

Agile User Experience Design for Product-Service Systems

(Agil upplevelsedesign för produkt-tjänstesystem)

Andreas Larsson
Lund University
andreas.larsson@design.lth.se

VINNOVA: Innovativ produktframtagning: Projekt av förstudiekaraktär 2012
Project reference: 2012-03765

1. Background

Many of today's consumer products are in essence product-service systems (PSS) (Baines et al, 2007), including a combination of hardware, software and services. This increases the complexity during both development and use. At the same time, the users' total experience of the "product in use" is central, meaning that developers must understand the product in terms of its social use context, and the needs that it is helping to fulfill, rather than merely understanding the technical context and worrying about whether or not the requirement specification has been met. The developers of such complex systems – and user experiences – must effectively integrate a wide variety of competences and professional disciplines as well as include both current and future users in the development.

Agile methods for quick and effective communication between customers, project managers and developers have gained significant attention in the software industry (Cockburn, 2007), although many companies find it difficult to include also aspects related to usability and user experience in what is still considered a relatively technology-focused development methodology. As companies to a greater extent than before are partnering with other companies to develop product-service systems, where software is part of a larger whole, there is a challenge to adapt these agile methods to better suit the development of a combination of hardware, software and services.

Research on agile methods confirm the challenge to integrate software development and user experience design (Sy, 2007; Hussain et al, 2009; Ferreira et al, 2012) with respect to, for instance, involvement of users, coordination of software developers and user experience designers, and balancing user testing with technical testing. If moving beyond the collaboration between software developers and user experience designers, the challenge becomes even greater.

One of the fundamental problems with agile methodology, commonly shared with other development approaches, is that internal or external "customer needs" are given greater attention during development than the needs of the end user, i.e. that demands on technical performance sets the majority of the agenda for development work. Too often, this results in products that might very well meet technical demands, but which risk missing the target with respect to the total user experience. Although usability testing is often performed, this testing is often carried out late in the development process, more as an evaluation of single

functions and properties of the software or hardware, than as a holistic evaluation of the product, service or experience that users pay for.

Developers are starting to understand that users actually do not buy products based solely on technical performance. Companies are increasingly redirecting their development work to attain useworthiness (i.e. is it worth using?) rather than merely usability (i.e. is it easy to use?). Companies need to make sure that they are "building the right it before they build it right" (Savoia, 2011), and to make that happen they need to develop, and continuously update, their understanding of user needs, attitudes and emotions in relation to products.

Taking the perspective of, for instance, Sony Mobile Communications – a partner company in this research project – there is a number of aspects that clearly shows how important it is to improve the methods and approaches for agile experience design. A mobile phone, including its software and services, is a complex product where many technical areas interact with continuously changing behavior patterns on the consumer side. These mobile communication devices (they are truly more than "telephones" today) contain processors, memory chips, radio transmitters and receivers, microphones, speakers, Wi-Fi, Bluetooth, batteries, cameras, touchscreens, GPS, accelerometers, gyroscopes, etc. They are still used for calling, but also to send and receive SMS, MMS and e-mail, to read the latest news, listen to music, surf the web, share documents, chat, send and receive pictures and movies, navigate with the help of maps, playing games, blogging, social networking, etc. The mobile phone could, for instance, also be seen as a symbol of status or as a device that helps keep people safe.

Sony Mobile Communications is continuously changing the appearance, content and functions on the products in the company's product portfolio, and an immense challenge is how to know – preferably already during the earliest stages of the development process – what value the proposed changes will add (or possibly destruct) while still making sure that new products and functions do not "leak" to consumers or competitors. The best tests are performed by real users in real situations, but how could such tests be performed as early and continuously as possible without risking the company's competitive edge? Sony Mobile Communications, and many other product developing companies, test early and often, but are often forced to use their own staff and focus on technical performance rather than usability and useworthiness in real contexts of use.

In a "normal" agile project in the software domain, the products are usually tested with real end users, but Sony Mobile Communications have millions of new and potential customers each year, which puts extremely high demands on understanding the user. Many agile software development projects are also based on clients ordering and specifying requirements on a specific product, whereas the mobile phone industry is forced to develop new functions without having enough insight into what end users really would value.

2 The Turn to Agile

Within the software development domain, the turn to agile methodology in the mid-1990s was a response to strongly regulated and micro-managed projects that too often resulted in software that was delivered late, over budget, and full of bugs (Denning, 2012). The

"waterfall model" of software development, where progress flows downwards in sequential phases, starting from well-defined requirements specifications, was criticized for being ill-fitted to the types of projects that were becoming increasingly common in the software industry – high-risk, high-exploration projects in which uncertainty is high (Highsmith, 2002).

As Highsmith (2002, p.4) puts it, these *"high-exploration projects, do not succumb to rigorous, plan-driven methods"*. Conformance to the plan has little meaning, and in likening such "extreme" projects to a battlefield situation, Highsmith (2002, p.4) argues that battlefields are managed by *"constant monitoring of conditions and rapid course alterations – by empirical process"*, where adapting to changing conditions is absolutely vital.

Kidd (1994) defined agility as a rapid and proactive adaptation of enterprise elements to unexpected and unpredicted changes. An equally important attribute of agility is the effective response to change and uncertainty (Kidd, 1994; Goldman et al, 1995).

Also the manufacturing industry turned to agile approaches when aiming to regain the leadership it lost in the 1970s and '80s. Companies like General Electric Aircraft Engines, IBM Corporation, Motorola Corporation, and Chrysler Motors Corporation took part in the development of the 21st Century Manufacturing Enterprise Strategy (Nagel & Dove, 1991), formulated to initiate a transition from mass production to agile manufacturing.

Although the original concept of agile manufacturing bears relatively little resemblance to what we would consider e.g. agile software development today, there are examples of agile-related approaches to manufacturing that resemble the practices from the software industry. For instance, the Wikispeed project (Denning, 2012), where Seattle-based C-Corporation developed a road-safety-legal automobile prototype in three months using off-the-shelf parts, combining radical management methods of Agile, Scrum and Kanban.

Today, product-oriented organisations are increasingly shifting their scope from selling "hardware" to providing "functions" (Ericson & Larsson, 2005; Baines et al, 2007) with the purpose of delivering added value to customers. This added value is not only based on technical performance, and is changing in meaning depending on the context and experiences of customers, users and other stakeholders. The types of problems that developers of product-service systems are facing are simply not the types of problems that they were facing when their industry was dominated by hardware or software "only".

One of the main reasons that companies are turning to agile-style methodologies is that they are increasingly facing "wicked problems" (Rittel & Webber, 1973), which means that a problem can not be understood without knowing its context. Furthermore, since there is no definitive problem, there can be no definitive solution. Paradoxically, then, you cannot clearly define the problem until you have solved it. In effect, the problem solving process does not stop at "final" and "correct" solutions, it stops when you run out of resources (such as time or money). The cost of striving for an optimal goal is simply too high, and since it is uncertain that the outcome of such an optimisation struggle will be much better, designers dealing with wicked problems often have to stop at "good enough" solutions – what Simon (1969) calls "satisficing". (Larsson, 2005)

For clarification, not all problems found in the product-service system domain are wicked problems. Even when developing such systems, there are still "tame" problems that might require approaches and methods that fall outside of the agile domain, for example if the problem statement does not change over time, if no disagreement is created over it, and if it is obvious that a proposed solution will or will not solve the problem. For instance, developing a "passenger-friendly airplane" is a wicked problem, while calculating stresses in the fuselage is a tame problem, though time-consuming, difficult and not at all trivial. (Larsson, 2005)

It is also important to acknowledge that responsiveness to change and uncertainty is only part of the challenge. Whereas uncertainty is when you have identified the variable but do not know its value, ambiguity is when you are not even sure what the variables are. Uncertainty is essentially about lack of information, whereas ambiguity is about lack of clarity (Schrader et al, 1993) or, as in Weick's (1979) concept of equivocality, the existence of multiple and conflicting interpretations about an organizational situation.

When it comes to development work in general, and innovation work in particular, reducing ambiguity is not always the goal, at least not early in the development process. Development and innovation is also about preserving ambiguity to support creativity and accelerate learning, and according to Leifer & Steinert (2011), it is important to not only live with change, but also to promote it, to create a maximum of flexibility and to support divergent activities by removing institutional and procedural barriers (Leifer & Steinert, 2011). Rather than focusing on searching for the "right" pieces of information to solve a well-defined problem, companies are increasingly focusing on identifying and understanding ill-defined problems by continuously exploring "unknown unknowns" (Rumsfeld, 2002) i.e. the things we do not know that we do not know.

"Through courting ambiguity, we can let invention happen even if we cannot make it happen. We can nurture behaviors that increase the probability of finding a path to innovation in the face of uncertainty. Emphasis is placed on the questions we ask as well as the decisions made." (Leifer & Steinert, 2011, p.151)

3 Agile User Experience Design for Product-Service Systems

In the Manifesto for Agile Software Development (Beck et al, 2001) the overarching values were described as:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In this section, the above values will be explored in light of empirical material, gathered through interviews with 20 practitioners representing a range of industries, functions, and roles. Some of the informants have extensive experience from developing and/or applying agile methodology, while others have had only limited encounters with values, principles and methods in the agile domain.

Below is a description of the main themes that emerged during the analysis of the interview material, organized according to the four overarching values of the Agile Manifesto. It should be noted that many informants talked about methodologies, methods and practices that could be described as "agile-style", including Agile, Scrum, Kanban, Lean Startup, Commitment Based Project Management and other complementary concepts. For the sake of clarity, we are using "agile" as an umbrella term for approaches that are consistent with the agile values above.

3.1 Individuals and interactions over processes and tools

On cross-functional communication:

One informant from a design consultancy mentioned that a holistic view is crucial for successful projects. In his experience, isolating competences often led to poor compromises where each individual performed their task without a view towards the overall goal. An informant from a medical technology company noted that cross-functional teams and co-localization is one key to success, and an informant from a window and door manufacturer mentioned that they are too dependent on individual functions today, wanting to move towards cross-functional research and development.

Several informants highlighted that the role of meetings in agile approaches might need to be reconsidered, since e.g. Scrum could actually mean that team members need to join more meetings than they would have in a waterfall-style process. One informant from the telecommunications industry commented that, in his experience, developers don't like meetings, and an informant from a design services consultancy noted that when his organization tried stand-up meetings, people still went into status meeting mode, waiting for instructions. The same informant expressed similar reluctance towards retrospectives, meaning that it is yet another status meeting they could do without. An informant from an advanced technology and research company described that people generally show limited interest in the details of other people's work and that such status meetings are like coffee breaks without the coffee.

One informant from an IT company commented on a lack of methods for communicating internally within the company. Similar comments were made from other informants, who reflected on how a strong focus on technology seems to make companies neglect aspects of information- and knowledge sharing – especially when working across functional, departmental or corporate boundaries.

On self-organizing teams:

An informant from a design services consultancy said that people are not used to autonomy and that he had never seen a self-organizing team. An informant from a medical technology noted, similarly, that entirely self-managed teams is a problem because there is a tendency to *"be ready when they are ready"*. The same informant commented, however, that agile approaches provide transparency in everyday work and leads to good team communication. However, the informant noted, it is crucial to keep team goals in focus to maintain flexibility that can be compromised by too strong individually targeted goals. According to this

informant, 4-8 persons in a team provides good "granularity" and allows for shared accountability among team members.

An informant from a window and door manufacturer commented that transparency in teamwork is key to creating a sense of urgency for team members. If you continuously monitor and evaluate progress against goals, it is natural that people develop an incentive for change, since no one wants to report, over and over again, that they have not succeeded with their part of the project.

One informant from the telecommunications industry mentioned the importance of having team members contribute to e.g. backlog grooming, not only the Product Owner. This goes for developing a vision as well. If teams become part of the planning, they understand the vision and engagement will be higher.

One the role of managers:

An informant from a packaging material company commented that some managers have a need for control, to know what everyone is doing. Such managers tend to like stage-gate approaches. According to this informant, the managers want great ideas and innovation, but they also must make sure to deliver in the short-term.

One informant from the telecommunications industry argued that managers might need to facilitate change by "*pushing a bit*". It is important to try new ways of working more than once. If team members do not like it after a couple of times, then stop. People need to see a value in what they do, the informant noted.

Although not specifically talking about the role of managers, an informant from the aerospace industry highlighted that some agile-style methods are self-disseminating after half an hour, which makes it easier for the organization to pick up on new ways of working in a bottom-up way. However, this informant noted, some methods require skilled facilitators, which is often forgotten. Another informant from a design and innovation consultancy commented that the success of some methods is highly dependent on who is facilitating the method.

An informant from the medical technology industry reflected on the multiple roles that a manager needs to fill. "*Sometimes you need to be captain, sometimes you need to be coach, sometimes you need to be a bridge builder.*" One of the informants from a project management consultancy commented on the need to clarify roles; who decides what to do, who defines the needs, who keeps an eye on the process, who is getting the work done, etc. Also, this informant noted, it is crucial to involve someone with expertise around the transformation.

On commitment:

One informant from a management consulting company commented on the need to improve the execution step, to get people to start doing things (agile or not), to create a discipline. An advocate for Commitment Based Project Management (CBPM) (Esque, 2010),

this informant talked about how to be agile without changing the process – how to essentially change work practices without violating the key processes in the company. One of the keys, this informant argued, is to avoid the activity trap, which according to Odiorne (1974, p.6) means that people “...tend to become so enmeshed in activity they lose sight of why they are doing it, and the activity becomes a false goal, an end in itself. Successful people never lose sight of their goals, the hoped-for outputs.”

This informant argued that the activity trap can be avoided by helping managers and team members making commitments, or promises, to each other rather than focusing on handing out tasks and activities. The idea behind the framework is that “*things get done because individuals make promises to each other and keep them.*” (Esque, 2010, p.1)

Inspired by, for instance, Sull & Spinoza’s (2007) work on Promise-Based Management, this informant was developing ways of “*building commitment muscle*” to increase coordination and collaboration, to increase agility, and to increase employee engagement. (Sull & Spinoza, 2007)

“*Because both sides have voluntarily agreed to the commitment—and have put their reputations on the line—they are likely to act with urgency and discipline.*” (Sull & Spinoza, 1997, p.3)

Another point with a commitment based approach to collaboration is that commitments focus a team on the future they have committed to creating, rather than the past.

“*So when a project team operating from commitments goes into a project review meeting, instead of discussing what went wrong, they are inclined to discuss what needs to happen now to make sure this week’s commitments are met.*” (Esque, 2010, p.4)

3.2 Working software over comprehensive documentation

On documentation:

An informant from an energy management company commented on how comprehensive documentation can be detrimental to development and innovation. Before agile methodology came in place, people had to use their imagination to figure out everything that anyone might need to know in a distant future. Now, it is much more effective to discuss the role of documents; for whom is it useful, for what purpose is it going to be used, and when it is going to be used. The informant suggested that team members would get the person who need the document to work with them.

On experimentation:

One informant from a design services consultancy noted that they could have tests running in a couple of weeks rather than months, which would mean a “*steady stream of results*”. With results being shared on a weekly basis, the informant argued that the “*results speak for you*”. When you can continuously show what works and what does not work, there is less need of extensive documentation.

An informant from an energy management company argued that agile offers a systemic way of experimenting, where you could get access to early real evidence instead of speculating over and over again. According to this informant, a speculation often becomes a fact, which can cause other teams to make decisions based on those speculations. The same informant mentioned that people working with financing in companies are so used to make *"fancy ten million Euro"* decisions that require evidence that backs up a large investment. In a more agile situation, they could say: *"We don't know what to build, and for whom. But we only need fifty-thousand Euro"*. Less lobbying is required, and experiments that previously were going on under the radar are now out in the open. The informant noted that they have achieved transparency and rather than relying on *"skunk work"* performed by a couple of persons who got along well, agile means that they can involve a larger part of the organization.

An informant from an advanced technology and research company commented that *"feasibility wins"*, meaning that companies tend to prioritize ideas that are feasible. For this informant, it is crucial to ensure high visibility, to help people see that things have happened. For example, they are exploring ways in which you could explore a new idea in 24 hours, e.g. *"hack days"*, which could then be turned into 30, 60, or 90 day challenges if needed. In line with the above informant, this informant mentioned that making experimentation official makes it OK. It is a way of *"materializing skunk works"* and reducing the mystique surrounding innovation and design. Another informant from the same company noted that several employees that previously have worked with technical development have felt a sense of relief when being exposed to a more agile and flexible service-oriented process, partly because they traditionally have been forced to create e.g. business cases before even starting to explore a new topic. In the new process, this is not required, or even feasible.

In other organizations, experimenting with cardboard is one of the ways in which new ideas are explored and communicated. An informant from a design consultancy commented that *"cardboard models have turned into our thing"*. The informant considered these models as very quick to build, even at full scale. Further, the informant noted, people see that it is a cardboard model rather than the real thing, which helps them understand that it has an explorative purpose. The models are not only important for communication with customers, but are also considered very useful to facilitate internal communication and negotiation. An informant from a packaging material company also commented that they have started to work more heavily with paper-based prototypes (e.g. cardboard, sack paper, kraft paper, etc.) One of their lab technicians hand-folded some 800 packages in a year, which made it easier to verify the value chain in-house.

An informant from a project management consultancy commented that we need to be looking at the cost of change. Refactoring is expensive when it involves concrete and steel, but when it comes to software, you can experiment. Regardless of the type of project, though, this informant found that failing fast and working with crude things are a couple of agile principles that are useful.

In light of these comments from informants, it could be useful to reflect on how a culture of experimentation changes the expectations on how a design and innovation process should

be organized, and what the expected outcomes are.

"When we reduce the cost of experimentation enough, the entire economics of how we develop products changes – it switches from a process based on anticipation (define, design, and build) to one based on adaptation (envision, explore, and refine). When the cost of generating alternatives plunges and the cost of integrating them into a product is low, then great products aren't built, they evolve..." (Highsmith, 2004, p.7)

Savoia (2011) makes a difference between prototype and pretotype. According to him, prototypes require an investment of days or weeks to answer the question "can we build it?". Pretotypes, on the other hand, require an investment of hours or days to answer the question "would we use it"?

One of the key concepts of lean manufacturing is that of "one-piece flow" (Shingo, 1989), essentially meaning that you should avoid to build inventory between processes. However, inventory in product development is both physically and financially invisible (Reinertsen, 2009, p.55), but there can still be queues that reduce quality by delaying feedback. For instance, an informant from a window and door manufacturer noted that there is a false safety in producing material (e.g. drawings) that are not immediately delivered to co-workers and/or colleagues.

"Even when ideas fail, engineers prefer to get rapid feedback, because it allows them to truncate a bad path quickly. It is inherently demotivating to invest a lot of effort proceeding in the wrong direction, and this ultimately leads to risk avoidance and reduced innovation. Small batch sizes are a powerful tool to increase urgency, motivation and accountability." (Reinertsen, 2009, p.117)

Prototypes are not only evaluation and validation instruments, they also accelerate learning and facilitate communication. When you are building a prototype, you are building to think, and building to learn. It is still somewhat of an explorative process in the sense that you are still trying to find out if your idea will work. In essence, you are also actively seeking out "what might not work", because finding out what fails takes you one step closer to finding out what works. As noted earlier, prototypes and experiments of various kinds in early stages lower the cost of learning, since they allow you to fail quickly while failure is still cheap.

On simulation:

Beth Comstock, Senior Vice President and Chief Marketing Officer of GE, has stated that *"Our traditional teams are too slow. We're not prototyping fast enough, not innovating fast enough. We need to systematize change."* (Safian, 2012)

A common critique of agile approaches is that they are suitable for software companies or for small startups, not for big companies delivering complex, capital-intensive solutions like jet engines, CT scanners or power transformers. This is something that Comstock herself responded to in a conversation with Lean Startup author Eric Ries with regard to how a concept like Minimum Viable Product (Ries, 2011) (e.g. any version of a product that can

begin the process of learning, using the Build-Measure-Learn feedback loop) would translate to an industrial setting like at GE: *"OK, software guy ... you can do 50 change orders in a day in software, but I'm making a jet engine, it's not exactly easy to change that..."* (Comstock & Ries, 2012)

Comstock further commented:

"Engineers want quality, they want perfection, and that's great, we want them to have that. Push the limits of science, but often it means things have to be perfect, and they have to appeal to every possible use case, every application, so before you know it you overengineer, and you have something that, that is hard to get to market, hard to understand for the customer ... I can get this out fast, I can test it, learn on the fly, and sort of giving people the tools and permission ... that resonates, I think, in any kind of industry setting." (Comstock & Ries, 2012)

One of the areas that is related to experimenting and testing fast to accelerate learning, is the area of simulation. In the conversation with Ries (Comstock & Ries, 2012), Comstock mentioned that GE is looking into how to get better analytics from their big, intelligent machines. A single jet engine produces 1 TB of data per day, and bringing that data to use is crucial.

One of the informants from an advanced technology and research company commented that there is a lot of data in IT systems, vehicle electronics systems, driving cycles, maintenance databases, etc. The majority of this data is currently not organized or coded, alternatively it is being used for diagnostics rather than for driving innovation and simulating future product-service systems.

An informant from the aerospace industry mentioned that simulations are cheap, in comparison, in this particular industry, where physical tests can cost several million EUR. In the development stages, it would simply be too expensive to rely on physical tests only. If a physical test fails, you lose the money and time spent on developing the tested product, and then you have to invest even more to come up with alternative solutions. Not to mention the costs related to delaying an entire aircraft programme.

3.3 Customer collaboration over contract negotiation

An informant from a construction equipment company mentioned that the company is ramping up the development of services, and that they now have a service development process that emphasizes the importance of getting close to customers. An informant from an advanced technology and research company noted that they have the ambition to get a more intimate understanding of the customer's activities. When it comes to service development, making a thorough need study is crucial. One challenge here, according to the informant, is that their employees are not used to being "out there" in real settings, and it is often difficult for them to come out and do observations and interviews.

One informant from an IT company commented that they are working a lot with customer discovery, and that there is a doctrine within the company that there are no answers in the

house. This means that people are eager to seek direct, rather than second-hand, sources of knowledge, and to not immediately trust what colleagues in the company are saying and how they are interpreting customer needs.

An informant from a metal working tool manufacturer noted that it is important to make tests with customers, but also to make sure that the knowledge from such tests must be multiplied to more situations, helping a lot of customers at the same time. According to this informant, this is a challenge. They have been largely technology focused, talking to customers about what they need within a specific tool segment. However, this informant commented, a lot of needs are related to products but are not product-specific. It is important to explore how current technologies could be used in new markets, or how new technologies could improve business in current markets. This informant reflected on good experiences with Gemba, a lean-inspired method that involves going to "the real place" to interact with customers, be it the manufacturing plant, or any other kind of context in which users live and work. However, while this informant reported that they get many good insights, there are evaluation systems in place (e.g. gates, profitability analysis, etc.) that make it harder to explore concepts that cannot be produced today.

Apart from observing and interviewing customers and end users, several informants spoke about the value in building closer partnerships with customers and other stakeholders in the development process. For instance, one informant from an energy management company mentioned that they have established a "*partner-like relationship*" with suppliers of e.g. prototypes. While there is an existing process for ordering prototypes that could take relatively long time, a closer partnership allows the company to get their prototypes quicker, partly by going beyond the buyer-supplier relationship and allowing the suppliers to understand how they could add value to the energy management company's products and processes. Similarly, an informant from a design consultancy mentioned a new method that they have developed that builds on close customer collaboration, to help customers understand what they are paying for. By performing this method as a team exercise with customers, a fruitful discussion about what to prioritize is created, and the customers are actually involved with both preparing and making the final decisions.

3.4 Responding to change over following a plan

Although the importance of accelerating feedback cycles to improve the speed of learning has already been emphasized in the previous sections, there are a few other insights that emerged from the interviews with respect to responding to change.

One informant from the telecommunications industry commented that the team is fixed, the time is fixed, but the scope is not fixed. According to this informant, quality is most important, so rather than squeezing everything in at the expense of quality, it is better to adjust the scope and wait until the next release. Teams might be frustrated when they do not deliver everything they have promised, and it is important for managers to talk about expectations with them and to let them understand that scope is variable and quality is the number one priority. The concept of "timeboxing" is well-known in agile methodology, and it basically means that you plan and design for what is most important.

It is interesting to note that this is not in contrast with the concepts of "good enough" or "satisficing". "Good enough" does not automatically mean that you sacrifice quality, it could mean that you sacrifice implementing some features to actually raise the quality of others. The important thing is that the you deliver something on time that is "sufficient-to-purpose" (Cockburn, 2007).

"Rather than try to perfect things as you go along, try to complete things first. Then go back and improve them." (Meier, 2010, p. 77)

An informant from an advanced technology and research company commented that the iterative nature of their service development process was crucial because this process relied heavily on understanding and refining customer needs rather than solving well-defined problems. In such cases, iteration is a natural part of the learning process, not something that is done in case something goes wrong. One informant from the energy management industry also commented that agile methodology means that it is now OK to engage in activities of an iterative nature, which was not always the case before.

4. Development of Research Methodology

The overall project consisted of a pre-study and a pilot project. The former focused on exploring the state of academic research and the state of industrial practice in the field, and the latter focused on exploring a way of working that was assumed to deepen academia's (both researchers' and students') understanding of industry needs and practices, to facilitate the uptake of academic results in industry, and to help the case company (Sony Mobile Communications) to evolve their user experience design methodology with respect to involving real users.

Pilot project with LTH and Sony Mobile Communications:

As noted earlier, when you develop software within an Agile/Scrum methodology, you are supposed to test your software as early as possible to get feedback. The best testing is done by the customer. However, with mobile phones it is not so easy to get in touch with the customers and companies often want to keep their development confidential.

At Sony Mobile Communications, Scrum is used to develop software, they test early and often, but most of the tests are done by the Scrum team (includes software developers, interaction designers and graphical designers, Section Manager and Product Owner); when the team tests the developed product, they are often looking at how it is designed according to the blueprint and the quality of the code, but they are more or less blind when it comes to usability. Usually, they do a comprehensive usability test once or twice a year on their software.

Sony Mobile Communications have tried to use internal people from Human Resources to test their products and it gave them a completely different result compared to their traditional team test, as they gave the team a very good insight in the User Experience problems that may arise. The only problem when they test with people from HR was that they had a hard time to understand the "real" problem and Sony Mobile Communications is

running out of new HR people very fast (they do not want to continue to use the same persons to test several times, as they will soon become familiar with the product like our team members).

A way to solve Sony Mobile Communication's need for new testing resources and get feedback that they can easier digest, involves starting a partnership with LTH and our UX, design and innovation researchers. Also, it is an opportunity to also give the students some practical experience in how to do usability studies (which was something that the Section Manager at Sony Mobile Communication missed when he learned about UX at the university).

The original idea was to start a limited pilot project (where the number of students and Sony teams could be expanded when we have an established way of working), where students from LTH work with 3 teams at Sony. The LTH students meet with Sony teams every second week, meaning that they meet each team every sixth week, i.e. they meet team 1 at week 0 and 6, team 2 at week 2 and 8 and team 3 at week 4 and 10, etc. At each meeting we book a usability room (double room separated by a wall with a one-way window), and a number of students will participate in the usability test, while the Scrum team and remaining students will be in the observation room. Every time a student is done with the "test", they will join the people in the observation room to get a view of how this works. After the test (max 1-2 hours), the Scrum team and the LTH students meet for 1/2- 1 hour to discuss and list the findings – a voting will prioritize the findings and the Product Owner will select two items that will be put on the backlog and implemented before the team meets with the LTH students next time.

An additional way to improve the Scrum teams' work and utilize the research from LTH, would be to use the idea from the QUPER model:

The team will together with the Product Owner and LTH define a couple of parameters that they believe are important for their work, i.e. for Social Experience it could be 1) sharing framework, 2) ability to see friends' pictures on the phone, 3) view friends' music, etc. For each parameter, the team and the students will evaluate the solution from Sony and a couple of competitor products (i.e. iPhone, Samsung Galaxy, Nokia etc.), where we can then visually show where Sony are compared to our competitors and discuss what we need to do to improve and become better – the ideas should be described and evaluated (based on complexity and importance), and the Product Owner should select a few ideas to improve our implementation.

It is important that the Product Owner only selects a few ideas each time a list of ideas are presented. It is tempting to select as many as possible, but if you select many ideas, there is a risk that you will not implement any ideas, but if you only select a few, you will almost always manage to do them.

Research project:

1. Literature study (State-of-the-Art): Dec 2012-Feb 2013

- a. Focus on reviewing literature regarding agile methodology within user experience design and product-service system development.
2. Interviews (State-of-Practice): Mar-June 2013
 - a. Focus on understanding current challenges and opportunities on the topic of agile methodology, with a particular emphasis on the suitability of the approach in projects where there is a closer integration between hardware, software and services.

A total of 20 qualitative interviews were performed, with representatives from 19 different companies in various industries.

Reflections on possible improvement:

Pilot project: We have performed three user experience design tests involving both teams from Sony Mobile Communications and students/researchers from LTH (as test persons and observers). It has been difficult to find new participants, so the recruitment process has required more administrative work than we had assumed. at both Sony Mobile Communications and LTH.

Sony Mobile would have liked to run tests every second week to keep to an "agile" pace. However, that has not been possible given the amount of students that have signed up as participants. Although we have combined LTH-internal recruiting with a Facebook group that Sony Mobile Communications is running to find participants. It seems crucial to find a more structured way forward, i.e. by scaling up this kind of testing and introducing it as part of LTH courses within the areas of interaction/UX design, electrical and information technology, etc. (preferably with other companies than Sony Mobile Communications as case companies). This would ensure a continuous flow of new students. Other than that, the tests that have actually been performed have been reported to be very useful for Sony Mobile Communications.

From the LTH side there are some obvious benefits with regard to getting a closer relationship to industry companies and getting feedback on how user experience design plays out in live projects, but there is also an interest to bring the companies into education and research in a more effective way. Today, collaboration is often carried out as short-term or long-term research projects or as short-term student projects. In both situations, companies are often limited to a role as "sponsors" of research and education, and they are "receivers" of results coming from e.g. PhD projects or class projects. Although there are many good examples from industry-driven research projects, e.g. from EU FP7, it would be beneficial to look into ways of stimulating projects (short-term or long-term, small-scale or large-scale) where there is a more equal ownership, and where industry partners are taking a more active role in driving the projects forward. In this particular project, there has been a quite natural division of roles, as Sony Mobile Communications has taken the lead on the pilot project, whereas LTH has taken the lead on the research project.

Research project: The term "pre-study" has been a very fitting title for this project, given the highly explorative nature of the research. First, much of the previous research in the field

has focused on agile methodologies in the software domain, which has made it problematic to scope and carry out a literature review. While there is plenty of work on Agile User Experience Design, our view on "experience design" in the product-service system domain encompasses a broader spectrum of competences than merely programmers and user experience designers. Second, the domain of agile methodologies is not limited to the approaches that use the word "agile". A term like "lean" is often used with very similar connotations. Third, a majority of the informants who do not officially practice agile methodology do not use this kind of language, although I, as a researcher, could clearly understand that there were many similarities between the way they worked – or wished that they were working – and what could easily be considered an agile approach.

In essence, experience design has always been agile, since it has always been iterative and emergent, and it has always focused on collaboration with end users and deep empathy with both users and other disciplines. Also, experience design has been highly responsive to feedback from customers, with no fear of changing paths if user insights point in another direction.

However, experience design is no longer for experience designers only, and it matters little if we call it agile experience design, lean startup, business model generation, or design thinking. The four overarching values from the agile manifesto does not point towards a single, "right" methodology. In fact, they do just the opposite; they point towards using whatever method, toolkit or practice that empowers teams, sees prototypes as resources for learning and communication, develops empathy with customers, users and other stakeholders, and which lets work progress through timely iterations. Thus, similar to what has happened with design thinking methods and practices lately, agile methodology also shows the potential of breaking with its software-only reputation and becoming a low-threshold approach to improve management practices and team behaviors across functional, departmental and corporate boundaries in the product-service system domain.

5. Research Dissemination

Sony Mobile Communications and LTH is co-organizing a workshop session at Innovation in Mind 2013, held in Lund on 18-20 September, 2013. The workshop description is:

Lean + Agile = Innovation

"Faster, better, cheaper." We have heard this well-known mantra from the proponents of lean manufacturing, and we also recognize it from the agile development philosophy that have gained traction in the field of software development. However, making a product "better" by gradually improving the technical performance of individual pieces of hardware or software is no longer sufficient to stay ahead of the competition.

Today, innovation is more about great user experiences than the latest technological advances, and if we aim to deliver such experiences, we must understand how to more effectively work in cross-functional teams that transcend the traditional boundaries between hardware, software and service design.

This workshop will give participants a deeper understanding of how lean and agile approaches can be combined and adapted to more effectively support cross-functional development teams working with product-service system innovations.

All of the participants in the study will receive a copy of each report and paper that follows from this study. Several of the participants have expressed an interest to create a seminar on this topic in their company during the next six months.

References:

- Baines, T.S., Lightfoot, H.W., Evans, S., Neely, A., Greenough, R., Peppard, J., Roy, R., Shehab, E., Braganza, A., Tiwari, A., Alcock, J.R., Angus, J.P., Bastl, M., Cousens, A., Irving, P., Johnson, M., Kingston, J., Lockett, H., Martinez, V., Michele, P., Tranfield, D., Walton, I.M. and Wilson, H. 2007. State-of-the-art in product-service systems. *Journal of Engineering Manufacture*, Vol. 221, Part B, pp. 1543-52.
- 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. & Thomas, D. 2001. *The Agile Manifesto*. Downloaded 2013-06-10 from <http://www.agilemanifesto.org/>
- Cockburn, A. 2007. *Agile software development: the cooperative game*. Boston: Addison-Wesley.
- Comstock, B. & Ries, E. 2012. GE's Beth Comstock in conversation with Eric Ries – Lean Startup Conference. Downloaded 2013-06-10 from <http://www.youtube.com/watch?v=UTXdV1DU0mU>.
- Denning, S. 2012. How Agile can transform manufacturing: the case of Wikispeed. *Strategy & Leadership*, Vol. 40, Issue 6, pp. 22 - 28.
- Ericson, Å. & Larsson, T. 2005. A service perspective on product development - towards functional products. In 12th International Product Development Management Conference, CBS, Copenhagen, Denmark.
- Esque, T. J. 2010. Commitment-based Project Management. Downloaded 2013-06-10 from http://ensemblemc.com/wp-content/uploads/2011/10/2010-Ensemble-paper_Commitment-Based-Project-Management.pdf
- Ferreira, J., Sharp, H. & Robinson, H. 2012. Agile Development and User Experience Design Integration as an On-going Achievement in Practice. *Agile 2012*, 13-17 augusti, Dallas, TX, USA.
- Goldman, S. L., Nagel, R. N., & Preiss, K. 1995. *Agile competitors and virtual organizations: Strategies for enriching the customer*. New York: Van Nostrand Reinhold.
- Highsmith, J. 2002. What Is Agile Software Development? *Crosstalk*, Oct. 2002, pp. 4-9.
- Hussain, Z., Slany, W. & Holzinger, A. 2009. Current State of Agile User-Centered Design: A Survey. In: *HCI and Usability for e-Inclusion*. 5th Symposium of of the Austrian Computer Society, USAB 2009, Lecture Notes in Computer Science LNCS 5889. Springer: New York, NY, USA, pp. 416-427.
- Kidd, P. T. 1994. *Agile manufacturing: Forging new frontiers*. Wokingham: Addison-Wesley.
- Larsson, A. 2005. *Engineering Know-Who: Why Social Connectedness Matters to Global Design Teams*. Doctoral Thesis. Luleå University of Technology, ISSN 1402-1544, ISRN LTU-DT--05/19.

- Leifer, L. J. & Steinert, M. 2011. Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning. *Information, Knowledge, Systems Management* 10(1-4): pp. 151-173.
- Meier, J. D. 2010. *Getting Results the Agile Way: A Personal Results System for Work and Life*. Bellevue: Innovation Playhouse LLC.
- Nagel, R. & Dove, R. 1991. *21st Century Manufacturing Enterprise Strategy: An Industry Led View of Agile Manufacturing*. Iococca Institute, Lehigh University, Bethlehem, PA, USA. Volumes I and II.
- Odiorne, G. S. 1974. *Management and the Activity Trap*. New York: HarperCollins.
- Reinertsen, D. G. 2009. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Redondo Beach: Celeritas Publishing.
- Ries, E. 2011. *The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Business.
- Rittel, H. & Webber, M. 1973. Dilemmas in a General Theory of Planning, *Policy Sciences*, 4, Elsevier, Amsterdam, Netherlands, pp. 155-169.
- Rumsfeld, D. 2002. News transcript, DoD News Briefing – Secretary Rumsfeld and Gen. Myers, Department of Defense, 12 February 2001. Downloaded 2013-06-10 from <http://1.usa.gov/knownknowns>
- Safian, R. 2012. This Is Generation Flux: Meet The Pioneers Of The New (And Chaotic) Frontier Of Business. *Fast Company, Business + Innovation*. Downloaded 2013-06-10 from <http://www.fastcompany.com/1802732/generation-flux-meet-pioneers-new-and-chaotic-frontier-business>
- Savoia, A. 2011. Pretotype It: Make sure you are building the right it, before you build it right. Downloaded 2013-06-10 from <http://www.pretotyping.org>
- Schrader, S., Riggs, W. & Smith, R. 1993. Choice Over Uncertainty and Ambiguity in Technical Problem Solving. *Journal of Engineering and Technology Management* 10: pp. 73-99.
- Shingo, S. 1989, *A Study of the Toyota Production System*, Portland: Productivity Press.
- Simon, H. A. 1969. *The Sciences of the Artificial*, 2nd Edition, MIT Press, Cambridge, MA, USA.
- Sull, D. N. & Spinosa, C. 2007. Promise-based management: The essence of execution. *Harvard Business Review*, April.
- Sy, D. 2007. Adapting usability investigations for agile user-centered design. *Journal of Usability Studies*, 2(3), pp. 112-132.
- Weick, K. E. 1979. *The Social Psychology of Organizing*. Addison-Wesley, Reading, MA.