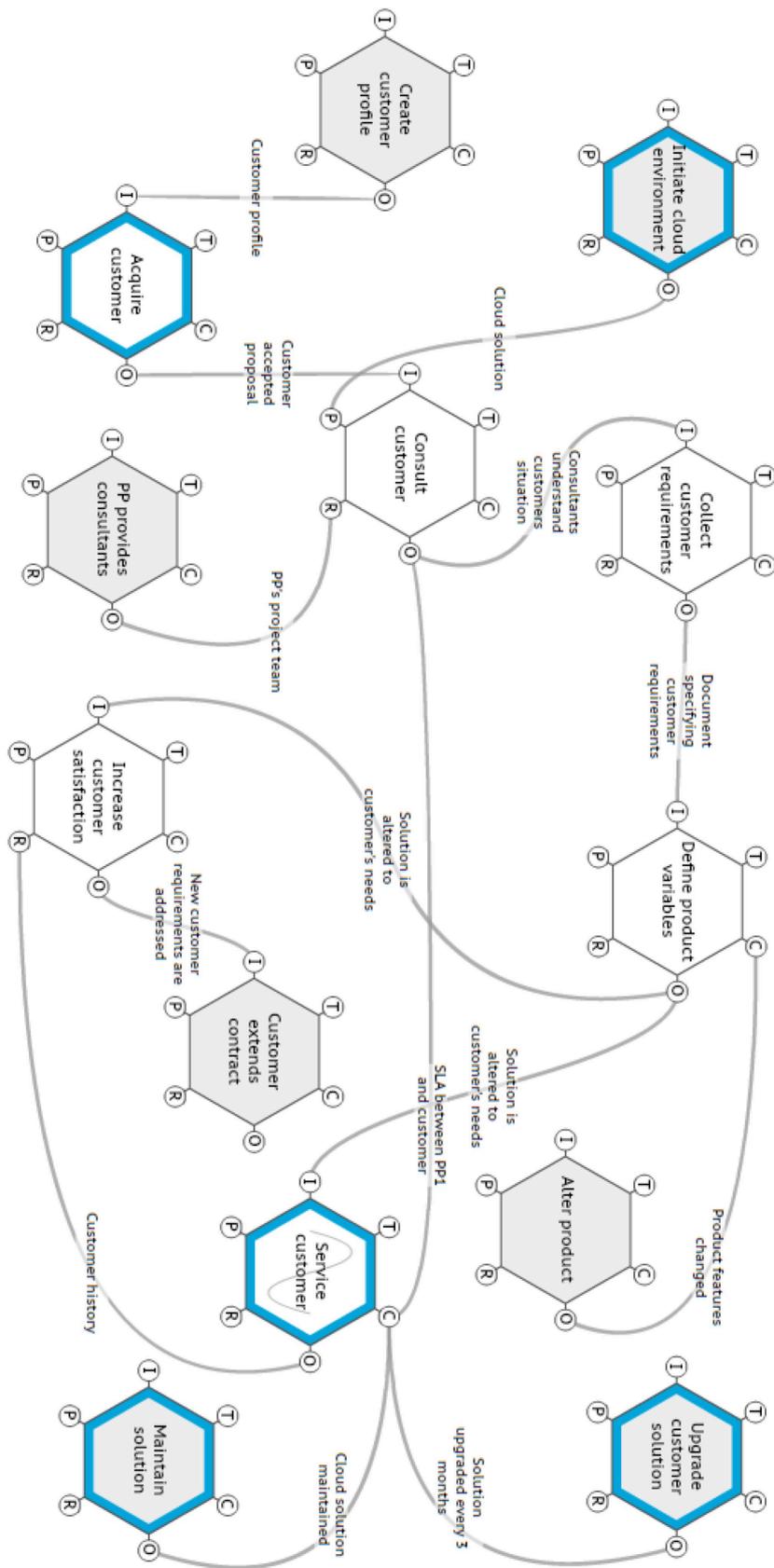
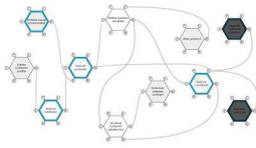


Figure 8 - The cloud FRAM. Changed functions have been highlighted by a blue frame around the function

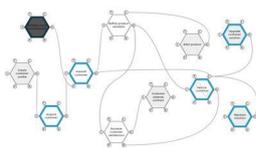
By investigating the effects of the changes in responsibilities on the three most important functions of the PP <Consult customer>, <Service customer>, and <Increase customer satisfaction> the following can be concluded (sub-step 1). <Consult customer> does not need to change. The function is mostly about analysing the customer and its business processes to know how the product needs to be tailored to the customer. For this function it is, in other words, irrelevant if the PP sells on-site or cloud-based products. The resources are the consultants, the software product, procedure documents and product documentation. The following resources do not change: consultants, procedure documents, and product documentation. The software product does change, but they are irrelevant for the Consult customer function (as only the back-end technology changes and the functionality of the product remains the same).



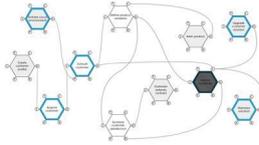
<Service customer> needs to change, as it will take over responsibilities from the customer. Next to supporting the customer in their everyday use of the product, in the cloud this function is also responsible for maintaining the product and installing updates. It is recommended to introduce two new functions. <Maintain solution> to ensure that the product is running as is expected by the customer. <Upgrade customer solution> to fix bugs and install new features.

<Increase customer satisfaction> can change but does not have to. The function does not have to change because its resources are not affected by the move to the cloud and stay the same. The PP might not want to change the function during the initial move to the cloud in order to have more resources for essential changes. Furthermore, the function did not experience any performance variabilities in the past so there is no immediate need to improve the function. If the PP should decide to change the function at a later stage, a new resource is available to the PP. As the PP has more possibilities of analysing how their product is being used (partly through the introduction of <Maintain solution>), the PP might use this knowledge to anticipate product features that can increase customer satisfaction. In other words, the new resource available is Product usage.

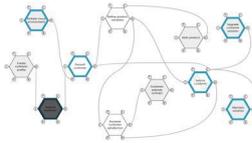
After developing this initial plan of the move to the cloud, the impact of this plan on the performance variabilities need to be assessed (Sub-step 3). Three functions had performance variabilities in the initial FRAM.



The performance variability in <Customer sets up product environment> existed because customers often needed too long to install necessary hardware in their data centres. Through a move to the cloud, this performance variability can be eliminated because the function became obsolete. The function is replaced with <Initiate cloud environment>. It is responsible for acquiring the cloud resources on which the customer's solution will be installed. Therefore, the PP is now in control of acquiring the cloud resources



The performance variability in <Service customer> cannot really be dampened. The reason for the performance variability was that customers sometimes did not install updates correctly and made the PP responsible for a faulty product. In the cloud, the PP is responsible for installing updates. They can ensure that these are installed properly and in a timely manner. However, they have to rely on the cloud provider for the infrastructure. If the cloud provider has an outage the PP's product is unavailable. Therefore, the performance variability in <Service customer> has essentially been replaced by a new one.



The performance variability in <Acquire customer> has not been addressed so far because it was concluded that the function can change through a move to the cloud but does not have to. It is suggested that the PP uses the move to the cloud to change their payment models. The reason for the performance variability was that the PP's product is high-value, thus, high-priced, which required people high in the hierarchy of the customer to approve the buy. In the cloud, it is possible to offer a subscription model, which would spread the initial price over a longer time (e.g. 12 months). By requiring customers to rent the product for a minimum of 12 months, for example, the PP can ensure that they still get the same amount of money as before the cloud. Hence, the performance variability in this function can also be dampened by a move to the cloud.

To understand the impact of the changes on the service delivery management capability, the PP needs to investigate if it is likely to enhance or stifle the changes (Sub-step 3). It was previously concluded that the only <Service customer> has to change for a move to the cloud. <Service customer> takes over responsibilities from the customer to maintain their solution and install updates, etc. The existing capability, service delivery management, however, is only responsible for customising the product to the customer's needs and supporting the customer during everyday activities. In other words, the capability is more about soft than technical skills. The two new functions that have been introduced are <Maintain solution> and <Upgrade customer solution> and require more technical than soft skills (see Table 7 for a list of changes to the resources). It is, therefore, questionable if the existing capability is suitable for the new functions. The PP should develop a new capability that is responsible for managing the cloud environment. In other words, it is responsible for acquiring and integrating cloud resources and other cloud-based services into the company. Subsequently, it is also responsible for releasing the resources and services when they are no longer needed. The capability also allows the PP to increase their resilience as it helps in dampening previously experienced performance variabilities, for example, in <Customer sets up product environment> or <Service customer>. The capability will be called *cloud service management* and will reside in <Maintain solution>, <Upgrade customer solution>, and <Initiate cloud environment>.

The service delivery management and cloud service management capability have a close relationship. The service delivery management capability needs to inform the cloud service management capability about the requirements for new cloud resources and services and the cloud service management capability is then responsible for acquiring these. Only if both capabilities work in sync can the software vendor provide the products and services customers need.

In the following, the impact of cloud computing on the remaining functions will be assessed. <Define product variables> does not need to change as, similar to <Consult customer>, it uses the PP's front-end of the product, which does not change through a move to the cloud.

**Table 7 - Summary of changes that have an impact on capabilities**

<b>Adapted functions</b>	<b>Changes to resources</b>	<b>Viewpoint</b>	<b>Explanation</b>
Service customer	Support personnel	Changes from Cultural complexity to <i>Management complexity</i>	Personnel has to take over responsibilities from customer to manage the product installation
	The software product	Application complexity	Back-end of the product changes which impacts the responsibilities the function has to fulfil
	Software product updates	Changes from Application complexity to <i>Management complexity</i>	Instead of the customer the PP is now responsible for installing updates
Maintain solution	Cloud environment	Management complexity	The PP is responsible for managing the computing resources on which the product is executed
	The software product	Management complexity	The PP is responsible for managing the installation of every customer and offer a reliable operation
	Procedure documents	Governance complexity	Documents should describe how the tasks of managing the cloud environment are to be carried out
Upgrade customer solution	Software product updates	Management complexity	The PP is responsible for updating every customer solution properly and in a timely manner
	Procedure documents	Governance complexity	Documents should describe when and how updates are installed for a customer so that every customer gets the same service

## **cFRAM: Using the cFRAM as a long-term planning tool**

The steps in this handbook describe a one-time application of the cFRAM before the migration of software products to the cloud. It is possible, however, to use elements of the cFRAM to continuously check throughout the migration if the organisational changes are proceeding as planned. Furthermore, it is possible to use the cFRAM in conjunction with another method, *constructive engagement*, to prepare employees and customers for the cloud.

### **Applying the cFRAM throughout the migration**

Applying the cFRAM throughout the migration allows software vendors to track their progress of the migration and react to or anticipate changes more quickly. The advantage of applying the cFRAM continuously during the migration is that it makes companies aware of systemic changes that cloud computing might require. In other words, if we change customer acquisition, does customer support need to change in a similar way to provide customers with a coherent experience?

To use the cFRAM throughout the migration two steps are recommended. First, software vendors will need to go through the questions proposed in the first two steps to collect the data to create a FRAM showing the latest functions and performance variabilities in the migration process. Some of the questions may need to be changed in order to accommodate organisational changes that have been carried out successfully. After having created the FRAM showing the functions and performance variabilities in the current stage of the migration, software vendors should compare the FRAM to the FRAM from step 4, i.e. the desired functions. Then the software vendor can track the progress of the functions they have already developed and those they still need to develop. The results from this comparison can inform the next steps a software vendor has to take in the migration, e.g. which function to adapt next.

How often the cFRAM should be applied throughout the migration and in what intervals depends on the software vendor and their products. Studies about the migration of software products into the cloud have shown that for some companies it takes a few months to move to the cloud, whereas for others it takes several years. Software vendors may want to apply some of the steps of the cFRAM before and after a major organisational change. By applying it before an organisational change, they can understand what functions and couplings they have at the moment, and to work out how they need to be changed to accomplish the organisational change. By applying it afterwards, they can use the cFRAM as a control instrument to check if the organisational changes have been carried out as planned. This will also make it easier to react to and anticipate future organisational changes.

### **Accommodating the viewpoints during the migration**

The four viewpoints, cultural complexity, management complexity, governance complexity, and application complexity, have been introduced to assist software vendors in categorising the resources of a function and identifying their dependencies. When categorising the resources with the viewpoints it can be that one viewpoint, e.g. cultural complexity, appears particularly dominant, e.g. if it appears in several functions several times. Software vendors might therefore feel the need to pay particular attention to cultural complexity.

Software vendors should not, however, concentrate too much on one viewpoint as all the viewpoints are linked together. In other words, if one viewpoint changes, these changes are likely to appear in other viewpoints too. For example, if the software vendor changes product development processes, as part of application complexity, these changes are likely to have some effect on cultural complexity too, as the daily work of the employees changes and they need to re-learn how to behave in some situations.

It is, however, also not possible to balance the four viewpoints equally because sometimes they might be in conflict with each other (e.g. cultural complexity might be in conflict with governance complexity). Instead, they have to be accommodated. Software vendors need to choose one viewpoint that is of particular importance to them and organise the other viewpoints around it. It is possible for the most important viewpoint to change over time. For the migration of software products into the cloud, for example, application complexity could be particularly important if the software product is expected to change a lot during the migration. If the software vendor is more concerned with the customer's opinion on cloud computing, it can be more appropriate to make cultural complexity the most important viewpoint. After the migration has been carried out successfully, the most important viewpoint might change to management complexity, in order to optimise business processes in the cloud in terms of efficiency.

## **Preparing employees and customers for the cloud**

Organisational changes required by cloud computing will not only affect business process but also everyday activities of employees and sometimes even customers. The examples used in this handbook have shown that for the majority of functions, people are, in one form or another, a resource of them. This means, employees and customers need to be prepared for the organisational changes that will be carried out during the migration. In order to prepare employees and customers and assist them in developing the appropriate skills for the cloud, the notion of constructive engagement is proposed.

Constructive engagement activities can be used to provide a means of integrating the work of software engineers and other employees into the organisational change process. There are three types of constructive engagement: (1) defining the problem; (2) constructing the solution; (3) evaluating the solution. By defining the problem everyone involved in the development of the cloud product gets the same understanding of why and how

cloud computing is used. This helps to align the development processes with the organisational objectives. Constructing the solution involves making the software engineers and other employees aware of specific cloud computing issues (e.g. that changes to the software product are available to everyone instantly after release). This includes reaching agreement about which methods are used and integrating them into everyday work routines. Evaluating the solution means understanding how the product is meeting customer expectations. In other words, when new requirements arise, or existing requirements change, or when problems arise with satisfying the original requirements, these need to be assessed in their own right, and in terms of the wider development project.

It can be useful to apply constructive engagement on different levels throughout the migration as not all employees and customers will be affected by cloud computing equally. For example, the work of software engineers is likely to be affected by cloud computing from the start, as the software products have to be changed to accommodate the cloud. In the next stage, other employees of the software vendor are likely to be affected as business processes, such as support, start to change to reflect the changes in the product. Only in the last stage, customers are likely to be affected, when the software vendor begins to sell the cloud-based product. Therefore, it is recommended to apply constructive engagement first to software engineers, then to the software vendor as a whole and only then to other stakeholders, such as customers.

## **Appendix A – Questions to stimulate discussions**

The following questions aim to go into more detail than the high-level questions proposed for the first step of the cFRAM (Collecting data).

Regarding question 1: How do we acquire new customers?

- i. How do we advertise our products? How do we find potential customers?
- ii. How do we contact potential customers?
- iii. How do we demonstrate our products?
- iv. How long does the process take until a potential customer becomes an actual customer?

Regarding question 2: How are new customers set up to use our products?

- i. Who is involved on our side?
- ii. Who is involved on the customer's side?
- iii. How long does it take to set up a new customer?

Regarding question 3: How are users supported in their everyday use of our products?

- i. Who in our company is responsible for user support?
- ii. How are bugs reported and fixed?
- iii. How are our product(s) updated? Are they updated on a regular basis?

Regarding question 4: How do we achieve long-term customer satisfaction?

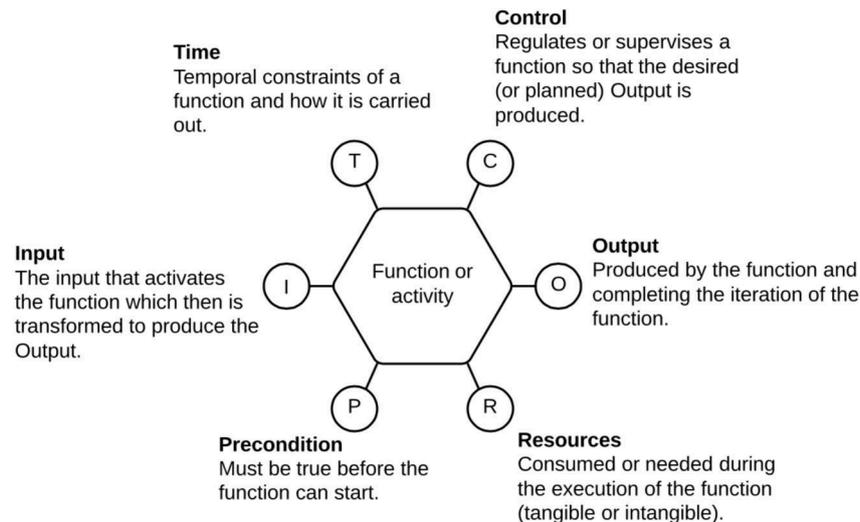
- i. How are new customer requirements implemented?
- ii. How do we develop new features/products?
- iii. Who in our company is responsible for long-term customer satisfaction?

## cFRAM notes

### Steps of the cFRAM

1. Create FRAM for current state of business
2. Identify performance variabilities
3. Identify capabilities
4. Adapt FRAM to cloud computing
5. Investigate impact of adaptations on performance variabilities and capabilities

### The FRAM:



### Performance variabilities

Failures in today's systems emerge because the performance of functions vary (due to technological, human or organisational elements) and sometimes the variabilities reinforce each other causing the variability of one function to be higher than expected and affecting coupled functions (negatively or positively).

### Identifying capabilities

1. Identify three most important functions
2. Identify resources of these functions (internal resources or through Resource Aspect)
3. Discuss how resources are exploited or constraints overcome to reveal capabilities
4. Repeat steps for remaining functions

### The viewpoints

- *Cultural complexity:* Plays a role for resources where people are involved. People can be the resource (or part of the resource) or people can be affected by a resource.
- *Management complexity:* Plays a role when the use of software needs to be coordinated with other resources, e.g. people or business processes
- *Application complexity:* Is likely to be part of functions and resources that are part of product development or distribution.
- *Governance complexity:* Plays a role where documents are involved, which describe how a task needs to be carried out, e.g. corporate policies or laws and regulations

### **Creating FRAM for adopting cloud computing**

1. Go through functions of FRAM and investigate effects of cloud computing by analysing changes to resources of functions or changes in responsibilities
2. Adapt functions, delete functions, or introduce new functions to react to changes in resources or responsibilities. Then adapt capabilities accordingly.

# Appendix F – Ethics form

UNIVERSITY OF ST ANDREWS  
SCHOOL OF COMPUTER SCIENCE  
PRELIMINARY ETHICS SELF-ASSESSMENT FORM

This Preliminary Ethics Self -Assessment Form is to be conducted by the researcher, and completed in conjunction with the Guidelines for Ethical Research Practice. All staff and students of the School of Computer Science must complete it prior to commencing research.

This Form will act as a formal record of your preliminary ethics considerations.

PROJECT TYPE (please ✓)

- Staff  
 Undergraduate  
 Postgraduate  
 MSc  
 PhD

PROJECT TITLE

Services to the Cloud – Strategy evaluation of partners	
Name of researcher(s)	Marc Werfs
Name of supervisor (for student research)	Dr. Gordon Baxter, Prof. Ian Sommerville

OVERALL ASSESSMENT (to be signed after questions, overleaf, have been completed)

Self-audit has been conducted (Please ✓)

- YES       NO

Ethical Issues (Please ✓)

- There are **NO** ethical issues raised by this project  
 There are ethical issues raised by this project

Signed *M. Werfs* Print Name MARC WERFS Date 29.11.2012  
(Student Researcher(s), if applicable)

Signed *I. Sommerville* Print Name IAN SOMMERVILLE Date 30/11/2012  
(Lead Researcher or Supervisor)

This form must be date stamped and held in the files of the School Ethics Committee. The School Ethics Committee will be responsible for monitoring assessments.

## Appendix G – Glossary

**Aspect:** Functions in the FRAM are described through six aspects, namely: Input, Output, Precondition, Control, Resource, and Time. It is not necessary to describe all aspects of all functions. Instead, the analysis should focus on those aspects that are deemed important and for which information is available. For foreground functions, at least the Input and Output have to be described. For background functions, it is sufficient to describe the Input *or* Output.

**Background function:** Background functions are those functions that are important for the analysis but are not in its centre. In other words, background functions are included in the analysis because they might affect the performance of foreground functions but they do not vary in performance themselves. Background functions can be described in less detail as foreground functions. Over time, or in the course of the analysis, a background function can become a foreground function and vice versa.

**Capability:** Capabilities combine tangible and intangible resources in a structure way to achieve a specific task. Resources can be business processes, skills of employees, patents, knowledge, etc. In other words, they are organisational routines.

**cFRAM:** The cFRAM is an add on to the FRAM. The cFRAM allows companies to plan their move to the cloud by planning organisational changes and relating these to the technical adoption challenges of cloud computing. The organisational changes are structured through capabilities (hence the name cFRAM). With the cFRAM companies can identify their existing capabilities and investigate how these will be affected by cloud computing and, if necessary, what new capabilities need to be developed.

**Control (as aspect):** The Control aspect of a function supervises or regulates a function so that the desired Output is produced. Control can be a plan, a schedule, documentation documents or procedures, etc. But Control can also be about social expectations, e.g. the expectation of co-workers or customers.

**Foreground function:** Foreground functions, in comparison to background functions, are those functions that are at the centre of the analysis. In other words, they are more important than background functions. Over time, or in the course of the analysis, a foreground function can become a background function and vice versa.

**FRAM (Functional Resonance Analysis Method):** The FRAM can be used in a prospective analysis, to identify vulnerabilities in terms of safety, or in a retrospective analysis, to investigate the cause of an accident. It builds on the theory of socio-technical systems and systems are analysed through functions (see below) and aspects (see above). The FRAM has originally been designed for safety and accident investigations but is modified by this thesis to be used for planning organisational changes to increase resilience when adapting to new technologies.

**Function:** In the FRAM, a function refers to the activities – or set of activities – to achieve a goal. The function describes what people – individually or in a team – have to do in order to achieve that goal. Furthermore, the function captures everything necessary to achieve that goal, e.g. materials, documents, etc. Functions can either be foreground functions or background functions (see Glossary for further explanation). *The description of a function should be a verb or verb phrase.*

**Functional Resonance:** Functional resonance is the result that can be detected from the unintended interaction of the normal variability of functions. Functions, especially those performed by humans, vary in their performance every time, as they react to changes in the environment, for example. Most of the time, this variability is the reason why functions are carried out successfully. In some cases, however, the variability of several functions cause another function to fail (when that functions variability is too high). Hence, the term functional resonance as the correlations of the normal variability of functions can lead to non-linear effects. See sections 2.4.2 and 6.2 for a more detailed discussion.

**Input (as aspect):** The Input of a function is that which is used or transformed by the function to produce the Output. The Input can be anything from material to

information. The Input also starts a function. In other words, the Input needs to be detected or recognised by the function in order for it to start.

**Output (as aspect):** The Output of a function is that what is produced after the function is completed. Similar to the Input, the Output can be anything from material to information (e.g. as the outcome of a decision). The Output can also be the Input (or different aspect) for downstream functions.

**Performance variability:** See Functional Resonance

**Precondition (as aspect):** A function cannot start before all Pre-conditions have been fulfilled (i.e. are available or true). Pre-condition in itself, however, cannot start a function. Starting the function happens through the Input aspect. This rule can help in distinguishing whether something should be an Input or a Pre-condition.

**Resilience:** Resilience describes the ability of a function (or system) to perform in a dependable manner during changing conditions (which can be both expected and unexpected). Furthermore, resilience should be considered as something a system does rather than something it has. Hence, a system can be more or less resilient, but never resilient. A system is, for example, more resilient if it is able to bounce back to normal performance after a major negative event. See section 2.2 for a more detailed discussion.

**Resource (as aspect):** A Resource is used or consumed during the execution of the function. Resources can be anything from material and information to machines, software, or manpower. Time can also be a Resource but is treated separately (see Glossary). A Resource is consumed during the execution of the function, which means that there will be less of it after the function has been executed. Execution Conditions, on the other hand, need to be available during the execution of the function but are not diminished during execution. The difference between Precondition and Executive condition is that the former is required before the function starts and the latter during the execution of the function.

**Socio-technical system (STS):** The term socio-technical emphasizes the importance of the interdependencies and interactions between the social and technical elements of systems, which can lead to emergent behaviours. STSs involve a complex interaction between different levels of organisation, in particular, between people, technology, and the environment in which the systems are deployed. See section 3.1 for a more detailed discussion.

**Technological discontinuity:** The theory of technological discontinuities claims that technological progress is evolutionary with rare events of discontinuous change (Tushman & Anderson 1986). These events create a major technological shift, which can be classified as either *capability-enhancing* or *capability-destroying*. Capability-enhancing discontinuities build on existing capabilities and improved attributes like price or performance. They can replace older technologies but do not require companies to develop new capabilities to exploit the technology. Capability-destroying discontinuities require companies to develop new capabilities as existing ones can become inappropriate for the technology. These kinds of discontinuities can either create a new product class or substitute an existing product. See section 1.1 for a more detailed discussion.

**Technology uncertainty:** The term technology uncertainty aim to capture how some of today's technologies affect companies in novel ways. Recent technologies work differently by taking away control from the IT department and giving it to a third party, like cloud computing which gives control to the cloud provider. IT departments also rely more on bigger and more connected systems (or systems of systems) that are vulnerable to unforeseeable and cascading failure events. See section 2.3 for a more detailed discussion.

**Time (as aspect):** Time can be considered as a form of Control. For example, Time can mean that a function is not allowed to start before another function has been completed (or parallel to, or before, etc.). Time can also affect the execution of a single function, e.g. in relation to either clock time or elapsed time (e.g. that a function has to be completed before 3pm). In that case, Time would be seen as a

Resource. Time can, however, also be seen as a Pre-condition, e.g. when a function is not supposed to start before 3pm.

**Use uncertainty:** The term use uncertainty aims to capture how users today create uncertainty by the way they use certain technologies. Users increasingly expect to use the latest technologies because they also use them in other areas. Furthermore, some of these technologies can be used without the IT department knowing about them (e.g. smartphones or cloud computing). See section 2.3 for a more detailed discussion.