JOURNAL OF NURSING SCHOLARSHIP

SPECIAL ISSUE

Implementation Science Meets Software Development to Create eHealth Components for an Integrated Care Model for Allogeneic Stem Cell Transplantation Facilitated by eHealth: The SMILe Study as an Example

Lynn Leppla, RN, MScN[1], Sandra Hobelsberger, MA[2], Dennis Rockstein, MSc[3], Viktor Werlitz, BSc[4], Stefan Pschenitza, BSc[5], Phillip Heidegger, Dipl-Inf, PhD[6], Sabina De Geest, RN, PhD, FAAN, FRCN, FEANS[7], Sabine Valenta, RN, PhD[8], Alexandra Teynor, Dipl-Inf, PhD[9,*], & the SMILe study team

1 Research Assistant, Institute of Nursing Science, Department Public Health, University of Basel, Switzerland & Advanced Practice Nurse, Department of Medicine I, Medical Center, Faculty of Medicine, University of Freiburg, Germany
2 Research Assistant, Designer, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany
3 Research Assistant, Software Engineer, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany
4 Research Assistant, Software Engineer, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany
5 Research Assistant, Software Engineer, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany
6 Professor of Software Engineering, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany
7 Professor of Nursing, Director and Department Chair Institute of Nursing Science, Department Public Health, University of Basel, Switzerland & Professor of Nursing, Academic Centre for Nursing and Midwifery, Department of Public Health and Primary Care, KU Leuven, Belgium
8 Postdoctoral Research Fellow, Institute of Nursing Science, Department Public Health, University of Basel, Switzerland & Advanced Practice Nurse, Department of Hematology, University Hospital Basel, Switzerland
9 Professor of Software Engineering, Faculty of Computer Science, Augsburg University of Applied Sciences, Germany

Key words
Agile software development, eHealth, implementation science, integrated care model, user-centered design

Correspondence
Dr. Alexandra Teynor, Department of Computer Science, Augsburg University of Applied Sciences, An der Hochschule 1, 86161 Augsburg, Germany.
Email: alexandra.teynor@hs-augsburg.de

Accepted November 30, 2020

doi:10.1111/jnu.12621

Abstract

Purpose: To describe a process of creating eHealth components for an integrated care model using an agile software development approach, user-centered design and, via the Behavior Change Wheel, behavior theory-guided content development. Following the principles of implementation science and using the SMILe project (integrated care model for allogeneic stem cell transplantation facilitated by eHealth) as an example, this study demonstrates how to narrow the research-to-practice gap often encountered in eHealth projects.
Methods: We followed a four-step process: (a) formation of an interdisciplinary team; (b) a contextual analysis to drive the development process via behavioral theory; (c) transfer of content to software following agile software development principles; and (d) frequent stakeholder and end user involvement following user-centered design principles.
Findings: Our newly developed comprehensive development approach allowed us to create a running eHealth component and embed it in an integrated care model. An interdisciplinary team's collaboration at specified interaction points supported clear, timely communication and interactions between the specialists. Because behavioral theory drove the content development process, we formulated user stories to define the software features, which were prioritized and iteratively developed using agile software development principles. A prototype intervention module has now been developed and received high ratings on the System Usability Scale after two rounds of usability testing.
Conclusions: Following an agile software development process, structured collaboration between nursing scientists and software specialists allowed our interdisciplinary team to develop meaningful, theory-based eHealth components adapted to context-specific needs.

Clinical Relevance: The creation of high-quality, accurately fitting eHealth components specifically to be embedded in integrated care models should increase the chances of uptake, adoption, and sustainable implementation in clinical practice.

on behalf of the SMILe study team (See Acknowledgments for list of team members)

eHealth has dramatically influenced the possibilities of healthcare services in recent years and is soon expected to become a major driver for improvement (World Health Organization, 2020). Many European countries promote innovation in care delivery for chronically ill persons, whose burgeoning numbers challenge healthcare systems in view of their volume, the complexity of their needs, and their exploding care costs (Melchiorre et al., 2020). International policy agendas commonly include calls for eHealth-facilitated care delivery.

Evidence supports eHealth's potential to fuel innovation. In particular, studies testing eHealth applications embedded within integrated care models (ICMs) for chronically ill persons have shown clear links to improved clinical, behavioral, and economic outcomes (Aapro et al., 2020; Barello et al., 2016; Elbert et al., 2014; Kuijpers, Groen, Aaronson, & van Harten, 2013; Michaud, Zhou, McCarthy, Siahpush, & Su, 2018; Warrington et al., 2019).

eHealth-facilitated ICMs are also promising as systems to strengthen the care of allogeneic stem cell transplantation (alloSCT) recipients. With a 70% to 90% risk for developing complications and late effects, these patients are chronically ill and require comprehensive care (Majhail, 2017). Using eHealth solutions as part of ICMs would allow improved support of alloSCT patients, for example, by detecting health deterioration early, even after discharge.

However, despite increasing empirical evidence supporting eHealth integration into care delivery, sustained implementations of eHealth solutions in routine clinical practice are rare: uptake is often insufficient, adoption rates low, and sustained use rates lower still (Bates & Wright, 2009; Elbert et al., 2014; Furlong et al., 2019; Widmer et al., 2015). Studies testing ready-made eHealth tools commonly report serious adoption and sustainment problems, with 44% to 67% of patients prematurely discontinuing use (Jeffs et al., 2016; Simblett et al., 2018; Thies, Anderson, & Cramer, 2017). Also, clinicians consider a misfit with their existing clinical workflow one of the most frequent barriers for adoption to clinical practice (Granja, Janssen, & Johansen, 2018). Most of these failures can be traced either to a poor application fit—meaning both with end users' (patients and clinicians) needs and with the target context—or to a lack of attention to the content's theoretical foundations (Jeffs et al., 2016; Simblett et al., 2018; Thies et al., 2017). Indeed, of the 23 reviewed eHealth tools for patients with cancer, only 6 are theory based (Hamel, Thompson, Albrecht, & Harper, 2019). Additionally, fewer than 20% refer to empirical studies or background source information, only 11.3% are evidence-based, and just 10.3% to 50% involved clinicians in their development processes (Collado-Borrell et al., 2016; Giunti, Giunta, Guisado-Fernandez, Bender, & Fernández-Luque, 2018; Zhao, Freeman, & Li, 2016). Such omissions result not only in gaps between the offered eHealth solutions and the needs and preferences of end users, but also in ineffective content in terms of behavior change (Siqueira do Prado, Carpentier, Preau, Schott, & Dima, 2019).

Such research-to-practice gaps sometimes reflect differences between the design context and the implementation context: design decisions might not acknowledge end users' or contextual restrictions; or they might simply use too many resources (Dopp, Parisi, Munson, & Lyon, 2019). To narrow the gaps between successful trials and long-term real-world use, various researchers have proposed combining computer science methods such as user-centered design (UCD) with implementation science principles (Chen, Neta, & Roberts, 2020; Dopp, Parisi, Munson, & Lyon, 2020) or behavioral science (Hamel et al., 2019).

Each of these three fields can add to the others' strengths. By providing a firm theoretical backing for content development, behavioral science principles are essential for effective behavior change interventions. UCD is an iterative process that places end user needs at the center of the design process, with the goal of developing highly usable and accessible products (Nielsen, 1994). And implementation science, as the study of methods that facilitate the uptake of research findings in clinical practice (Eccles & Mittman, 2006), applies specific methodological considerations (e.g., contextual analysis and stakeholder involvement) to assure the integration of content-specific requirements in the final result (Peters, Adam, Alonge, Agyepong, & Tran, 2013).

Considering these three methodologies' individual advantages, we believe that a mix of the three (implementation-, behavioral-, and computer science methods) can help balance context with end user needs, while guaranteeing a robust theoretical underpinning for content. If this proves true, we further expect that it will speed the resulting interventions' technology adoption while increasing their overall sustainability. To the best of our knowledge, no paper has yet been published on the combined use of the three methodologies. Even combinations of two of the three are rare and are employed mainly in the conception phase of eHealth-facilitated care models (Chen et al., 2020; Dopp et al., 2020; Hamel et al., 2019)

Besides, descriptions of eHealth tools are mostly limited to reports on their effectiveness regarding outcomes or health behaviors. The process of how associated eHealth software components are created is mentioned only marginally, if at all. In particular, questions regarding how end users have been involved, which theoretical framework guided content development, or how collaboration was organized within the development team remain unanswered. This leaves developers without guidance. Even worse, the team responsible for defining content is often completely separate from that creating the software. But handing over a fully defined blueprint to a software team and waiting until it is finished bears the obvious risk that the delivered components will not look or function as intended (Schwaber & Beedle, 2002). In software development, employing agile software development processes can prevent just such a situation from arising.

Coined in 2001, the term agile software development signifies a family of development processes that value individuals and interactions, working software, customer collaboration, and responses to change over processes and tools, comprehensive documentation, contract negotiation, and following a plan (Beck et al., 2001; Hohl et al., 2018). As the name implies, it promotes fast and iterative development: working software increments are created, delivered, and discussed regularly via collaboration between self-organizing, cross-functional teams (Schwaber & Beedle, 2002).

This article reports on our approach to applying UCD and agile software development principles in an implementation science project underpinned by a theory-guided content development process using the Behavior Change Wheel (BCW; Michie, Atkins, & West, 2014). We describe this process by detailing the development of the software components for an ICM in allogeneic stem cell transplantation facilitated by eHealth (SMILe). The SMILe ICM combines human as well as software components. First, within the outpatient transplant team, it embeds an advanced practice nurse in the role of care coordinator (CC). Within the first year post-alloSCT, the CC delivers each patient 12 face-to-face visits to support patients' self-management and other health-related behaviors. Second, it includes the SMILe technology, which supports the follow-up process and complements the CC-delivered intervention content. The SMILe technology consists of a mobile app (SMILeApp) for patients, and a browser app (SMILeCare) for the CC. With the SMILeApp, patients can record notes on their daily well-being and a set of 3 medical (blood pressure, temperature, weight), 3 behavioral (steps, medication intake, infection prevention measures), and 13 symptom-related parameters (pain; signs of bleeding; nausea; emesis; diarrhea; skin rash; mouth or throat sores; shortness of breath; pain or burning at urination; fatigue, tiredness, or lack of energy; difficulty swallowing; decreased appetite; cough). The app is also complemented by self-management information about conditions and symptoms. The entered data are transferred to the hospital's data center, where, with the patient's consent, the CC can view all incoming values, visualize their development over time in SMILeCare, and contact the patient via telephone if necessary. This remote monitoring enables the CC to provide continuous and extended follow-up care for a larger number of alloSCT patients and to detect signs of possible health deterioration early.

Methods and Findings

We developed the SMILe ICM's software components via a four-part process: (a) setting up an interdisciplinary team; (b) performing a contextual analysis, the results of which would drive the content development process and allow us to define possible software functionalities based on behavioral theory; (c) transferring content to software following agile software development principles; and (d) regularly involving stakeholders and end users following UCD principles. The following paragraphs describe each of the abovementioned processes and the methods used to perform them.

The Interdisciplinary Team

One primary agile software development technique is the formation of a cross-functional, interdisciplinary development team (Schwaber & Beedle, 2002). All competences necessary to accomplish the project should be available within this team. Therefore, we involved experts in implementation science (nursing scientists), behavioral science (psychologists, nursing scientists), and software engineering (software developers, usability experts). While each team member preserves his or her area of expertise, all should be regarded as members of a single team rather than as independent actors sharing a common vision. Nevertheless, not everyone can be involved in every task.

Therefore, two sub-teams were formed—one responsible for content development, the other for software development—while ensuring that representatives of each group were present at the other's meetings. Our meeting intervals and constellations are described in detail in the section "Transferring Content to Software Following Agile Software Development Principles," where we describe the collaboration process.

The development team needs exchanges with two other stakeholder groups: clinicians working in the field (in our case, allogeneic stem cell transplantation: physicians, clinical nurse specialists, nurse managers) and the end users. Their involvement is described in detail in the section "Stakeholder and End User Involvement Following User-Centered Design Principles."

Driving the Development Process via the Contextual Analysis Findings

After setting up the team, we performed a contextual analysis (Figure 1, a) to understand any context-specific characteristics and practice patterns as well as the technology openness of the target setting's clinicians and patients. A detailed description of the used methodology and results of the contextual analysis are available elsewhere (Leppla et al., 2020). While the contextual analysis was mainly performed by nursing scientists, members of the software team accompanied them and observed the process to develop a shared understanding. The results indicated support needs in four areas: (a) monitoring and follow-up of symptoms to improve patients' symptom recognition and evaluation of how to react to them; (b) infection prevention measures; (c) medication adherence; and (d) physical activity (Leppla et al., 2020). We used the BCW as our guiding behavioral theory while developing the content of each intervention module (Figure 1, b) (Michie et al., 2014). The BCW is a widely used behavioral theory. By combining 19 previously developed behavior change frameworks, it effectively helps researchers understand, explain, and modify behaviors. With the Capability-Opportunity-Motivation-Behavior (COM-B) Model at its hub, it supports a rigorous approach to developing effective behavior change interventions. A detailed description of how this process was applied to the medication adherence module is described elsewhere (Ribaut et al., 2020).

Because the contextual analysis also revealed that not all necessary functionalities should (or could) be delivered by technology, their modes of delivery (face-to-face or technology) were decided and intervention descriptions for the face-to-face sessions (Figure 1, d) were formulated alongside the functionalities of the software parts. These were formulated into user stories, a common method of describing desired functionalities in agile software development. These typically follow the role-feature-reason format: within one sentence the role (As a patient or clinician), the feature (I want to [action], e.g., monitor my pain intensity), and the description of the reason (so that [expected outcome], e.g., I can keep track of my pain trajectory) are all explained.

The user stories as formulated by the content team were written down in a prioritized list, the so-called product backlog, which then contains the main functionalities to be programmed (Figure 1, c). Each story's priority was gauged by the content team based on the contextual analysis. According to agile philosophy, this priority can change according to the insights gained during the realization process. Since the user stories only focus on desired main functionalities, those functionalities' exact details, mechanics, and appearance within the working software will be defined in collaboration with the software team immediately before creation. To guide software production, more detailed stories will be formulated.

For the SMILe ICM, 39 user stories were formulated for the four target areas, with the nine focusing on monitoring and follow-up of symptoms assigned the highest priority. These nine (listed in Table 1) were selected for the realization of a first intervention module.

Transferring Content to Software Following Agile Software Development Principles

This section describes the interdisciplinary team's collaboration process, which relies heavily on agile software development techniques. One characteristic is to develop the necessary components in iterations (sprints), which typically last from 2 weeks to 3 months (Beck et al., 2001; Schwaber & Beedle, 2002). In planning each sprint, subsets of the user stories to be realized in the next iteration are selected, based on their priority (Cohn, 2004). To decide how many of the tasks at hand can be included in each sprint, the software developers estimate the expected time and effort necessary for each and file them in a sprint backlog (Figure 1, e).

Because user stories roughly encapsulate each desired functionality, they have to be understood fully by the

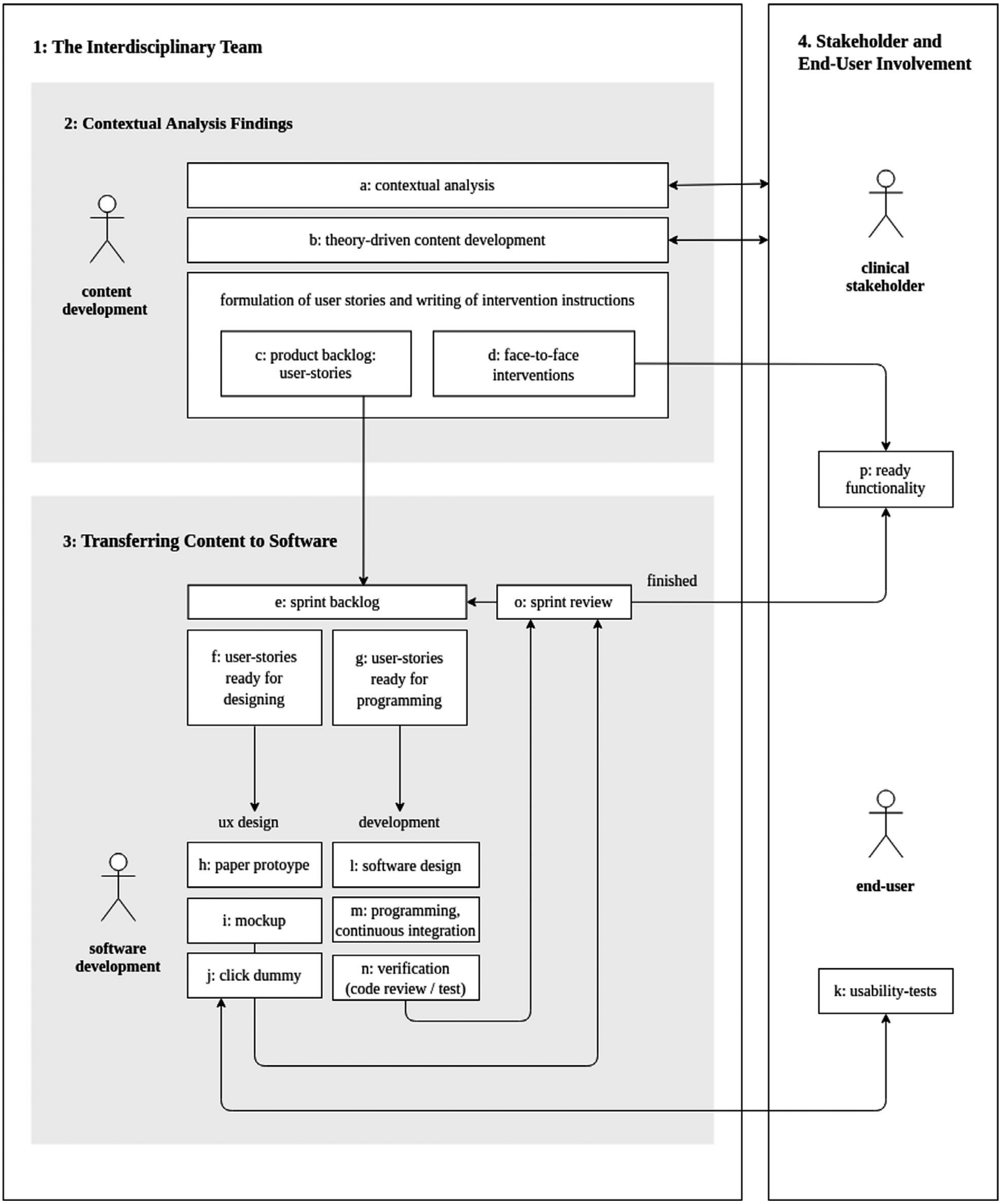
1: The Interdisciplinary Team
2: Contextual Analysis Findings
content development
a: contextual analysis
b: theory-driven content development
formulation of user stories and writing of intervention instructions
c: product backlog: user-stories
d: face-to-face interventions
3: Transferring Content to Software
e: sprint backlog
o: sprint review
finished
f: user-stories ready for designing
g: user-stories ready for programming
ux design
development
h: paper protoype
i: mockup
j: click dummy
l: software design
m: programming, continuous integration
n: verification (code review / test)
software development
4. Stakeholder and End-User Involvement
clinical stakeholder
p: ready functionality
end-user
k: usability-tests

Figure 1. Overview of the SMILe software development process.

software team before programming begins. This demands close collaboration between the content developers and software team's user experience (UX) designers, who ensure that the user interface is functional, comfortable, and pleasantly designed. Factors influencing the desired functionality's definition should be explained

Table 1. Extract From the SMILe User Stories as Formulated by the Content Team

	SMILe role	SMILe feature	Expected outcome
1	As a patient,	… I want an electronic system to assess symptoms	… so that I feel more secure.
2	As a CC,	… I want an electronic system to monitor symptoms and vital signs of patients at home within the hospital	… so that I can detect complications early.
3	As a patient,	… I want a daily reminder for using the system	… so that I do not forget to enter my data.
4	As a patient,	… I want to have the option to assess and share entered data	… so that I have the certainty that someone is watching over me.
5	As a patient,	… I want feedback on my self-assessed vital signs and symptoms	… so that I have support in my self-management and decision making.
6	As a CC,	… I want an overview of how frequently patients entered their data into the system	… so that I can give feedback.
7	As a patient,	… I want positive feedback when I use the SMILeApp on a regular basis	… so that I keep motivated.
8	As a patient,	… I want an overview about my entered data	… so that I can see changes over time and feel motivated.
9	As a patient,	… I want to have contact information within the system	… so that I know who to contact.

Note CC = care coordinator.

and possible approaches discussed. After that, the user stories are ready for the visual and interaction design (Figure 1, f). Only after these processes' respective drafts are satisfactory are the user stories ready for programming (Figure 1, g). To sufficiently define each functionality's specifications, then, the UX designers always have to work at least one sprint ahead of the programmers.

One efficient way of communicating and discussing ideas at an early stage is paper prototypes (Figure 1, h; Figure 2). This category of low-fidelity prototype is determined by rather rough, analog-crafted monochrome wireframes, which lay out the basic structure and target functionalities. The users move across the software (user flow) to use each functionality, which should already be recognizable. After the paper prototype has been discussed, the next stage is to convert it into a mockup (Figure 1, i). This high-fidelity, static design representation of the user interface serves as a draft canvas upon which to experiment with colors, shapes, textures, spacing, and fonts. Such a mockup is useful not only for communication with the content team, but also for the programming of the components. In a third step, for major functionalities, click dummies can be created (Figure 1, j). For these, a partially interactive demo of the user interface is constructed to simulate the tested software's functions and end user interactions (Figure 1, k).

Once the designs are ready to give to the programming team, the technical aspects of the functionalities are further detailed (Figure 1, l) and programming starts (Figure 1, m). Regular integration of the new software into the existing code ensures that a concise software version is always available. Functional unit-level software tests and code reviews are essential to obtain high software quality (Figure 1, n). During the sprints, the software team needs at least weekly meetings with the content team in order to clarify any remaining open details, with openings for more to deal with urgent matters.

After the completion of each sprint, the designs or implemented functionalities from the current sprint are presented and discussed within an interdisciplinary team sprint review (Figure 1, o). All stakeholders have to decide whether the requirements of user stories have been applied correctly. With the acceptance of designs, the related user story is ready for programming and any complete programmed functionalities (Figure 1, p) can be released into the final software. Partially completed or insufficient work results are considered incomplete and have to be allocated to the next sprint.

Stakeholder and End User Involvement Following User-Centered Design Principles

Since stakeholders in the field and end users are not part of the interdisciplinary team, team members have to be very careful to adequately incorporate their needs, ideas, and feedback.

Involving stakeholders. All clinical stakeholders were involved by inviting them to regular meetings with the interdisciplinary team. In the project's initial stage, we also invited patient representatives. Due to their high symptom burden, patients could not attend continuously. Meetings were held at least twice yearly during the 2

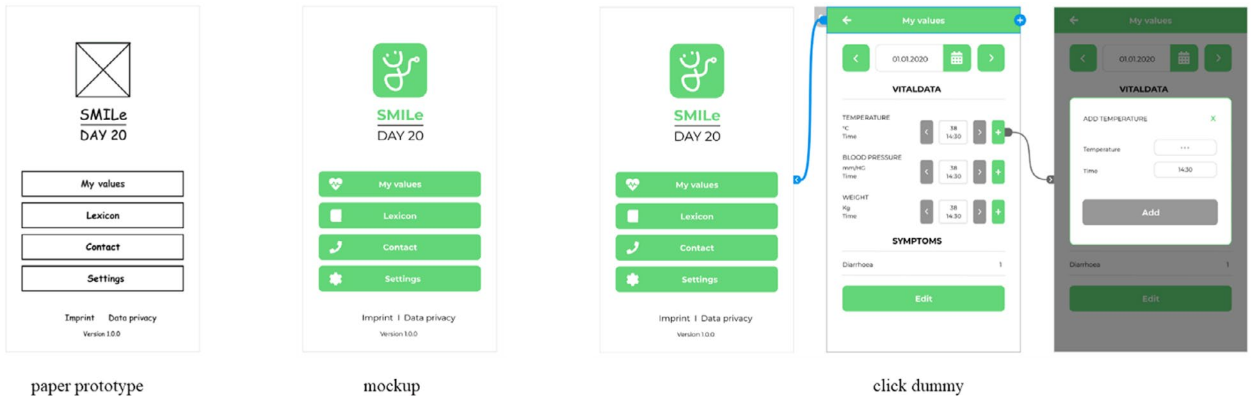

Figure 2. SMILeApp user experience design process. [Colour figure can be viewed at wileyonlinelibrary.com]

years of development—more frequently if decisions were urgently needed. This ensured a context-adapted development process that integrated setting-specific stakeholder needs.

Involving end users. End users were directly involved at three points. The first was very early in the process during the contextual analysis (Leppla et al., 2020). Later, for the click dummy usability tests and, of course, for the completed software, end users' feedback was also essential. For usability tests, end users are asked to verbalize their thoughts (think-aloud method) as they perform specific application-based tasks while UX experts observe and document their behavior. As an evaluation instrument, usability testing is designed not only to determine whether an item is adequately adapted to the target group and their needs, but also to ensure that its use entails no risks (Bastien, 2010).

One quick and widely used means to quantitatively assess a user's satisfaction with a software system is the System Usability Scale (SUS) (Brooke, 1996). The SUS is a highly robust, reliable, and valid 10-item questionnaire using 5-point Likert-style response options (4 = *strongly agree*; 0 = *strongly disagree*). The five odd-numbered items are calculated by subtracting 1 from the raw score, and even-numbered items by subtracting the raw score from 5. All scores are summed and multiplied by 2.5 to yield a total score between 0 and 100. SUS scores above 68 are considered above average; those above 80 belong to the top 10% of user experience (Sauro, 2011).

Usability test results. We performed two rounds of classical user testing. For the first round we recruited a convenience sample of five alloSCT patients from the outpatient clinic by asking all available patients on a certain day whether they would be willing to take part in the user test. For the second round, we recruited 6 alloSCT patients over 1 week by applying a purposive sampling approach to guarantee that all educational levels, genders, and ages are represented by the two user tests. Patients of both groups tested the SMILeApp interface regarding the symptom monitoring and follow-up module. According to existing guidelines, this number of patients is sufficient to identify up to 80% of usability problems (Kushniruk & Patel, 2004). The first test included slightly younger patients (test 1: mean age 43.7 years; test 2: mean age 48.2 years). Regarding gender and cohabitation status, the two groups were very similar, with 60% male and 90% living in partnerships for both; however, 90% of test group 1 patients had completed at least some post-secondary education, while 90% of those in group 2 had not. Each round included 14 practical tasks (e.g., opening the SMILeApp, entering blood pressure), which the patients had to solve while thinking aloud. After completing the tasks, the participants filled in their SUS questionnaires, yielding mean scores of 88 points for group 1 and 79.5 points for group 2. We also elicited and integrated feedback on SMILeCare's usability from the hospital CCs via regular discussion rounds.

Extended end user involvement. Besides incorporating patients in the contextual analysis and usability tests, further patient involvement was sought over the course of development. Partly to help the development team settle design questions, and partly to increase the software's acceptability, we set up an additional round of patient input, this time with a self-help group of cancer survivors (*n* = 11) and caregivers (*n* = 8). After presenting feature designs based on an intermediate version of the app, we asked them about their preferences, particularly concerning the frequency of data entry, different SMILeApp logos, different types of reminders, and feedback on set behavioral goals (e.g., reached step goal). Because monitoring of behavior was one included behavior change technique, we were interested in how frequently patients would be willing to enter data in terms of a performed behavior. Of the 11 patients, 6 were willing to enter medical and symptom data daily, 4 every third day, and 1 once weekly. Regarding a pro-active weekly schedule for questions monitoring positive health behaviors, 12 of 19 patients and caregivers would be willing to receive four questions distributed across the week, 5 no more than two questions, and 2 just one per week (e.g., Did you remember to apply correct hand hygiene?). All 19 favored the idea of daily pop-up reminders, preferably graphical, to enter data into the app.

Discussion

In this article, we described how, within an implementation science project, agile software development principles can be applied alongside user-centered design and theory-guided content development to create and embed tailored software solutions in an ICM. This contrasts with the more frequent approach of integrating software developed in an unrelated process or even preexisting software that basically fits the purpose. In our experience, there are typically two main approaches to eHealth software development: either software companies drive the process, seeking contact with medical teams to receive content to be implemented, or clinician teams specify a concept to be programmed by a software team. In either case, even if both groups provide their best efforts to do their jobs separately, gaps remain between content development and software production, making inaccuracies and misunderstandings virtually inevitable.

Team Effort

To minimize such shortfalls, we treated the content developers and the software developers as a single interdisciplinary team working towards a custom-fit software solution. Accordingly, we included meetings between various constellations of specialists at various intervals in our agile development process. Including software developers in the contextual analysis and nursing scientists in the software development process (e.g., at the sprint reviews and usability tests) greatly helped these groups' mutual understanding. Similarly, regular feedback from both stakeholders in the field and end users helped validate the anticipated solutions.

Theory-Based Content Development

Guided by a solid contextual analysis underpinned by the BCW as behavioral theory, we were able to build meaningful theory-based user stories as a basis for software development. Since this employment of user stories was newly introduced to the content team by the software team, the content developers were often tempted to incorporate design ideas into the user stories early in the process; however, design and interaction ideas should be developed at later stages, in collaboration with the UX team, as its members have greater expertise in such matters.

Application of User-Centered Design

The use of UCD techniques, especially mock-ups and user flows, increased the discussion between the content and UX teams. The early usability tests helped to identify several weaknesses affecting previous designs, leading to an improved user interface. This part of the process was rather straightforward to accomplish: as the software programming remained incomplete, the programmers were less reluctant to make changes.

In line with the UCD principles of (a) focusing on users and tasks, (b) measuring usability empirically, and (c) designing and testing usability iteratively (Dabbs et al., 2009), usability tests offered opportunities for members of both the content team and the software team to evaluate their work. The tests were not only used summatively (i.e., to determine the SUS score), but also formatively (i.e., to help us find improvable points within the software). With a score of 88, the first test group's mean SUS score was already within the top 10% of possibilities (>80) (Sauro, 2011). The second group's lower mean SUS score (79.5) might reflect the complexity of the functionalities tested, that group's higher mean age, or their lower educational level. This indicates that the raw numbers have to be interpreted with caution. For our purposes, the qualitative results of the think-aloud methods were more beneficial.

Principles of Agile Software Development

This project's use of iterative processes reduced its overall complexity by dividing it into manageable parts. Typically, each iteration deals with several user stories, but only as many as the software team thinks can be completely finished in the time allotted. Ideally, each iteration review includes the presentation of a usable piece of software that adds value for the customer. In our setting, though, we found that this was often not possible, as the content team's stories were too long to finish in a single iteration and had to be broken down into multiple shorter ones. Also, our process, whereby the stories first have to be designed, then programmed, led to the same story stretching across several sprints. For example, after looping two or more times through the design cycle, a story could do the same with the programming cycle.

Not knowing how many iterations would be necessary made it difficult to schedule the associated functionalities' release dates. Because our two-weekly sprints were rather short, the most obvious solution would be simply to lengthen the sprints to deal with each story. On the other hand, since no components were released before we agreed they were satisfactory, the quality of the resulting software components was very high. As noted above, this was reflected in the end users' test ratings.

One considerable drawback of using agile software development in a medical context is that the agile development style values working software over highly detailed documentation. This priority conflicts with the U.S. Food and Drug Administration's (FDA's) stipulations surrounding Software as Medical Devices and with the European Union's Medical Device Regulation (MDR; EU2017/745), both of which define apps with certain functionalities as medical devices. Both cases entail exhaustive documentation of each detail of the production process.

For the first SMILe module, then, since the certification process was simply not feasible for our group, we chose to realize only functionalities that fall outside the MDR classification (e.g., rather than receiving integrating algorithm-based automated feedback via smartphone messages, patients can look them up in a lexicon integrated into the app). The chosen functionality is now operational and can be used by patients as well as caregivers on the targeted devices.

In future work, we hope to tackle the remaining functionalities. This will require following a more stringent software development process—a need we can meet by defining specification and verification documents as required by the FDA or ISO62304 as sprint goals and further formalizing the development process.

In February 2020, following the agile principle of evaluating prototypes early, we initiated a randomized controlled trial (DRKS00020347) at Germany's Freiburg im Breisgau, University Hospital (FiB). As the first participating center, FiB is using only the symptom monitoring and follow-up eHealth module. At the time of writing, the other modules are in a face-to-face only delivery mode. In parallel, the software development team is constructing software features to continue with the digitalization process.

Conclusions

Across countries, diseases, and settings, there can be little doubt that eHealth will play a central role in the future of health care. However, successful implementation of the necessary technology into any clinical setting requires first thoroughly analyzing not only end users' needs but the entire target context, then using the results to develop effective and meaningful content. In our case, an interdisciplinary team with expert knowledge of implementation science, behavioral science, and computer science methods—particularly UCD and agile software development—facilitate the planning, design, construction, and testing of eHealth components with an increased chance of uptake, adoption, and long-term sustainability. Reflecting context-specific requirements, these components are tailored to end users' needs and can effectively influence their behavior in ways we fully expect will lead to improved medical outcomes. In addition to providing guidance for other groups interested in developing and integrating eHealth components into care models, this report presents an innovative method of redesigning chronic illness care towards more effective and resource-efficient eHealth-facilitated clinical processes.

Acknowledgments

We would like to thank Chris Shultis for editing the manuscript, and the German B. Braun Stiftung and the Swiss Cancer League for funding our work.

The SMILe Study Team in alphabetical order were: Sonja Beckmann, Dora Bolliger, Yves Chalandon, Sabina De Geest, Sabine Degen, Fabienne Dobbels, Nathalie Duerinckx, Monika Engelhardt, Margarita Fürmann, Michael Fürmann, Sabine Gerull, Florian Grossmann, Monika Hasemann, Phillip Heidegger, Sandra Hobelsberger, Mylen Husel, Toska Kelmendi, Katharina Koehly, Rainer Krais, Marina Lemcke, Lynn Leppla, Birgit Maier, Anne-Claire Mamez, Stavoula Masouridi, Juliane Mielke, Gyathri Nair, Daniela Neupert, Jakob Passweg, Stefan Pschenitza, Siegrun Reitwiessner, Jannette Ribaut, Dennis Rockstein, Urs Schanz, Anja Schmid, Helen Schoemans, Tobias Schulz, Vanessa Schumacher, Yulia Senft, Alexandra Teynor, Anja Ulrich, Sabine Valenta, Viktor Werlitz, Verena Witzig-Brändli, Robert Zeiser.

Clinical Resources

- agilemanifesto.org. Manifesto for agile software development. https://agilemanifesto.org/
- MeasuringU. Measuring usability with the System Usability Scale (SUS). https://measuringu.com/sus/

- Universität Basel. SMILe website. https://nursing.unibas.ch/de/forschung/forschung-227/forschung/smile/
- World Health Organization. eHealth definition. https://www.who.int/ehealth/about/en/

References

Aapro, M., Bossi, P., Dasari, A., Fallowfield, L., Gascón, P., Geller, M., … Porzig, S. (2020). Digital health for optimal supportive care in oncology: Benefits, limits, and future perspectives. *Supportive Care in Cancer*, *28*, 4589–4612. https://doi.org/10.1007/s00520-020-05539-1

Barello, S., Triberti, S., Graffigna, G., Libreri, C., Serino, S., Hibbard, J., & Riva, G. (2016). eHealth for patient engagement: A systematic review. *Frontiers in Psychology*, *6*, 2013.

Bastien J. M. C. (2010). Usability testing: a review of some methodological and technical aspects of the method. *International Journal of Medical Informatics*, *79*(4), e18–e23. http://dx.doi.org/10.1016/j.ijmedinf.2008.12.004

Bates, D. W., & Wright, A. (2009). Evaluating eHealth: Undertaking robust international cross-cultural eHealth research. *PLoS Medicine*, *6*(9), e1000105. https://doi.org/10.1371/journal.pmed.1000105

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … Jeffries, R. (2001). *Manifesto for agile software development*. Retrieved from https://agilemanifesto.org/

Brooke, J. (1996). SUS—A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4–7.

Chen E., Neta G., Roberts M. C. (2020). Complementary approaches to problem solving in healthcare and public health: implementation science and human-centered design. *Translational Behavioral Medicine*, http://dx.doi.org/10.1093/tbm/ibaa079

Cohn, M. (2004). *User stories applied: For agile software development*. Boston, MA: Addison-Wesley Professional.

Collado-Borrell, R., Escudero-Vilaplana, V., Ribed-Sánchez, A., Ibáñez-García, S., Herranz-Alonso, A., & Sanjurjo-Sáez, M. (2016). Smartphone applications for cancer patients: What we know about them. *Farmacia Hospitalaria*, *40*(1), 25–35.

Dabbs, A. D. V., Myers, B. A., McCurry, K. R., Dunbar-Jacob, J., Hawkins, R. P., Begey, A., & Dew, M. A. (2009). User-centered design and interactive health technologies for patients. *Computers, Informatics, Nursing*, *27*(3), 175.

Dopp, A. R., Parisi, K. E., Munson, S. A., & Lyon, A. R. (2019). Integrating implementation and user-centred design strategies to enhance the impact of health services: Protocol from a concept mapping study. *Health Research Policy and Systems*, *17*(1), 1. https://doi.org/10.1186/s12961-018-0403-0

Dopp, A. R., Parisi, K. E., Munson, S. A., & Lyon, A. R. (2020). Aligning implementation and user-centered design strategies to enhance the impact of health services: Results from a concept mapping study. *Implementation Science Communications*, *1*(1), 1–13.

Eccles, M. P., & Mittman, B. S. (2006). Welcome to *Implementation Science*. *Implementation Science*, *1*(1), 1. https://doi.org/10.1186/1748-5908-1-1

Elbert, N. J., van Os-Medendorp, H., van Renselaar, W., Ekeland, A. G., Hakkaart-van Roijen, L., Raat, H., … Pasmans, S. G. (2014). Effectiveness and cost-effectiveness of ehealth interventions in somatic diseases: A systematic review of systematic reviews and meta-analyses. *Journal of Medical Internet Research*, *16*(4), e110.

Furlong, E., Darley, A., Fox, P., Buick, A., Kotronoulas, G., Miller, M., … Maguire, R. (2019). Adaptation and implementation of a mobile phone-based remote symptom monitoring system for people with cancer in Europe. *JMIR Cancer*, *5*(1), e10813. https://doi.org/10.2196/10813

Giunti, G., Giunta, D., Guisado-Fernandez, E., Bender, J., & Fernández-Luque, L. (2018). A biopsy of breast cancer mobile applications: State of the practice review. *International Journal of Medical Informatics*, *110*, 1–9.

Granja, C., Janssen, W., & Johansen, M. A. (2018). Factors determining the success and failure of ehealth interventions: Systematic review of the literature. *Journal of Medical Internet Research*, *20*(5), e10235.

Hamel, L. M., Thompson, H. S., Albrecht, T. L., & Harper, F. W. (2019). Designing and testing apps to support patients with cancer: Looking to behavioral science to lead the way. *JMIR Cancer*, *5*(1), e12317.

Hohl, P., Klünder, J., Bennekum, A. v., Lockard, R., Gifford, J., Münch, J., … Schneider, K. (2018). Back to the future: Origins and directions of the "Agile Manifesto"—Views of the originators. *Journal of Software Engineering Research and Development*, *6*, Article 15.

Jeffs, E., Vollam, S., Young, J. D., Horsington, L., Lynch, B., & Watkinson, P. J. (2016). Wearable monitors for patients following discharge from an intensive care unit: Practical lessons learnt from an observational study. *Journal of Advanced Nursing*, *72*(8), 1851–1862.

Kuijpers, W., Groen, W. G., Aaronson, N. K., & van Harten, W. H. (2013). A systematic review of web-based interventions for patient empowerment and physical activity in chronic diseases: Relevance for cancer survivors. *Journal of Medical Internet Research*, *15*(2), e37. https://doi.org/10.2196/jmir.2281

Kushniruk, A. W., & Patel, V. L. (2004). Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of Biomedical Informatics*, *37*(1), 56–76.

Leppla, L., Mielke, J., Kunze, M., Mauthner, O., Teynor, A., Valenta, S., … Zeiser, R. (2020). Clinicians' and patients' perspectives on follow-up care and eHealth support after allogeneic hematopoietic stem cell transplantation: A mixed-methods contextual analysis as part of the SMILe study. *European Journal of Oncology Nursing*, *45*, 101723. https://doi.org/10.1016/j.ejon.2020.101723

Majhail, N. S. (2017). Long-term complications after hematopoietic cell transplantation. *Hematology/Oncology and Stem Cell Therapy*, *10*, 220–227.

Melchiorre, M. G., Papa, R., Quattrini, S., Lamura, G., Barbabella, F., & Consortium, I. E. (2020). Integrated care programs for people with multimorbidity in European countries: eHealth adoption in health systems. *BioMed Research International*, *2020*, 9025326.

Michaud, T. L., Zhou, J., McCarthy, M. A., Siahpush, M., & Su, D. (2018). Costs of home-based telemedicine programs: A systematic review. *International Journal of Technology Assessment in Health Care*, *34*(4), 410–418.

Michie, S., Atkins, L., & West, R. (2014). *The behaviour change wheel: A guide to designing interventions*. Surrey, UK: Silverback.

Nielsen, J. (1994). *Usability engineering*. San Francisco, CA: Morgan Kaufmann.

Peters, D. H., Adam, T., Alonge, O., Agyepong, I. A., & Tran, N. (2013). Implementation research: What it is and how to do it. *British Medical Journal*, *347*, f6753.

Ribaut, J., Leppla, L., Teynor, A., Valenta, S., Dobbels, F., Zullig, L. L., … SMILe Study Team. (2020). Theory-driven development of a medication adherence intervention delivered by eHealth and transplant team in allogeneic stem cell transplantation: The SMILe implementation science project. *BMC Health Services Research*, *20*(1), 827. https://doi.org/10.1186/s12913-020-05636-1

Sauro, J. (2011). *Measuring usability with the System Usability Scale (SUS)*. Retrieved from https://measuringu.com/sus/

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.

Simblett, S., Greer, B., Matcham, F., Curtis, H., Polhemus, A., Ferrão, J., … Wykes, T. (2018). Barriers to and facilitators of engagement with remote measurement technology for managing health: Systematic review and content analysis of findings. *Journal of Medical Internet Research*, *20*(7), e10480.

Siqueira do Prado, L., Carpentier, C., Preau, M., Schott, A.-M., & Dima, A. L. (2019). Behavior change content, understandability, and actionability of chronic condition self-management apps available in France: Systematic search and evaluation. *JMIR Mhealth Uhealth*, *7*(8), e13494. https://doi.org/10.2196/13494

Thies, K., Anderson, D., & Cramer, B. (2017). Lack of adoption of a mobile app to support patient self-management of diabetes and hypertension in a federally qualified health center: Interview analysis of staff and patients in a failed randomized trial. *JMIR Human Factors*, *4*(4), e24. https://doi.org/10.2196/humanfactors.7709

Warrington, L., Absolom, K., Conner, M., Kellar, I., Clayton, B., Ayres, M., & Velikova, G. (2019). Electronic systems for patients to report and manage side effects of cancer treatment: Systematic review. *Journal of Medical Internet Research*, *21*(1), e10875. https://doi.org/10.2196/10875

Widmer R. J., Collins N. M., Collins C. S., West C. P., Lerman L. O., Lerman A. (2015). Digital Health Interventions for the Prevention of Cardiovascular Disease: A Systematic Review and Meta-analysis. *Mayo Clinic Proceedings*, *90*(4), 469–480. http://dx.doi.org/10.1016/j.mayocp.2014.12.026

World Health Organization. (2020). eHealth at WHO. Retrieved from https://www.who.int/ehealth/about/en/

Zhao, J., Freeman, B., & Li, M. (2016). Can mobile phone apps influence people's health behavior change? An evidence review. *Journal of Medical Internet Research*, *18*(11), e287. https://doi.org/10.2196/jmir.5692