Challenges of Traditional Usability Evaluation in End-User Development

Daniel Rough[1] and Aaron Quigley[1]

School of Computer Science
University of St Andrews, Scotland, UK
djr53,aquigley@st-andrews.ac.uk

Abstract. End-user development (EUD) research has yielded a variety of novel environments and techniques, often accompanied by lab-based usability studies that test their effectiveness in the completion of representative real-world tasks. While lab studies play an important role in resolving frustrations and demonstrating the potential of novel tools, they are insufficient to accurately determine the acceptance of a technology in its intended context of use, which is highly dependent on the diverse and dynamic requirements of its users, as we show here. As such, usability in the lab is unlikely to represent usability in the field. To demonstrate this, we first describe the results of a think-aloud usability study of our EUD tool "Jeeves", followed by two case studies where Jeeves was used by psychologists in their work practices. Common issues in the artificial setting were seldom encountered in the real context of use, which instead unearthed new usability issues through unanticipated user needs. We conclude with considerations for usability evaluation of EUD tools that enable development of software for other users, including planning for collaborative activities, supporting developers to evaluate their own tools, and incorporating longitudinal methods of evaluation.

Keywords: End-User Development · Usability · Case Studies

1 Introduction

Creating end-user development (EUD) tools that support users in their working practices is a significant challenge, compounded by the difficulty in evaluating their success in doing so. In the deployment of any novel technology in a professional environment, intended users' interactions with this technology depend on organisational factors, including other individuals with whom communication and collaboration take place, or existing technology used in working practices. As such, EUD tools are intended to address this difficulty of anticipating the needs of end-users in advance, by providing the flexibility to adapt software to their context-specific needs.

Given the contextual influences of EUD in practice, it is somewhat surprising that a prevalence of lab-based usability studies in the evaluation of EUD is contrasted by the lack of research into their real-world utility [23], a disparity recognised within HCI as a whole [8]. EUD evaluations are largely focused

on the programming paradigm and how users' mental models of programming tasks affect the usability of particular paradigms. However, successful deployment of EUD tools requires knowledge of who the potential end-users are, what their goals and motivations are, and how such tools could fit within their current working practices. In this regard, Mehandjiev et al. highlight a lack of *"necessary knowledge of how to deal with problems and conflicts which are likely to emerge from the formalization of EUD"* [14]. A recent review by Barricelli et al. explicates the breadth of EUD research and the contexts in which it is applied [1], from personal web mashups to complex industry-standard software. Thus, an EUD tool's ease-of-use is contingent not only on the development paradigm, but on the domain in which it is employed, its users, and other external conditions.

In this paper, we show that the external variables pertaining to ease-of-use cannot be resolutely determined for EUD, posing a challenge to lab-based evaluation. We discuss the issues and related requirements emerging from a lab-based think-aloud usability study of *Jeeves*, our EUD tool. Following this, two case studies are described where Jeeves was employed by psychology researchers to address their own research questions. These studies were intended to enable analyses of Jeeves in its context of use, with results expected to reinforce those of our lab-based usability study. However, this was not the case, challenging the established view of the efficacy of lab studies in professional EUD contexts.

1.1 Related Work

Prior research has attempted to understand, and consequently bridge, this evaluation gap between the lab and the real world. Field methods such as contextual inquiry provide an understanding of usability "in use" and thereby external validity [25]; log data of user actions provides unobtrusive *in-situ* usage; longitudinal approaches such as the Experience Sampling Method (which Jeeves aims to facilitate, incidentally) can be employed to collect usability issues from users as they occur [12]. Such methods aid understanding of software usability outside the lab, but are seldom employed in EUD usability evaluation [23].

Irrespective of this preference for lab usability studies, continuous co-design with software end-users is a core component of an EUD approach. This is formalised by Fischer et al. through the *Seeding, Evolutionary Growth and Reseeding (SER) model*, which recognises the need for continuous re-evaluation and restructuring of tailorable systems [5]. The SER model supports Fischer's *meta-design* framework, advocating users as participating designers of software during use [6]. A pertinent example of meta-design in practice is described by Maceli, whose case study into the co-design of meme creation tools [13] showed how developers and end-users naturally engage in meta-design to improve their tools in the absence of formal research processes.

In short, we as meta-designers, must respond to in-use evaluation if our EUD tools are to be successfully employed. How, then, do lab usability studies help or hinder our identification of in-use issues? The remainder of this introduction provides an overview of Jeeves to afford context for our own lab and field studies.

Jeeves and its Context of Use Jeeves is an EUD tool intended for non-programmer researchers to create smartphone apps that collect data from participants as they go about their everyday lives, based on the aforementioned Experience Sampling Method (ESM) [12]. Jeeves employs a blocks-based programming paradigm through which researchers define different time and context-based triggers upon which to execute actions (primarily sending surveys, but also sending prompts or capturing contextual information such as location). Recent additions to the library of Jeeves blocks include participant "attribute" blocks, akin to programmatic variables, conditional statement blocks, and context-sensitive triggers. These extensions were derived from a review of literature detailing the potential benefits of modern smartphone ESM, and were positively received by interviewed psychology researchers [20]. However, the researchers who were interviewed did not actually use Jeeves.

Other novel ESM creation tools exist both in research and the commercial domain, many of which are listed by van Berkel et al. in their review of mobile ESM [3]. However, there is a notable lack of research into challenges of introducing ESM creation tools into practice. One exception is the work of Batalas and Markopoulos [2], who provide a detailed discussion of results related to real-world use of their *TEMPEST* platform. We seek to build upon this work by focusing on the contrasting results of different evaluation methods.

2 Lab-based Usability Study

This paper focuses on a real-world contrast with the third lab-based usability study undertaken with Jeeves. (We refer the reader to [21] and [22] for details of prior studies.) This study was intended to focus on issues encountered in completion of complex tasks with the newly implemented extensions. Participants were 10 students at our university, recruited via advertisement in weekly student memos, and through circulating emails to students in the school of psychology. In total, six participants studied psychology, two studied medicine, and two studied humanities, with a mix of undergraduate and postgraduate students. Three psychology students reported experience with MATLAB, but no other programming experience was stated.

Prior to running this study, a 10 minute tutorial video was shown to guide participants through the necessary information they would need to complete the study tasks, by demonstrating an example app specification being built. This also served as a useful reference for when participants were unsure how to proceed. Participants were instructed to think aloud as they completed their tasks, in order to understand *why* specific issues were encountered, but also to explicate participants' mental models of triggers, conditions and attributes.

2.1 Tasks

Nielsen's guidelines on designing study tasks were followed closely, by ensuring that participants were not primed with the trigger-action terminology of

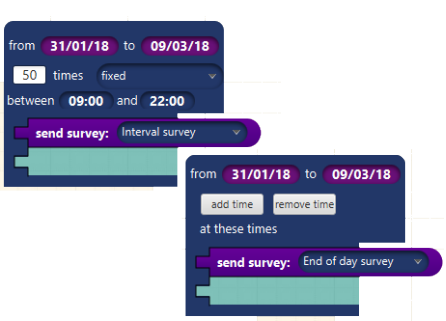

Fig. 1. Two of the 'faulty' triggers in the Task 3 specification

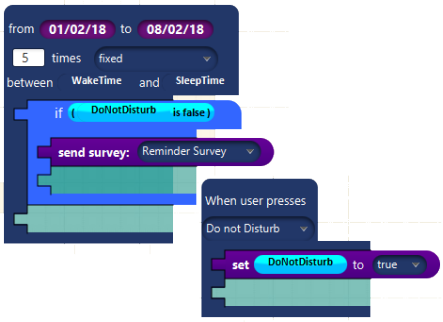

Fig. 2. An example 'Do Not Disturb' specification in Task 4

Jeeves [16]. The tasks were intended to address updated functions of Jeeves that were perceived as useful for researchers to create personalised ESM study specifications [20] - primarily the use of "attribute" blocks to represent variables, as well as conditional statements. Note that the hypothetical app end-users are referred to in study tasks as *patients* rather than *participants*, to distinguish them from the participants in this study. For the sake of space, we do not quote the tasks verbatim. Instead, we describe the features each was intended to evaluate.

Task 1 - Attribute Usage: Participants were asked to design a specification that would acquire patients' waking and sleeping times, and trigger a survey at random times during the patients' waking hours. This task assessed whether the sequence of creating attributes, assigning values to them, and then using these attributes in the blocks specification, was understandable by participants.

Task 2 - Survey Button Creation: Participants were asked to design a specification allowing a patient to enter data upon pressing a button (i.e., event-contingent experience sampling [12]). Further, they were asked to utilise actions that would capture sensor data, including Bluetooth and GPS.

Task 3 - Patient Compliance Reaction: Participants were asked to load a study specification that had been populated with simulated data of poorly compliant patients. The compliance issue was due to a fault in the specification causing a trigger to send surveys 50 times a day. Further, one survey was not sent at all because its trigger did not have a time, both of which are shown in Figure 1. This task primarily assessed the readability of the blocks notation.

Task 4 - Do Not Disturb: Finally, participants were asked to implement a "Do Not Disturb" button that would stop patients receiving prompts, an example solution of which is shown in Figure 2. This was a comparatively difficult task that assessed setting attributes through actions, and adding conditional statements into triggers. A similar application was demonstrated in the tutorial video, which participants could adapt to the task's requirements.

2.2 Usability issues analysis

The study provided insights into how participants would tackle the more advanced features of Jeeves, with an aim to address problems that could lead to researchers' frustration and consequent abandonment of the EUD tool. Think-aloud monologues and post-study interviews were transcribed, and analysed in parallel with screen-capture recordings of their task completion.

While the blocks-based programming paradigm was found to be intuitive and liked by participants, who provided positive feedback on the visual metaphor, they also experienced confusion and frustrations that sometimes led to fundamental design breakdowns. The following issues were encountered most frequently or severely, and illustrate the primary concerns we had regarding the usability of Jeeves for a realistic deployment.

Issue 1 - Hidden dependencies

Creating and assigning attributes in Task 1 raised issues with most participants, who were confused by the sequence of creating an attribute, creating a survey, assigning the attribute to the survey, and then sending the survey. Participants frequently attempted shortcuts by dragging and dropping attribute blocks into incorrect places in a misunderstanding of each step's purpose. It also gave rise to barriers where participants were oblivious to the fact that they had missed one of these steps. These instances of what Green and Petre define as *"hidden dependencies"*, suggest that a shorter sequence of actions might be necessary, or dependencies communicated more explicitly to users [7]. As attributes are created in one section of Jeeves, and applied in another, this caused confusion and provoked suggestions of alternative solutions from participants:

"Like in this window, you [define an attribute] in this window, in the survey design. But in the end you are using it [on the blocks canvas] so, I mean it transfers but for me it was confusing that you did it here" (P8)

Issue 2 - Too much abstraction

Issues were caused by unsuitable levels of abstraction. Five participants struggled to find a trigger that would only fire once. While this simply involved customising a timed trigger to fire at a single time, this was unclear to participants, who hunted for a more specific "one-off" trigger type. Over-abstraction also resulted in participants experiencing a barrier when attempting unavailable customisation. For example, one participant wished to specify the desired Bluetooth data returned from the "Capture Data" action:

"*Capture data from...will this do? I'm not sure whether this is enough. Do I have to do anything about this Bluetooth? Um...I'm not sure"* (P4)

Conversely, one participant suggested the possibility of creating his own abstractions out of more specific components for ease-of-use:

"So I have my press commands and maybe [I could] group them...maybe you could pull them onto each other and make them into a group" (P6).

Issue 3 - Gulf of Evaluation

Task 3, where participants were asked to read a pre-created specification, gave rise to frequent evaluation barriers. Although participants could observe that patients were not completing surveys, six participants missed at least one

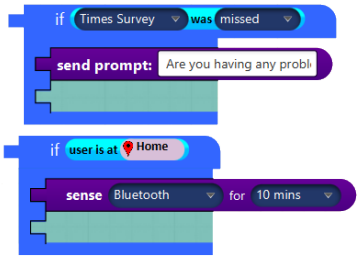

Fig. 3. If-conditions were misinterpreted as event-based triggers

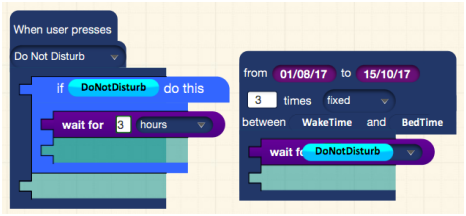

Fig. 4. Two erroneous attempts at 'Do Not Disturb' functionality in Task 4

of the faulty triggers shown in Figure 1. This suggests a *"Gulf of Evaluation"*, defined by Norman as *"the amount of effort that the person must make to interpret the physical state of the device and to determine how well the expectations and intentions have been met"* [17, p. 39] As Jeeves does not currently provide feedback on the function of created specifications, some participants expressed doubt as to the correctness of their solutions to tasks, or were unaware of subtle mistakes that could cause faults in the apps they created. Some participants expressed a desire to test their apps, enabling trial-and-error learning:

"Can I run it and see like what I've placed and understand what's gonna happen? Like when you do a webpage you can see it straight away like what's happening that's what you do when you learn it online." (P9)

However, most participants were satisfied with their task completion and did not outwardly question their solutions, whether they were correct or not.

Issue 4 - Events and States

Task 4 cased the majority of issues for participants in attempting to combine event and state triggers. As found in prior work by Huang and Cacmak, participants had trouble separating triggers involving discrete, instantaneous events, and continuous, ongoing states [10], such that combining the trigger *event* with the "Do not Disturb" *state*, as shown in Figure 2, introduced barriers for most participants. P2 initially assumed that an if-condition detached from a trigger would enable a change in the value of an attribute to be detected like a discrete event: *"I'm going to assume that this (if-condition) will be continuously running and that it doesn't need to be attached to a trigger object"* (P2) Although the if-condition block has an external connector to afford nesting within a trigger, the visual notation did not stop participants detaching it. Other examples of this error during task completion are shown in Figure 3.

Issue 5 - Lack of abstraction

In addition to barriers caused by over-abstraction, further barriers in Task 4 arose through *under*-abstraction. Five participants continuously searched for a "Do not Disturb" action, rather than implementing this functionality themselves. P5 explained this behaviour in her post-study interview:

"I realise a lot of the things that I wanted to do could be composed of the triggers and actions and conditions themselves but I guess for me, they'd be kind of like, simpler options? Or simpler components?" (P5)

As suggested by a participant in resolving Issue 2 (too much abstraction) further research could establish with researchers what useful fundamental blocks would be, so that commonly employed features could be constructed once and then reused as custom abstractions when necessary.

Issue 6 - Ambiguous Actions

Finally, the apparent ambiguity of the previously implemented "wait for" action was a notable source of error, causing barriers for many participants. This action was intended to pause execution of subsequent actions in a particular trigger. However, participants assumed that this action would pause notifications across the entire application, shown by one participant's faulty specification in Figure 4, suggesting a misunderstanding of trigger concurrency. Further, three participants attempted to use a "Do Not Disturb" attribute with this action, in an attempt to wait until the attribute was false (Figure 4, right).

2.3 Summary

Our observations, collated with direct (interview) and indirect (think-aloud) participant feedback, suggested that Jeeves could be applied with no prior programming experience, but that issues of *hidden dependencies*, *abstraction*, *evaluation*, and *ambiguity*, could lead to abandonment by researchers attempting more sophisticated behaviour. P5, a medicine postgraduate, explained that 'walk-up-and-use' functionality is not expected from highly useful software:

"a lot of programs you use in research you do need training, like SPSS... you have to use YouTube videos or you have to go on a course. It's not unusual for researchers to be used to having to do tutorials, classes, sessions..." (P5)

However, even if perceived usefulness overshadows ease-of-use in software, as suggested by Greenberg and Buxton [8], a follow-up quote from the same participant cemented the ongoing need to keep ease-of-use above a certain threshold:

"There is a frustration tolerance. You run the risk that people like me would get to this phase and go 'y'know what? I don't know how to do this. I'm just gonna email surveys through Qualtrics because I know what to do'." (P5).

3 Case Studies

Prior to updating Jeeves based on the lab study's feedback, the two case studies described in this section were conducted, with the intention of triangulating emergent "real" usability issues with participants' task-specific issues. The assumptions made about researchers' typical usage of Jeeves was a clear limitation of the study itself; while tasks were informed by publications in psychology journals that utilised ESM, the constraints of a lab study are not representative of practical application of an EUD tool in its intended context of use. It was therefore of interest to determine to what extent these usability issues would impact on researchers' use of Jeeves and if new usability issues would arise.

3.1 Case Study 1 - ESM During Sport Events

This study was conducted in collaboration with a psychology researcher at a local university and a researcher at a university in Germany, whom we refer to as Paul and Oliver in this paper. Both researchers had an interest in capturing experiences of fans during sporting events, but their knowledge of programming would not allow them to do this themselves, thus Paul saw how Jeeves could be used for this purpose. This section summarises events of interest within the case study, pertaining to study organisation, piloting and running the full study.

Study preparation In November, email correspondence began with Paul and Oliver in which requirements of a potential study were ascertained. It was decided that the goal was to conduct an ESM study with supporting fans of a basketball team at Oliver's university during a live game, prior to which a pilot study would be run with local students watching a live football game on television.

In early January, a Skype call was held, during which the researchers watched the video tutorial of Jeeves and used the screen-sharing function of Skype to collaboratively design a study specification.

During the call, collaborative completion of the study between Paul and Oliver was observed to be difficult. Oliver would dictate survey questions from the plan document while Paul created the survey in Jeeves, which was slow and cumbersome. Complications further arose when a means to obtain informed consent from participants had to be implemented into a survey. Initially, Paul copied the text from the PDF consent document, but it was then incorrectly formatted. It was suggested that providing participants with a URL link to the informed consent document would be simpler, which was agreed upon.

The lack of preview functionality for surveys resulted in difficulty, and an inability to duplicate similar questions also became an issue. Both Paul and Oliver made comparisons with *Qualtrics* - software they were both familiar with in their research, as illustrated in this dialogue:

> **Paul**: "Is there a way of previewing questions? I mean I guess it's kinda here, that'd be really useful. That's something *Qualtrics* does and it'd be quite useful. There's not a way of copying a question is there?"
> **Oliver**: "Paul, you also use *Qualtrics* right? I think it has very...smart features, especially what you said, copying questions, preview of questions, and also these randomisation things, orders, stuff like that."
> **Paul**: "Yeah there's a lot of good stuff in *Qualtrics*, it can't do everything we want it to do, but in terms of user features it might be worth..."
> **Oliver**: "Yeah I have to agree, but they sell this for a lot of money so..."

Follwing the Skype call, Paul suggested that a form of annotation would be desirable for communicating ideas to Oliver:

"...to add a comment, annotation...a note to yourself to say 'I've still got to do this' or 'remember to change that' or in a collaborative project, 'I'm not sure how this works' or 'what do you think of this?'...just to say 'Oliver this is for the half-time survey, just starting it for you, you finish it' "

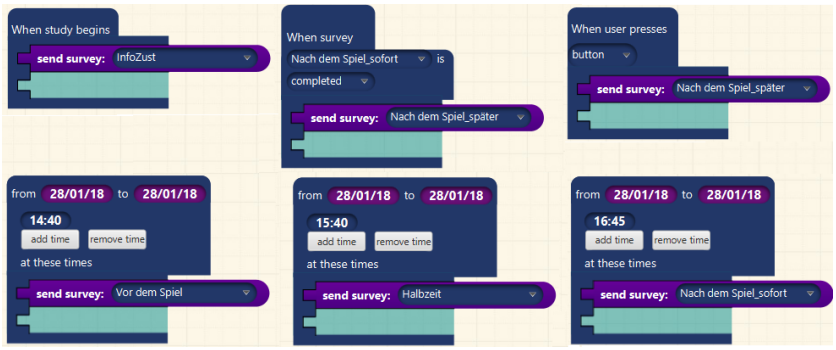

Fig. 5. The final specification for the study conducted at the basketball game

Conducting the pilot study On January 13th, the pilot study was run. The first author was responsible for conducting this study, such that no issues were directly experienced by Paul and Oliver. While most participants faced no problems, there were exceptions not previously considered in lab evaluation.

For example, one participant had an incompatible device, running a lower version of Android than was necessary for the study. Another participant had privacy settings enabled on their device, so that they did not receive surveys at the same time as other participants. One participant turned up particularly late, and by the time they had installed the Jeeves Android app and initiated the study, they had missed the first trigger. The time for this trigger was adjusted through Jeeves so that it would be sent to their device, which meant that it was also sent to all other devices, causing confusion amongst participants.

Preparation for full-scale study A Skype call was set up to discuss the results of the pilot study and to plan for the full-scale study in Germany. Only the audio of this call was recorded, as no use of Jeeves took place.

Oliver was responsible for recreating the pilot study, with survey questions written in German, and trigger times adjusted to key phases of the basketball game. Given that the design was otherwise identical, Oliver commented that a feature to simply duplicate the pilot study specification would have been useful. At this stage, Oliver had a greater workload, involving the translation of the previous Jeeves pilot study into German, testing the new app (Paul did not own an Android smartphone and thus was unable to do so) as well as engaging in recruitment activities with the university sports team.

Given the various organisational activities involved, as well as the researchers' other commitments, development activities were put on hold. After the last update by Oliver on January 23rd, no further updates were made until January 27th - one day before the full study - when a bug was discovered in which participants who had registered were already being sent study surveys.

Running the full study The full-scale study was run as planned; 40 participants initially signed up, and 30 completed every survey. Due to the variation of basketball match times caused by fouls and timeouts, Oliver was present at the match to adjust the half-time and full-time surveys as necessary, to ensure that participants would receive surveys at the appropriate times. However, in the final study specification, Oliver had left the "Button Trigger" and button he created for testing purposes in the version of the app that participants downloaded. Some participants found this button and ended up completing the post-match survey too early. (This trigger can be seen in the top-right of Figure 5, showing that when the button is pressed, a post-match survey is sent.)

Figure 5 further indicates the simplicity of the researchers' specification. Indeed, the only implementation issues encountered by Paul and Oliver with Jeeves were instances of unavailable functionality (i.e., presentation of participants' informed consent forms) for which the authors either implemented the requested function, or the researchers found workarounds as required.

3.2 Case Study 2 - ESM in the Menstrual Cycle

This case study describes the progress of a collaboration with a second psychology researcher at a local university, and her postgraduate research student, whom we refer to as Deborah and Lucy. Deborah's area of research is in aggression, for which Jeeves was considered suitable for exposing contextual factors of aggressive behaviour, outside the constraints of the traditional laboratory experiment. Lucy's thesis project involved investigating the general variation of female aggression during the menstrual cycle.

Unlike the previous case study, where the first author was involved as a collaborator, the role taken here was as a passive observer. This precluded direct involvement with Deborah and Lucy, such that face-to-face meetings on the project were often not observed. However, insight was obtained through direct observation of their use of Jeeves, as well as frequent email feedback.

Study preparation In-person meetings were arranged in November to plan the preliminary tasks that would need to be undertaken prior to designing the study specification. Again, ethical documentation had to be submitted and approved, which delayed progress. A further meeting was held at the end of January to discuss the study's requirements, and the capabilities of Jeeves in fulfilling them. Deborah and Lucy watched the Jeeves tutorial video in order to understand the available features. Rather than beginning to implement the specification immediately after watching the video as before, a week passed during which Lucy planned her study design, before the next meeting.

While Deborah and Lucy's research question ultimately determined their study design, in this case the use of attributes was required in order to tailor the app to each participant's ovulation dates. Attribute creation appeared to be straightforward. Lucy created a survey question, created the date attribute, and then assigned the attribute to the question with no further issues, unlike the usability study participants, who appeared to struggle with this sequence:

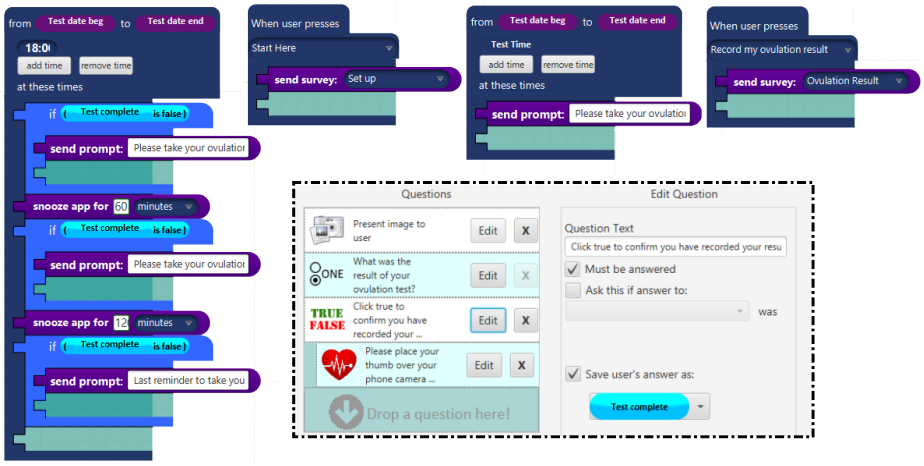

Fig. 6. Deborah and Lucy's pilot study specification, also showing part of their created survey (with an attribute designating confirmation of completion)

"you can create a survey that would get all the attributes out for the times so...then we can create an 'if this date send this survey' so that when it gets to the correct date they will just get another trigger" (Lucy)

An issue of abstraction arose when the researchers wished to prompt a participant at a particular time if they had not completed their survey, but with no simple block that would allow them to do so, a clever workaround was employed, as shown in the blocks specification and adjacent dialogue in Figure 7. In summary, the researchers had to add a question to their survey that would ask participants to confirm survey completion, thereby updating the "Test complete" attribute to stop reminder prompts being sent. (Researchers designed this workaround survey as shown in Figure 6.)

Unlike the crippling issues experienced by lab evaluation participants in combining triggers with conditions and interpreting trigger concurrency, Deborah and Lucy experienced only minor issues, which were resolved quickly through discussion and referral to the tutorial video. After 45 minutes, the researchers had finished designing their study, and expressed satisfaction that they had independently implemented the specification in this short time.

Conducting the pilot study Following the direct observation of study implementation, a series of circumstances arose that prevented the pilot study actually being initiated until one month later. The specification was not viewed by the researchers during this time (as indicated by the "last accessed" date and time feature of Jeeves). However, in the interim period, the researchers asked if functionality to capture participants' heart rate could be added. This resulted in a hasty integration of functionality from an unofficial online source, which it was not possible to rigorously test in the short time prior to study deployment.

Deborah: If when we send the user input, when they do their 'please input your ovulation results' survey, we can add a question at the end that's just 'click here to confirm you've taken the test today' then you've got a yes/no answer to save it as an attribute, which can then be placed into that condition.
Lucy: Okay. We could just put 'Click true to confirm you have recorded your results'. Oh, should I drag this [test complete attribute]?
Deborah: Yeah, 'if test complete is false'

Fig. 7. Researchers' workaround to implement detecting survey non-completion

The pilot study ran through March for 21 days, during which participants were required to report their ovulation result daily for 10 days of the study. Deborah and Lucy reported that the study had been a success, and were particularly pleased with the new heart rate functionality that worked without problems. However, other unforeseen issues arose, unrelated to their specification design:

"Participants were unsure about the permanent notification that said 'Jeeves running' on their phone...Two participants dropped out of the study saying that the app was 'annoying' them due to this" (Lucy)

It was surprising that the small notification icon would cause such irritation as to lead to study drop-out. The icon appears irrespective of specification design, such that a preview feature would not have helped researchers correct this.

4 Discussion

Between our two pairs of psychology researchers, there were some notable differences in their application of Jeeves. The studies took place over different time periods, contrasted in complexity, and were designed and tested by researchers through different processes. Common to both, however, were usability issues and requirements that could not be anticipated from a lab evaluation. We frame our insights as considerations for the usability evaluation of *"public EUD"* tools - wherein one group of end-users develops software for a separate group [4].

4.1 Plan for collaboration

In Paul and Oliver's case study, remote collaborative use of Jeeves emerged as a practice we had not considered. Previously, it was assumed that a single researcher would be responsible for EUD activities in a group collaboration, allowing single-user, task-based usability studies to retain some external validity. However, if the EUD task is a group effort, as it may often be, this introduces

additional factors that cannot be explicated in such a lab study. Pipek and Kahler discuss how collaborative tailoring may take place in single-user applications, as *"shared context scenarios"* [19]. Indeed, our lab study probed Jeeves from a single-user perspective, but we discovered that collaborative use can occur regardless of an EUD tool's capacity to deal with it.

Mitigating the potential difficulty of evaluating an EUD tool in a real-world deployment, discount methods have been proposed that conceptualise group activities as *"mechanics of collaboration"* [18], which focus on basic actions such as communicating with group members, or keeping track of members' activity. We suggest that such methods be incorporated into preliminary evaluations of a new EUD tool where there is potential for shared use.

4.2 Evaluate outcome quality objectively

In categorising measures of usability, Hornbæk distinguishes between *"outcome quality"* and *"perceived outcome"* as objective and subjective measures of usability respectively [9]. In designing the usability study, our measure of *outcome quality* was the accuracy of participants' final solution to a task. However, in the absence of an objective task-based measure, this became an issue for our case study researchers who were dissatisfied with the uncertainty of their *perceived outcome*:

"Whenever I do an online survey, I preview it and preview it and preview it multiple times, run through it, there are always errors...so I found that really frustrating that I couldn't actually see what it was that I'd coded." (Paul)

In EUD of ESM apps, Batalas discusses this tension between success from the perspective of the tool developer, and that of the researcher who uses the tool [2]. While the perceived outcome of a public/outward EUD tool can, to some extent, be measured through a preview function for researcher developers, the objective *outcome quality* is determined by the target group's response to the developed app. For example, in both our case studies, participants experienced issues that would not have been detected through an app preview. Thus, an ideal EUD tool should support the end-user developer in evaluating their own software with a target group, through a real-time feedback feature, for example.

4.3 Ensure learnability and retention

Irrespective of the assumptions of *what* researchers would do with Jeeves, further assumptions were made as to *how* and *when* their EUD activities would take place. While Paul and Oliver immediately began drafting a study specification after watching the tutorial video, Lucy took time to learn the functions of Jeeves prior to creating her specification. Thus, the **hidden dependencies** and **ambiguity** issues that hindered lab study participants were apparently surmountable through brief practice. This aligns with results of Mendoza and Novick, who show that prominent initial issues are often overcome through continued use [15].

However, "continued use" is unrealistic in some EUD contexts. As Tetteroo et al. observe, a successful EUD deployment is not necessarily that which is used

daily [24]. In our case studies, the time involved in preparation and data analysis dwarfed that of the time actually spent using Jeeves. Further, even during the period of specification development, researchers' use of Jeeves was often spread several days apart. Indeed, depending on the nature of research, separate ESM studies could themselves be months apart.

We suggest that our lab study unearthed the *wrong type* of usability issues that only occur in a single usage session. This is certainly why we observed that, even in creating a complex study, Deborah and Lucy did not encounter the pertinent lab study issues. Where infrequent, sporadic usage patterns of EUD are likely to occur, more emphasis should instead be put on evaluating **learnability** and **retention** [9]. Again, reliance on a single-user, single-session evaluation method cannot provide a full picture of usability of an EUD tool over multiple and periodic instances of use. Retention, however, is relevant insofar as researchers choose to return, and the following two points capture how improving ease-of-use through simplification could lead to immediate abandonment.

4.4 Usefulness first; ease-of-use later

Upon opening Jeeves with the intention of designing a new specification, researchers are faced with a choice - invest time in learning or re-learning this interface, or abandon it and return to familiar software? We assess how participants use Jeeves, but not why they might not attempt to do so when alternatives (such as *Qualtrics*, endorsed by Paul and Oliver) may be available. Deborah expressed how functionality issues eventually overshadowed ease-of-use in achieving her statistical analysis:

"SPSS is very easy to pick up, but you reach a point very quickly where what you want to do is beyond the scope of what it really does and then you have to give up and move to R and start at the bottom of the learning curve again"

A key concern of our usability study was that many participants expressed feeling initially "overwhelmed". However, it appears that the danger of iterative lab studies is not only that we continue to refine a sub-optimal design, but also that we may sacrifice necessary functionality in pursuit of usability goals. A quote from Oliver acutely exemplifies this danger:

"For us, the research question is very important - that really determines which study design we have - and that determines which tool we use, and NOT vice versa. You don't do a study just because Jeeves exists."

4.5 Plan for shifting goals

We could try to determine a minimal but comprehensive feature set required by researchers, and subject these features to lab-based usability evaluations, but this also poses an issue. The ISO standard of usability describes it as *"The extent to which a product can be used by specified users to achieve* ***specified goals*** *with effectiveness, efficiency and satisfaction in a specified context of use."* [11].

The notion of "specified goals" can be easily applied to non-EUD software where the functionality of the software is constrained to writing a document

or ordering a product online, for example. However, even in our attempts to introduce as much flexibility into Jeeves as possible, effectiveness cannot be easily measured by task completion when the tasks are at the liberty of users' imagination. A quote from Paul highlights a need for continuous innovation:

"As [Oliver] says, every study's different, and we're always dreaming up new and daft things and ways of asking things to participants...so there would always be a desire for more features. That will always happen."

Thus, usability is only relevant for as long as an EUD tool meets users' needs. With this consideration in mind, usability evaluation must be integrated into the SER process model, ensuring that not only the needs of users, but also the ease with which they can fulfil these needs, are accounted for.

5 Conclusion

By comparing a lab-based usability study and two case studies of an EUD tool in practice, we identified clear limitations to the insights that can be acquired through the former, when its use in practice is unconstrained and subject to unexpected changes. Perceived usability of Jeeves in practice was highly dependent on collaborative use, over multiple intermittent sessions, to develop software that meets the diverse and shifting requirements of both the researchers and their participants. Dispensing with the *individual* EUD view of single users developing for themselves, *public* EUD tools introduce a range of contextual variables that force different approaches to evaluating their usability.

By making this distinction of Jeeves as a public EUD tool, we do not suggest our findings are applicable to all EUD environments - many of which are intended for educational purposes, or for personal creations and customisations [1]. In such instances, the tasks that end-user developers desire to complete may be well-defined, developed software may be objectively assessed by its developer, and usage context may be predictable. Indeed, previous usability studies of Jeeves were critical in identifying major issues and bugs that detracted from user experience; such studies are still insightful if employed at key times.

Nevertheless, in just two field deployments of Jeeves, we identified new insights into *how* and *when* it is used that preclude evaluation in a lab setting. We conclude by reiterating that an EUD tool's usability, just like its usefulness, is inextricably linked to its context of use, and evaluations must go beyond the development paradigm and into this development context.

References

1. Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A.: End-user development, end-user programming and end-user software engineering: A systematic mapping study. Journal of Systems and Software **149**, 101–137 (Mar 2019)
2. Batalas, N.: EMA IDEs: A Challenge for End User Development. In: End-User Development. pp. 259–263. Springer International Publishing, Cham (2015)

3. Berkel, N.V., Ferreira, D., Kostakos, V.: The experience sampling method on mobile devices. ACM Computing Surveys (CSUR) **50**(6), 1–40 (2017)
4. Cabitza, F., Fogli, D., Piccinno, A.: "Each to his own": distinguishing activities, roles and artifacts in EUD practices. In: Smart Organizations and Smart Artifacts. pp. 193–205. Springer International Publishing, Cham (2014)
5. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A., Mehandjiev, N.: Meta-design: A manifesto for end-user development. Communications of the ACM **47**(9) (2004)
6. Fischer, G.: End-user development and meta-design: Foundations for cultures of participation. In: End-User Development. pp. 3–14. Springer Berlin (2009)
7. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a 'cognitive dimensions' framework. Journal of Visual Languages & Computing **7**(2), 131–174 (1996)
8. Greenberg, S., Buxton, B.: Usability evaluation considered harmful (some of the time). In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 111–120. Florence, Italy (2008)
9. Hornbæk, K.: Current practice in measuring usability: challenges to usability studies and research. Int. Journal of Human-Computer Studies **64**(2), 79–102 (2006)
10. Huang, J., Cakmak, M.: Supporting mental model accuracy in trigger-action programming. Proc. ACM UbiComp '15 pp. 215–225 (2015)
11. ISO: Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts (2018), https://www.iso.org/standard/63500.html, accessed 18/02/19
12. Larson, R., Csikszentmihalyi, M.: The experience sampling method. In: Flow and the foundations of positive psychology, pp. 21–34. Springer (2014)
13. Maceli, M.: Co-design in the wild: A case study on meme creation tools. In: Proceedings of the 14th Participatory Design Conference. pp. 161–170. ACM (2016)
14. Mehandjiev, N., Sutcliffe, A., Lee, D.: Organizational view of end-user development. In: End User Development, pp. 371–399. Springer Netherlands (2006)
15. Mendoza, V., Novick, D.G.: Usability over time. In: SIGDOC '05. p. 151. ACM Press, New York, NY, USA (2005). https://doi.org/10.1145/1085313.1085348
16. Nielsen, J.: Usability Engineering. AP Professional, Boston, MA, USA (1993)
17. Norman, D.: The design of everyday things: Revised and expanded edition. Basic Books, New York, NY, USA (2013)
18. Pinelle, D., Gutwin, C., Greenberg, S.: Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. ACM Trans. Comput.-Hum. Interact. **10**(4), 281–311 (Dec 2003)
19. Pipek, V., Kahler, H.: Supporting Collaborative Tailoring, pp. 315–345. Springer Netherlands, Dordrecht (2006). https://doi.org/10.1007/1-4020-5386-X15
20. Rough, D., Quigley, A.: End-user development in social psychology research: Factors for adoption. In: VL/HCC, 2018 IEEE Symposium. pp. 75–83 (Oct 2018)
21. Rough, D., Quigley, A.: Jeeves-a visual programming environment for mobile experience sampling. In: VL/HCC, 2015 IEEE Symposium. pp. 121–129. IEEE (2015)
22. Rough, D., Quigley, A.: Jeeves-an experience sampling study creation tool. BCS Health Informatics Scotland (HIS) (2017)
23. Tetteroo, D., Markopoulos, P.: A review of research methods in end user development. In: End-User Development. pp. 58–75. Springer International, Cham (2015)
24. Tetteroo, D., Markopoulos, P.: EUD survival "in the wild": Evaluation challenges for field deployments and how to address them. In: New Perspectives in End-User Development, pp. 207–229. Springer International Publishing, Cham (2017)
25. Wixon, D., Holtzblatt, K., Knox, S.: Contextual design: an emergent view of system design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 329–336. Citeseer (1990)