

The Agile UX Development Lifecycle: Combining Formative Usability and Agile Methods

Suzanne Kieffer, Aissa Ghouti, Benoit Macq
Université catholique de Louvain – ICTEAM/ELEN
Place du Levant, 2 – BE-1348 Louvain-la-Neuve, Belgium
{suzanne.kieffer, aissa.ghouti, benoit.macq}@uclouvain.be

Abstract

This paper contributes a method variation that helps cross-functional teams combine both formative usability and agile methods to develop interactive systems. Both methods are iterative, continuous and focus on delivering value to users, which makes their combination possible. The “agile UX development lifecycle” supports and facilitates the synchronization of the steps involved in both formative usability and agile sprints in an operable manner and is intended for design and development settings. We present a case study that illustrates the extent to which this tool meets the needs of real-world cross-functional teams, describing the gains in efficiency it can provide but also guidelines for increasing the benefits gained from this combination in design and development settings.

1. Introduction

Formative and summative usability testing are product evaluation methods broadly adopted in User-Centered Design (UCD). Formative usability is an iterative test-and-refine method applied early in the design process, which aims at detecting and fixing usability problems. On the other hand, summative usability is a singulative quality insurance method applied later in the design process, which aims at comparing the User Experience (UX) with a product against a set of UX goals or other products. Formative usability supports decision making during product design and development, whereas summative usability is a tool for describing the UX.

Agile is a widely accepted and adopted approach to software development in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. In Agile Software Development (ASD), iteration is a set period of time during which specific work (development) has to be completed and made ready for review (testing).

In this paper we explore the potential benefits that lie in combining formative usability and agile methods. We present the *agile UX development lifecycle*, a tool intended for development settings in which cross-functional teams with both UX and agile expertise wish to integrate UCD and ASD methods.

In this paper, the report of a case study illustrates how iterations (or *sprints* in the Scrum framework) were organized and implemented to develop a mobile application for the empowerment of patients with diabetes condition. In particular, guidelines for early sprints are presented to help teams focus on relevant UX metrics, namely user errors (sprint 0), user performances (sprint 1), and satisfaction (sprint 2).

The contribution of this work lies in that it identifies discrepancies in the literature regarding the integration of UCD and agile methods and sketches an approach by which the challenges related to their integration can be overcome. Second, the paper provides a first, realistic and successful account of the application of the *agile UX development lifecycle*. The reflection on a case study allows us to propose the approach as beneficial to software development and worthy of further application.

The paper is structured as follows: the next section presents how the integration of formative usability and agile methods can help improve software development by synchronizing UX and agile activities to meet UX goals while complying with agile principles. The agile UX development lifecycle is then presented. This is followed by the presentation of a case study conducted in the *mHealth* sector –mHealth being an abbreviation of mobile health and standing for the practice of medicine and public health supported by mobile devices. The questions raised during the case study such as “how is teamwork organized?”, “how is formative usability implemented?” and “why did the agile UX approach facilitate the integration of agile and UX methods?” are then discussed before concluding.

Table 1. AUCDI/UCASD related work summary

Reference	[3]	[7]	[12]	[16]	[23]	[28]
Year of publication	2014	2015	2015	2013	2014	2007
Systematic Literature Review		x			x	
Model for AUCDI/UCASD			x	x		x
Guidelines				x		x
Patterns, principles, or practice success factors	x	x			x	
Examples or case study	x			x		x
Cumulative empirical validation		x			x	
Scrum	x	x	x		x	x
Formative usability						x

2. Related work

Over the last decade, Agile User-Centered Design Integration (AUCDI) [23], also referred to as User-Centered Agile Software Development (UCASD) [7], has been extensively studied due to the complementary of the approaches and the challenges associated with their combination. The current scientific consensus on AUCDI principles is discussed in two independent, recent Systematic Literature Review (SLR) studies [7, 23]. It can be summarized as follows: design and development activities should be iterative and incremental, organized in parallel interwoven tracks, and continuously involve users.

On the other hand, Brhel et al. [7] also highlight methodological discrepancies: the first half (50.6%) of the publications included in their SLR discuss up-front analysis efforts and argue for *Little Design Up-Front* (LDUF) during which user research activities or low-fidelity prototyping are carried out. Especially, the LDUF concept is implemented by reserving a *Sprint Zero* before actual agile development iterations start in order to "define a broader view of the product, general goals, to roughly plan the next sprints and to define design principles" [4]. The principle of *Sprint Zero* is usually associated with that of *One Sprint Ahead* [3, 12, 18] according to which the UX team works at least one iteration ahead of development.

The second half of the papers examined in [7] does not mention any such up-front planning activities, most likely because they go against one of the fundamental principles of agile methods: "to remain responsive to changing requirements" [2, 27], despite the expensive late design changes that this may cause (the ratio of early-late design changes follows a rule of thumb and ranges from 1:4 to 1:100 [5]). There are, at least to the extent of our knowledge, no acknowledged procedures or practical guidelines regarding the effort to put into initial up-front analysis and design activities, and uncertainty remains regarding their duration [7].

Furthermore, most of the proposed AUCDI models lack cumulative empirical validation and need further

investigation before guidance can be provided for their successful implementation (Table 1).

This is especially the case of the UCD process for Scrum proposed in [12] in which no validation is reported. To a lesser extent, this is also the case of [16] and [28] in which a case study is reported as proof of concept. A model-driven methodology for the management of both functional and user requirements and the distribution in iterations of developmental and integration activities is proposed in [16]. However, only coarse and vague information for the integration of usability evaluation processes is provided.

A model to conduct usability tests, interviews, and contextual inquiry within an agile framework is proposed in [28]. To allow the UX team to iterate on designs, rapid formative usability testing is carried out at least one cycle ahead of developers (cycle 0), and the validated designs are then passed on to be implemented (cycle 1). Contextual inquiry is also conducted at least two cycles ahead (cycle 0), and usability testing is then conducted to check for design drift (cycle 2). This paper is the only one discussing the benefits of formative usability for design validation purposes in an agile approach (Table 1).

The agile UX development lifecycle is a method variation that builds on the Scrum framework [24, 27] and the following guidelines (G1 to G5):

- G1: UX and agile activities should be iterative, incremental, organized in parallel synchronized tracks and continuously involve users [7, 23]
- G2: Time should be allocated for up-front activities so as to achieve a shared product vision and elicit user and functional requirements [23]
- G3: Design should be chunked into Features so as to incrementally build the product prototype [23]
- G4: Rapid formative usability should be carried out so as to fulfill UX goals [28] while delivering working software frequently [2]
- G5: Up-front analysis, design and usability findings should be documented so as to avoid confusion in regards to UX deliverable [23].

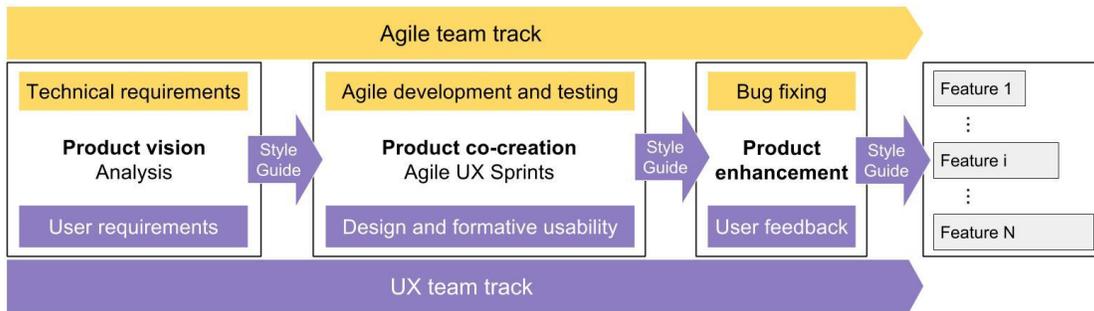


Figure 1. Agile UX development lifecycle

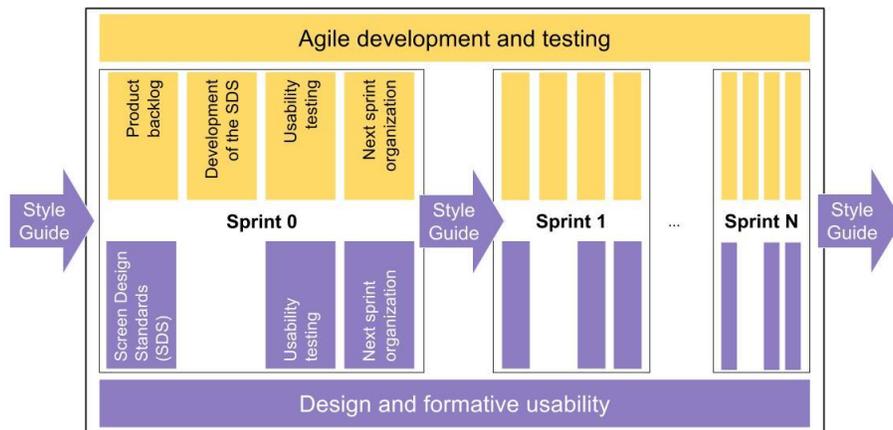


Figure 2. Agile UX sprint (product co-creation phase)

3. The agile UX development lifecycle

The agile UX development lifecycle is presented in Figure 1. It is organized in two tracks (G1): agile team track in yellow; UX team track in purple). It involves three phases (G1): product vision, product co-creation and product enhancement. *Features* are incremental versions of the product prototype released iteratively after each agile UX development cycle (G3): Feature 1 corresponds to a *minimum vital product* [14] whereas Feature N corresponds to the final product.

During product vision, the UX team conducts user requirements activities (G2) such as task analysis and user profile. Results are reported in a *Style Guide* (G5). *Style Guide* improves dynamics between both teams by allowing them to share a common understanding of user goals and design vision. Meanwhile, the agile team can focus on high-development low-UX cost activities: the development of generic back-end and front-end modules.

During product co-creation, efforts of both teams are synchronized in *agile UX sprints* (Figure 2). Agile UX sprints involve four steps. Product backlog and Screen Design Standards (SDS) are co-created at step 1, the first milestone that involves trade-offs between

development time and costs and UX design refinement. The agile team develops the SDS at step 2. Usability testing is conducted collaboratively at step 3 (G4): UX team members act as experimenters and agile team members as observers. Usability findings are reported by the UX team in *Style Guide* (G5), before deciding what design changes are necessary in the next sprint (step 4). Agile UX sprints are quickly reiterated until the release each feature (G4).

During product enhancement, the agile team focusses exclusively on fixing remaining bugs while the UX team analyzes user feedback which directly feeds into the user requirements of the next feature.

As can be seen from Figures 1 and 2, the *Style Guide* is an elementary interface between both teams as it documents the intended user interface standards for designers and developers to be followed during design and development. While functional spec documents the way things should be done, *Style Guide* presents a user-driven perspective and analysis of user profile, real-life instances of tasks, usability findings (issues and redesign solutions), and UX. A template can be found in [17]. This focus is core to meet user requirements and ensures that the product supports real-life tasks in real world environment.

Table 2. Types of user error [30]

Code	Type of error
E1	Behaviors that prevent task completion or take someone “off course”
E2	Mistaken believes that a task is completed when it is not (and vice versa)
E3	Misinterpretation of some piece of content
E4	Oversight of something that should be noticed
E5	Expression of frustration by the participant
E6	Remarque about “room for improvement” from the participant

Table 3. The 3-level severity scale [30]

Severity	Type of usability issue
Low (lo)	Any issue that annoys or frustrates participants but does not play a role in task failure. These are the types of issues that may lead users off course, but they still recover and complete the task. This issue may only reduce efficiency and/ or satisfaction a small amount, if any.
Medium (me)	Any issue that contributes to significant task difficulty but does not cause task failure. Participants often develop workarounds to get to what they need. These issues have an impact on efficiency and satisfaction.
High (hi)	Any issue that leads directly to task failure. Basically, there is no way to encounter this issue and still complete the task. This type of issue has a significant impact on effectiveness, efficiency, and satisfaction.

4. Case study

4.1. Approach

A multiple-case approach was adopted to study how and why the agile UX lifecycle and agile UX sprints in particular can facilitate AUCDI? Such a research strategy was favored as it is relevant when "how" and "why" questions are being posed, when the investigator has little control over events, and when the focus is on a contemporary phenomenon with some real-life context [31].

4.2. Research plan

The research plan involved the adoption of the agile UX lifecycle as development model, and the implementation of agile UX sprints to release a product prototype in a research project. Specifically, three agile UX sprints were run to release Feature 1 (sprints 0-2), Feature 2 (sprints 3-5) and Feature 3 (sprints 6-8). In this context, a "case" refers to what happened during each Feature, also referred to as agile UX development cycle or iteration.

4.3. Units of analysis

Formative usability is the core component of the agile UX sprint. Therefore, the units of analysis were selected to detect and fix usability problems and to support decision making during product design and development [30]. While earlier sprints (Feature 1) focused on issue-based metrics, performance metrics and self-reported metrics, later sprints (Features 2 and 3) focused on users' subjective feedback.

User errors (sprint 0) were used to analyze usability issues (Table 2). Severity rates were assigned to usability issues in order to help focus attention on the issues that really matter. A 3-level severity scale was chosen (Table 3).

Task success and task time (sprints 1 and 2) were used as measures of user performance. Task success measures how effectively users are able to complete a given set of tasks and was encoded as a binary score: 1 for success (i.e. successful task completion) *versus* 0 for failure (i.e. users requested assistance, gave up or thought the task was completed but it was not). Task time measures how much time is required to complete a task and was encoded as duration.

Self-reported user satisfaction (sprints 1 and 2) was assessed using the Questionnaire for User Interface Satisfaction (QUIS) [9]. The QUIS consists of 27 items distributed between five categories: overall reaction, screen, terminology, learning and system capabilities. The ratings are on 10-point scales whose anchors change depending on the statement. Polar opposites (e.g., difficult/easy) with no statements are used for the first category.

These units of analysis could have been enriched by other UX metrics such as efficiency, behavioral metrics, combined and comparative metrics [30].

4.4. Research project

The project aimed to develop a mobile application for the *empowerment* of patients with diabetes condition. The empowerment aims to help patients discover and develop the inherent capacity to be responsible for their own life [6]. Diabetes is a medical condition with the body that causes blood glucose (sugar) levels to rise higher than normal [1]. This is also called hyperglycemia.

Table 4. User involvement (sprints 0-8)

Feature	Sprint	P1	P2	P3	P4
1	0	x			
	1		x	x	
	2	x			
2	3	x			
	4		x		
	5			x	
3	6	x		x	x
	7				x
	8				x

Table 5. User errors made by P1 at sprint 0

Task	T1	T2	T3	T4	T5	T6	T7	T8
E1		x						
E2				x	x			
E3				x	x			
E4		x		x	x			
E5				x	x			
E6								
Severity		lo		hi				

4.5. Participants

Formative usability consists in iterative frequent lab-tests with a small number of participants [20]. All participants were diagnosed with Type 1 diabetes, i.e. when the body does not produce insulin [1]. Initially, three adults (P1; P2; P3) with diabetes condition were recruited from the patient base of a consortium partner. One teenager (P4) was recruited at sprint 6 for preliminary validation of the design before remote usability evaluation. The involvement of users is depicted in Table 4. P2 dropped out after the second session because of severe health issues.

The sample participants involved three standard and one advanced user of computer systems. P1, P2 and P3 were standard users of smartphones (i.e. assistance was needed for advanced functionalities) whereas P4 was an advanced user of smartphones (i.e. no assistance was needed). They all considered computers as important in their lives and all liked learning how to use a new computer system.

4.6. Formative usability testing

Formative usability usually involves one-on-one sessions between an experimenter and a participant [30]. Participants were required to perform a set of tasks and to answer questions about their UX with the prototype. The participants were likely to be thinking aloud as they performed the tasks, while the experimenter recorded their behavior and answers.

Experimental tasks were used in order to create goals to be achieved by the participants, and thus intentions that translate into action sequence (see Norman's Action Cycle [21]). This allowed us to collect and analyze factual user data. Eight tasks (T1 to T8) were identified during task analysis and used at sprint 0 in Feature 1:

- T1: create account
- T2: log in
- T3: change password
- T4: enter current weight
- T5: enter weight objective
- T6: find specific item in "tips and tricks"
- T7: save it as favorite
- T8: enter specific appointment in agenda

4.7. Sprint 0: focus on user errors

User errors were analyzed to detect usability issues. Table 5 presents the frequency distribution of errors made at sprint 0 by participant P1 per task with type of errors and severity rates (lo, me and hi).

Two usability issues were detected (Table 5). Firstly, P1 made a skill-based error, referred to as "slip" in [11], during T2, assuming the login procedure involved entering an email address, while it actually involved entering the username defined during T1. As P1 could easily recover from it, this issue was assigned a low severity rate and fixed by complying with most of today's connection standards in which the login is the email address.

Secondly, P1 made a mistake while executing T4 and T5 (i.e., entering values). P1 actually believed that an "objective" button stood for a static label and that the "+" button was the interactive component associated with it (E3). Consequently, P1 overlooked the "objective" button (E4), thought that T5 had already successfully been completed (E2) and was frustrated when told it had not (E5). As there was no way that P1 could encounter this issue and complete the task, it was assigned a high severity rate.

This usability issue was actually caused by the lack of consistency in the appearance of the two adjacent buttons. A label ("objective") was used to enter an objective whereas a symbol ("+") was used to add a value. In addition, the buttons had different background colors: white for "objective" and green for "+" (Figure 3a). Considering the high severity of this usability issue, the team decided to make design changes and test them at sprint 1 with P2 and P3. The design changes consisted in creating two buttons, both with the same background color, and labelled "objective" and "new entry" respectively (Figure 3b).



(a) Feature 1, Sprint 0 (P1) (b) Feature 1, Sprint 1 (P2 and P3) (c) Feature 3 (P1, P3 and P4)
Figure 3. Incremental prototyping of the glycaemia widget

Table 6. Task success per participant for task T1 to T8

Feature	Sprint	Participant	T1	T2	T3	T4	T5	T6	T7	T8	Score
1	0	P1	1	1	1	0	0	1	1	1	.75
1	1	P2	0	1	1	1	1	1	1	1	.87
1	1	P3	0	1	1	1	1	1	1	1	.87
3	6	P4	1	1	1	1	1	1	1	1	1

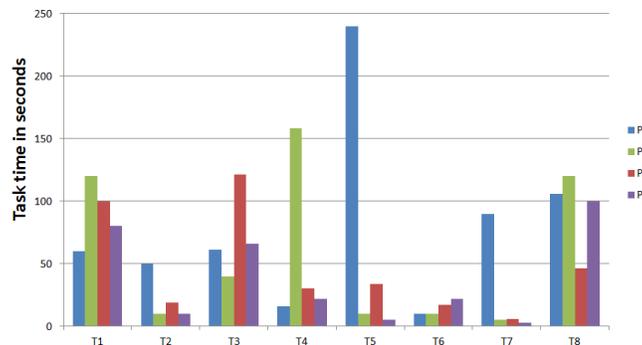


Figure 4. Task time per participant for the experimental tasks T1 to T8

4.8. Sprints 0-1: focus on user performances

User performances of P1 were compared with those of P2 and P3 to evaluate whether or not the design changes resulting from sprint 0 had improved the UX at sprint 1. Task success and task time are depicted in Table 6 and Figure 4 respectively.

No errors were made by P2, P3 and P4 while executing T4 and T5 (Table 6). This suggests that the related usability issue was fixed. This result is further consolidated by the improvement of task time (Figure 4) between P1 and the other participants, especially during T5. P2 explored the user interface during T4 which explains the longer task time observed during T4. Similarly, task times improved between P1 and other participants during T2, which suggests that the related usability issue was fixed.

However, another usability issue was identified at sprint 1. It was assigned a high-severity rate because

it led to task failure (Table 6). Both P2 and P3 overlooked the "create an account" button and mistook it with the "log in" button. This usability issue was fixed by simplifying the layout of the landing page. No such errors were made during any of the later sprints.

4.9. Sprints 1-2: focus on user satisfaction

The user satisfaction was assessed at sprints 1 and 2 using the QUIS [9]. This material was chosen as it uses polar opposites such as difficult/easy or frustrating/satisfying, which engages participants in elaborating their answers. The QUIS was preferred to Lewis's CSUQ [15] or Brooke's SUS [8] which are better suited for later development stages, that is to say when a more completed product prototype is available for testing. The user satisfaction increased between sprints 1 and 2 (Table 7).

Table 7. QUIS results

Sprint	1		2
	P2	P3	P1
Mean	6.81	7.04	7.75
Median	7	7	8
Min	2	2	6
Max	8	9	9
Range	6	7	3
St Dev.	1.55	2.05	0.90
Overall (1-6)	6.50	7.33	7.00
Screen (7-10)	7.25	6.00	7.75
Terminology (11-16)	6.20	6.33	8.25
Learning (17-22)	7.50	7.00	8.00
Functionalities (23-27)	6.60	8.40	8.00

As can be seen from Table 7, both the mean and the median satisfaction scores increased. What is particularly striking is the difference between min scores (2 at sprint 1 vs. 6 at sprint 2), ranges (superior or equal to 6 at sprint 1 vs. 3 at sprint 2) and standard deviation (superior or equal to 1.55 at sprint 1 vs. 0.90 at sprint 2). These differences seem to mostly result from the improvements in both terminology (+1.98 points) and screen (+1.13 points). The findings from the previous sprints 1 and 2 directly led to redesign choices which directly fed into the incremental product prototype.

4.10. Releases

The product prototype evolved as a sequence of features. Feature 1 consisted of the set of features necessary to a *minimum vital product* prototype [14]: i.e. a customizable dashboard and widgets for the monitoring of glycaemia, insulin intake and weight. The sprints necessary to release Feature 1 focused on the navigation between and the use of widgets, ease of learning, and the compliance with diabetes-related standards (terminology, units and key tasks).

New widgets were added in Feature 2: i.e. the monitoring of meals, social and physical activities, and moving habits. The sprints necessary to release Feature 2 focused on the improvement of Screen Design Standards (SDS). Several type of graphs (plot charts, bar charts and time series) and interaction means (tabs, scales and icons) were tested so as to support and facilitate the visualization of information in large datasets and the data entry respectively.

Feature 3 embedded a to-do list. The sprints necessary to release Feature 3 focused on simplifying the SDS (Figure 3) mainly by maximizing the data-ink ratio [29], further improving the interaction with graphs and gaining insight about the functionalities to be added to complete the product.

Nine sprints (0-8) were needed to produce the first version of the product (i.e. Feature 3). Each sprint lasted between two and four weeks depending on participant availabilities. The prototype is currently deployed for testing purposes within a group of ten patients including three adults and seven teenagers. User feedback is collected with Usersnap (<http://usersnap.com/>), a tool that allows users to send screenshot-based feedbacks in three steps: (1) take a screenshot; (2) comment; (3) send.

This tool was chosen for the following reasons. Firstly, the 3-step procedure is simple and intuitive. Thus, neither training nor extended effort is required from users. Secondly, feedbacks are well structured, and therefore easy to manage. Thirdly, feedbacks are produced as problems occur, which provides the agile UX team with concrete, meaningful and real time information about user experience. Finally, Usersnap offered the best quality-value compared to similar products available on the market.

5. Discussion

5.1. How was teamwork organized?

The agile UX team was composed of a UX expert, a project manager and a developer whose respective background and assignments are presented in Table 8. The UX expert led the UX track whereas the project manager led the agile track.

The UX expert conducted user research activities (task analysis, user profile, and card-sorting) and reported the findings in the Style Guide during the product vision phase. It also remained responsible for the production of SDS and the implementation of UX evaluations during agile UX sprints. Especially, the SDS were sketched by the UX expert, handed to the project manager, and trade-offs between development time and costs, platform capabilities and UX were made if necessary. The experimental tasks involved at sprints 0, 1 and 6 were selected by the UX expert so as to engage first-time participants in real-life scenarios. So were the units of analysis as this requires UX expertise.

Due to the limitation of the available resources, the project manager had to take on three roles of Scrum: product owner, scrum master and developer [24]. Specifically, the project manager served as an interface between the agile UX team and the users, and was responsible for the prioritization of design changes after UX evaluations (product owner). The project manager also coached the development team, made sure both the Scrum rules and the Style Guide were followed (scrum master).

Table 8. Agile UX team

Function	Background	Specific assignments in the agile UX development lifecycle
UX expert	Ph.D. in Human-Computer Interaction 10 years experience in UX research Skills: UCD, UE, UX	UX activities: user profile, contextual task analysis, usability/UX goals setting, platform capabilities and constraints, Screen Design Standards, formative usability, Style Guide
Project manager	MS degree in Computer Science 3 years experience in development Skills: JavaScript, Node.js, HTML5, CSS3, Unix, Scrum framework (certification)	Agile project management with Scrum framework Scrum roles: product owner, scrum master and development
Developer	BS degree in Computer Science 7 years experience in development Skills: JavaScript, Node.js, HTML5, Unix	Scrum role: development

The accumulation of roles to assume was made possible for the project manager because of the small size of the team which allowed one-to-one meetings with the different stakeholders of the project.

5.2. How did formative usability go?

Formative usability testing was led by the UX team and involved individual sessions with the participants. The project manager took part in all the usability testing sessions as observer.

Once the product prototype stabilized (i.e. starting from Feature 3), live editing was introduced into the sessions. This enabled us to quickly fix bugs and to engage participants in live discussions about the subjective quality of the user interface and the efficiency of data visualizations. In particular, P1, P3 and P4 suggested changes regarding the content (missing or incomplete information), the layout of interactive components (color, size, shape and alignment) and the type of graphs that were used (time-series, scatter graphs and bar charts). When they could quickly be implemented, these suggestions were live edited and informally discussed right away during the session. They were formally discussed by the agile UX team later on during the session debriefing: if judged relevant, they were selected as redesign changes to be implemented in the next sprint.

It should be noted that this was strictly avoided during earlier sprints (i.e. Features 1 and 2, sprints 0-5) as all the usability issues had not been fixed yet. Furthermore, changes made this early with only one participant would have been completely dependent on the problems this particular user struggled with.

5.3. Why did the agile UX approach facilitate AUCDI?

What facilitated AUCDI is the complementarity of formative usability and agile methods: one makes up for the shortcomings of the other.

From an agile perspective, the concern about UX methods is that they are resource intensive and delay product release. Textbooks for best practice in UX recommend creating and testing prototypes before product development [14, 17, 25, 30, 31] which is resource intensive and may spread over weeks, whereas the Agile Manifesto [2] recommends early, continuous and frequent delivery of valuable software: from a couple of weeks to a couple of months, with a preference to the shorter timescale.

From a UX perspective, the concern about agile methods is their lack of user requirements analysis which increases the risk of late design changes and therefore development time and costs [5]. The Agile Manifesto [2] does not recommend up-front analysis and welcomes changing requirements, whereas UX methods recommend reducing the risks of late design changes by maintaining users continuously involved and basing design solutions on user data [5].

The agile UX approach is intended to facilitate the trade-offs to be made by development teams concerned with AUCDI. The risks of expensive late design changes are reduced by the user requirements analysis. Formative usability allows maintaining user interest in the product, the continuous delivery of working software at reasonable speed, and facilitates the collection of user feedback.

5.4. How important was the Style Guide?

The Style Guide included the following sections: glossary, domain-related information (i.e. diabetes), user categories and user profile, task analysis, design implications, UX goals, platform capabilities and constraints, input/output modalities, screen design standards, findings from sprints 0-8, and final recommendations to minimize user errors.

Style Guide was used as communication artefact by both teams. While the UX team documented the Style Guide, the agile team used it as a guide to develop the intended user interface.

6. Lessons learned

6.1. Staff key positions adequately

Key positions such as project manager and UX expert should be adequately staffed. The importance of staffing key roles with skilled people has been acknowledged in both the UX and the agile community. See for example Bias's dangers of amateur UE [4] and Meyer's classification of agile ideas between brilliant (the good), touted (the hype) or poor (the ugly) [19]. This recommendation holds when adopting the agile UX development approach as its success relies on the expertise of the team members to efficiently carry out the relevant agile and UX activities.

Executive levels in the organization should refrain from hiring one single person to lead both the agile and the UX tracks as this would severely impede the efficiency of the approach. Both the agile and UX activities require high-level not only to increase the usability of products, but also to adequately feature its impact on organizational efficiencies, reduce uncertainties about schedule and budgets, and therefore reduce the risks of failure [1, 2, 13].

6.2. Involve actual users

Consortium partners or developers should not be substituted for actual patients. The involvement of real users is necessary to avoid biases in analysis, design and testing activities. This basic principle of UCD is unanimously acknowledged and adopted by the Human-Computer Interaction community.

The main barrier to overcome in the case study was to access users. This is a common problem in UX practice which was intensified in this case because of the medical context of the project. The participants to formative usability were recruited from the patient base of the consortium medical expert, Dr. in diabetology at a university clinic. The recruitment of participants took about six months in total. First, the experimental design of formative usability evaluation had to be validated by the Ethics Committee of the clinic (four months). Then, the medical expert solicited potential participants in person (two months).

To speed up the process of user recruitment, the medical expert should have been involved earlier in the project. In order to maintained his motivation and reinforced his relationship with the agile UX team, the medical expert should have been assigned a more rewarding role in the project, e.g. user representative.

6.3. Choose a development environment that supports fast prototyping and multi-platform deployment

The development environment involved in the case study integrated Bootstrap [13] and MEAN [18], which supports fast prototyping and multi-platform deployment.

7. Conclusion

This paper identifies a gap in the current research literature regarding the tools available to support the integration of formative usability and agile methods in development settings. It argued that use can be made of the agile UX approach to improve organizational efficiencies, provided that the key roles, UX expert and agile expert, are adequately staffed. A number of guidelines for the practical use in an industrial context of this tool were mentioned, primarily to align UX metrics with the state of the product prototype. A case study has been presented as proof of concept of mHealth project engineering with the agile UX development approach.

We have proposed a method which involves the synchronization of UX and agile tasks within agile UX sprints, the identification of relevant UX metrics to be considered, and the improvement of product quality through iterative evaluations.

The novelty of this work lies in the fact that agile UX releases are used as product prototype during usability testing. This allows collecting user feedback with a real product user interface while complying with the agile principle of frequent releases. Formative usability is usually conducted differently when combined with ASD: solely by the UX team, using low-fidelity prototypes [25], and before designs are passed on to the agile team for implementation [10, 28]. Alternatively, it is limited to acceptance or demonstration sessions [23].

Future research should provide more detailed methodological guidance for agile UX procedures. Further case studies are needed to justify the generalizability of the approach in different contexts of use.

8. Acknowledgments

The authors warmly thank the participants to the study. The authors acknowledge the support of this research by the project HealthCompass under the reference RBC/13-PFS-EH-2 funded by Innoviris, Belgium.

9. References

- [1] American Diabetes Association. "Diabetes Basics": n.d. <http://www.diabetes.org/diabetes-basics/?loc=db-slabnav>. Accessed: Sep. 7, 16.
- [2] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. "Manifesto for Agile Software Development": 2001. www.agilemanifesto.org/. Accessed: Sep. 7, 16.
- [3] Bertholdo, A.P., da Silva, T.S., Melo, C.D., Kon, F., and Silveira, M.S. "Agile usability patterns for UCD early stages", Design, User Experience, and Usability: Theories, Methods, and Tools for Designing the User Experience, Springer International Publishing, 2014, pp. 33–44.
- [4] Bias, R. G. "The dangers of amateur usability engineering", in Usability in Practice: Avoiding Pitfalls and Seizing Opportunities, S. Hirsh (Chair). Presented at ASIST 2003, Long Beach, CA, October 20–23, 2003.
- [5] Bias, R.G. and Mayhew, D.J. Cost-Justifying Usability, Second Edition: An Update for the Internet Age. Morgan Kaufmann, 2005.
- [6] Bravo, P., Edwards, A., Barr, P.J., Scholl, I., Elwyn, G. and McAllister, M. "Conceptualising patient empowerment: a mixed methods study," BMC health services research, vol. 15, no. 1, p. 1, 2015.
- [7] Brhel, M., Meth, H., and Maedcher, A. "Exploring principles of user-centered agile software development: A literature review", Information and Software Technology, 61(C), 2015, pp.163–81.
- [8] Brooke, J. "SUS-A quick and dirty usability scale", Usability evaluation in industry 189.194, 1996, pp. 4–7.
- [9] Chin, J. P., Diehl, V. A., and Norman, K. L. "Development of an instrument measuring user satisfaction of the human-computer interface", in Proc. CHI'88, 1988, pp. 213–218.
- [10] da Silva, T.S., Selbach Silveira, M, and Maurer F. "Usability Evaluation Practices within Agile Development", in Proc. HICSS-48, 2015, pp. 5133-5142.
- [11] Embrey, D. "Understanding Human Behavior and Error", Human Reliability Associates, Wigan, Lancashire, WN8 7RP, 1996.
- [12] Forbrig, P. and Herczeg, M. "Managing the Agile Process of Human-Centred Design and Software Development", in Proc. INTERACT'15, September 14–18, 2015, Bamberg, Germany, 2015.
- [13] "Getbootstrap.com the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web": 2011. Accessed: Sep. 7, 16.
- [14] Levy, J. UX Strategy: How to Devise Innovative Digital Products That People Want, O'Reilly Media, 2015.
- [15] Lewis, J.R. "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use", International Journal of Human-Computer Interaction, 7(1), 1995, pp. 57–78.
- [16] Losada, B., Urretavizcaya, M., and Fernández-Castro, I. "A Guide to Agile Development of Interactive Software with a 'User Objectives'-driven methodology", Science of Computer Programming, 78(11), 2013, pp. 2268–2281.
- [17] Mayhew, D.J. The Usability Engineering Lifecycle : A Practitioner's Handbook for User Interface Design (1st edition), Morgan Kaufmann, San Francisco, 1999.
- [18] "Mean.io: The Friendly & Fun Javascript Fullstack for your next web application", 2014.
- [19] Meyer, B. Agile!: The Good, the Hype and the Ugly. Springer Publishing Company, Incorporated, 2014.
- [20] Nielsen, J. and Landauer, T.K. "A Mathematical Model of the Finding of Usability Problems", in Proc. INTERCHI'93, April 24–29 1993, Amsterdam, 1993, pp. 206–213.
- [21] Norman, D.A. "Psychology of Everyday Action". The Design of Everyday Things. New York: Basic Book, 1988.
- [22] Rosenbaum, S., Rohn, J.A., and Humburg, J. "A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys", in Proc. CHI'00, April 1–6, 2000, The Hague, Netherlands, 2000, pp. 337–344.
- [23] Salah, D., Paige, R.F., and Cairns, P. "A systematic literature review for agile development processes and user centred design integration", in Proc. EASE'14, 2014.
- [24] "Scrumalliance.org", 2016. Accessed: Sep. 7, 16.
- [25] Sefelin, R., Tscheligi, M., and Giller, V. "Paper Prototyping: What is it good for? A Comparison of Paper- and Computer-based Low-fidelity Prototyping", in Proc. CHI'03, Ft. Lauderdale, Florida, USA, April 5–10, 2003.
- [26] Seffah, A. and Metzker, E. "The Obstacles and Myths of Usability and Software Engineering", Communications of the ACM, 47(12), 2004, pp. 71–76.
- [27] Sutherland, J. Harrison, N. and Riddle, J. "Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams", in Proc. HICSS-47, 2014, pp. 4722-4728.
- [28] Sy. D. "Adapting Usability Investigations for Agile User-centered Design", Journal of Usability Studies 2(3), 2007, pp. 112–132.
- [29] Tufte, E. The visual display of quantitative information, 2nd edition. Graphics Press, Cheshire, Connecticut. 2001.
- [30] Tullis, T. and Albert, W. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics, Morgan Kaufmann, San Francisco, 2013.
- [31] Vredenburg, K., Isensee, S., and Righi, C. "How to Introduce, Deploy, and Optimize User-Centered Design in Your Organization", Tutorial Notes CHI'02, April 20–25, 2002, Minneapolis, Minnesota, USA, 2002.
- [32] Yin, Yin, R. K. "Case study research: Design and method." (1998).