Redesigning Futurice

Introducing Service Design into an Agile Software Contractor

Master's Thesis
Markus Koljonen

Aalto University
School of Arts, Design and Architecture
Department of Media
New Media
21 March, 2012

AALTO UNIVERSITY
SCHOOL OF ARTS, DESIGN AND ARCHITECTURE
DEPARTMENT OF MEDIA
MA IN NEW MEDIA

Author	Markus Koljonen		
Type	Master's Thesis		
Title	Redesigning Futurice – Introducing Service Design into an Agile Software Contractor		
Date	21 March, 2012		
Language	English	Number of pages	92
Supervisor	Prof. Teemu Leinonen		
Supervisor	Risto Sarvas, PhD		

1 ABSTRACT

Agile software development provides efficient means for the creation of software at low levels of design, but fails to include high-level design into the development process. Service design works at these high levels, but there is unclarity as to how to combine the two.

In this thesis I explore ways to introduce service design into an agile software contractor, taking Futurice Ltd. as an empirical environment. I use co-creative methods to map Futurice's culture and as a solution present three actionable steps: A design team strategy; a novel role of the User Experience Product Owner; and a Service Design Method Toolkit, an artifact-based design heuristic.

Keywords: service design, design strategy, agile, software development

We must move forward, not backward!
Upward, not forward!
And always twirling, twirling, twirling towards freedom!

– Kodos

2 THANKS

Thank you Dr. Risto Sarvas

Thank you Futurice

Thank you all interviewees & workshop participants

Thank you Prof. Teemu Leinonen

Thank you Anttu

Markus

written in Helsinki & New York in August 2011 – March 2012

3 USED ABBREVIATIONS

EBIT	earnings before interest and taxes
FTU	the Facilities, Transformation, and Usage model of service
HCD	human-centered design
HCI	human–computer interaction
IHIP	the Intangibility, Heterogeneity, Inseparability and Perishability model of service
IT	information technology
IxD	interaction design
JIT	just-in-time
NPD	new product design
RND	research and development
ROI	return on investment
RODI	return on design investment
RUP	the Rational Unified Process model
UCD	user-centered design
UI	user interface
UX	user experience
UXPO	user experience product owner

4 TABLE OF CONTENTS

5 INTRODUCTION: ANSWERING CHALLENGES IN AN EXPERIENCE ECONOMY

Pine & Gilmore (1998) have correctly predicted the developed world's progress into an experience economy. In an experience economy, basic human needs of physiology and safety, such as shelter, food, health, and property, are sufficiently satisfied, and value increasingly takes an intangible, personal form, attending to social, moral, esteem-related, and expressive needs (see Maslow, 1943). For enduring quality, products need to be not only physically durable, they also need to be emotionally sustainable (Chapman, 2005).

Many of the major software corporations have been stuck in outdated ways of working, failing to properly integrate design in their strategies and processes, thus falling behind in providing user-oriented value. In development, agile methods have gained popularity and proven efficient by providing a framework for constant learning (Poppendieck and Poppendieck, 2003; Schwaber and Beedle, 2001). From a more design-oriented perspective, Hyysalo (2010) distinguishes between five different levels of design, ranging from detail to social environments. While agile methods provide excellent tools for iterative design work on the lower levels of the systems, they only provide a weak framework on the higher, more radical levels of design thinking.

These changes in the paradigms of and means for creating new products and services underline the importance of design as a discipline as able to focus on essential questions and providing means to answer them in uncertain environments. Experience design (UX) and human-centered design (HCD) offer cost-efficient ways to improve the quality of products and services. The novel field of service design, on the other hand, constantly gains momentum: Holistic and multidisciplinary, it offers means to spot and grasp value on several levels of design (Stickdorn and Schneider, 2010).

Digital products and services compete increasingly in terms of their design: how usable they are, what experiences they provide, and how they bring value to the user. This contradicts with a classical thinking in which technical execution is emphasised as a foundation for providing products and services. However, value created by new increasingly founds on ideas and surprising combinations, technological implementation only allowing for these innovations. In this situation, design skills, design processes, and design thinking step forth as strong value providing factors – especially when seeking high profile, quality solutions.

A collision of the development and design worlds is not only possible, it’s inevitable and already happening. The question is how to do it best. In developing valuable digital products and services superior in usability and user experience it is crucial how service designers, UX designers and usability engineers work together with software developers in development projects. Also, an additional challenge is provided by sales work and the customer interface.

Futurice Ltd. is a Finnish, medium-size software company that utilises agile methods. It has recently started augmenting its design capacity with a UX team and noticed the potential of design integration, but has not yet introduced service design as a prominent part of its offering and customer interface to address all levels of design in projects.

In this thesis, I start off with a literature review on relevant subjects and explore Futurice as an organisation. Based on this, I propose a set of actions for Futurice to more strongly embrace a service design approach in its activity, thus stepping towards becoming a functional design–development hybrid.

6 LITERATURE REVIEW

In this chapter, I bring together existing research and literature over agile development and different fields of design. This is an important step on the path to understanding the fundamental differences as well as similarities between the development and design and to be able to successfully combine them in practice. After an overview on both fields, I inspect the internal conflicts of different design fields, and proceed to considering the opportunities and challenges of a union between design and agile.

6.1 THE MANY FACES OF DESIGN

Despite its established position in society, no agreement has been reached upon the definition of design. In fact, the concepts of design, user experience design, service design, new product design and innovation intertwine and overlap, and in natural language may occasionally be even used to refer to one another. Design as a term has a multilateral nature: It can be viewed from different angles, such as outcome, activity, process, methodology, school, or even an academic paradigm.

In this section, I explore the many faces of design in order to uncover meaningful lenses and touchpoints with agile development, as well as ways to communicate the value of design across disciplines.

6.1.1 Design is Activity and Outcome

Perhaps the simplest way to look at design is to look at its outcomes and what activity produces that outcome. Objects and services are always a result of some design activity, conscious or unconscious (Norman, 1990). A design outcome may be traits such as simplicity, beauty, sustainability, durability, or ethics. It can be a solution to a problem, a proposal for the user, or an experience – or more abstractly, meaning, value, or quality.

In addition to activity that produces outcomes, design can be seen as sub-outcome activity, or processing behind the outcome: as planning, problem sensing and formalisation, anticipation, creation, sense-making, interpretation or redefinition of meaning, or reflection. Some even go so far as to regard all action as having a design aspect to it. (Verganti, 2009; Borja de Mozota, 2003)

Hyysalo (2009) distinguishes between five levels of design of use: detail; user interface; social interaction; product and business concept; and user activity as a whole. Each level requires different types of knowledge, and different working methods. Hyysalo’s taxonomy makes explicit the wide range of issues design activity can address, and the multitude of connections design has with other fields such as business and social sciences.

6.1.2 Design is Process

Some authors (see e.g. Borja de Mozota, 2003; Ulrich and Eppinger, 2007) present the design process as a rational sequence of distinct stages. This model, sometimes dubbed as the stage-gate model, however, does not reflect how, in reality, design work is done iteratively, jumping back and forth between viewpoints (Engwall et al, 2001). An action-centric model comes closer to actual work, highlighting highly contextual decision-making and experimentation.

As Stickdorn and Schneider (2011) summarise, several formulations of such a design process exist. Common to these definitions are their intrinsically iterative nature, motion of divergence and convergence, and a classification of activity into different but interconnected stages, typically three to seven in number. The labels for these stages include identification, building, measurement, insight, idea, prototype, delivery, discovery, definition, learning, looking, asking, trying, and development. Stickdorn and Schneider (2011) propose a four-stage division into exploration, creation, reflection, and implementation (see Figure 1), which I shall use in this thesis.

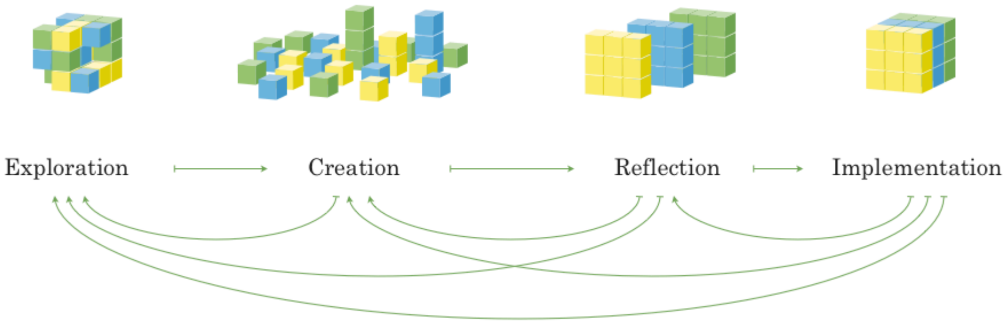


Fig. 1: Stickdorn and Schneider's (2011) four stages of service design (redrawn). Graphics cc-by-sa 3.0 unported Stickdorn and Schneider.

As Sanders & Simons describe (Stickdorn and Schneider, 2011), the format of the result of a design process may be unknown at the start, and defining the format – if any – is part of the design approach. Defining the process itself, then, actually becomes part of the process.

6.1.3 Design is Methodology

The scientific method builds around a core cycle of observations, hypotheses, predictions, and theories. The design paradigm extends this model by introducing elements of creative exploration. Design methodology can be considered to concern the philosophical assumptions and methods that lead to significant and meaningful design (Leinonen, 2010).

It could also be exacerbated that whereas logical thinking is exclusive, design thinking is inclusive (Stickdorn and Schneider, 2011). Design methods operate at a wider scope of design levels (see Hyysalo, 2009) than those used in agile approaches, often enabling serendipitous encounters and revealing unknowns rather than providing guaranteed, directly measurable value.

Design can be said to operate in all of Popper's three worlds: the physical world, the mental world, and the world of objective knowledge. Design may also operate in and mediate between any of Habernas's three interests of knowledge: technological, hermeneutic, and emancipatory. These interest seek to predict and control; understand and interpret; and emancipate and free from wrong knowledge, respectively. Design, then, mediates between worlds and interests. (Leinonen, 2010)

This serves to highlight that, first, design is a fundamentally different concept from business or engineering, and second, an important aspect to design is its capability to mediate between fields. This mediation, however, requires understanding over all of the fields to be mediated.

6.1.4 User Experience Design (UX)

User experience design, or UX, has been defined in various ways (Law et al, 2009). Common terms used in recent definitions include experience or emotion, interaction, perception, and quality. However, the extent to which UX is understood to encompass the design of concepts and whole services varies.

In terms of professional activity and research, UX seems to be commonly understood more mundanely as design work, mostly consisting of usability engineering, interaction design, user interface design, and graphic design, with the aim of providing the user with an usable interface and a pleasurable use experience. In this thesis, by UX I will refer to this rather limited professional definition to best reflect how the term UX is commonly understood and used in everyday communication and language at Futurice and its business environment.

6.1.5 User-Centered Design (UCD) and Usability

The ISO 9241–210 standard (2010) defines usability as “the effectiveness, efficiency, and satisfaction with which specified users can achieve specified goals in particular environments.” Nielsen (1994) extends the term by dissecting effectiveness into learnability, memorability and errors.

User-centered design (UCD), or human-centered design (HCD), is a design methodology in which the needs, wants, and limitations of end users of a product or a service are prioritised throughout the design process. UCD seeks to answer questions such as: Who are the users? What are their tasks and goals? What is their experience? What functions and information do they need? The answers to these central questions then work as sustenance for the iterative design of the product or service. (ISO, 2010)

The ISO 9241–210 standard (2010) summarises the principles of user-centered design process. It urges that there should be an explicit understanding of users, tasks, and environments; users should be involved throughout design and development; design work should be holistic in terms of user experience; and working teams should be multidisciplinary.

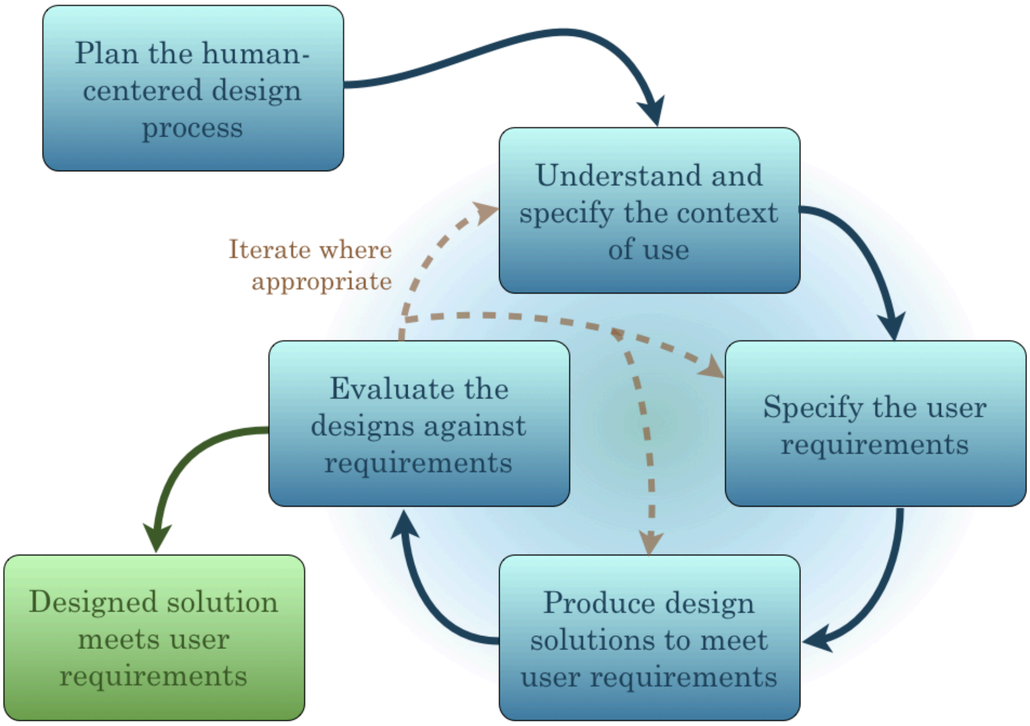

Plan the human-centered design process
Understand and specify the context of use
Iterate where appropriate
Evaluate the designs against requirements
Specify the user requirements
Designed solution meets user requirements
Produce design solutions to meet user requirements

Fig. 2: The ISO 9241–210 user-centered design process (redrawn).

The ISO 9241–210 standard describes the human-centered design process consisting of an initial planning phase followed by four iterative steps (see Figure 2). The initial phase is one of planning the role of usability in the whole process. The following phases proceed from understanding and specifying the context to use, to specifying user requirements, to producing design solutions, to evaluating the design.

Hyysalo (2011) has identified different sources for user knowledge: collecting user requirements; indirect representations such as designers as participants or experts; architectures; cultural maturing; and involving users in development. Different user-centered methods tap into different sources of knowledge, and appropriate methods should be chose sensitive to context, resources, and need (Hyysalo, 2009).

Hyysalo (2009) also presents a general process model for the product development process. This model progresses sequentially from product ideas to concept design, implementation, testing, possible adaptation, and updating. While Hyysalo presents the option of moving back and forth in the process pipeline, it is still relatively sequential (compare to e.g. Figure 1). However, unlike the more limited ISO model, it does address the business level of design.

6.1.6 Service Design

Offerings typically have both a service and a product aspect to them, both of them subject to design. Before looking at service design, the term service should be clarified, as it forms the lens through which design activity is viewed. I will not offer a rigid definition, but perspectives for examining services from.

The IHIP model of services describe services to have four characteristics: Intangibility, Heterogeneity, Inseparability and Perishability (hence the name IHIP). The FTU framework, on the other hand, focuses on Facilities, Transformation, and Usage of two types of resources, those of the customer and those of the provider (likewise resulting in the name FTU). From these established definitions it can be seen that a service is principally seen as abstract and situational, forming a sensible whole, and for the better part evidencing as experience. It also typically takes place in a predefined context, involves integration of resources as result of collateral decision making, and aims to benefit the customer of this activity. (Moeller, 2009)

Stickdorn and Schneider (2011) characterise service design as an approach, a field, and a set of methodologies. Service design incorporates approaches and methods from a wide variety of fields, such as UX, interaction design, UCD, HCI, product design, and graphic design, marketing, psychology, business strategy, operations management, and ethnography to name some. Of different theories, design theory, service dominant logic, activity theory, value chain analysis, and agile development could be mentioned as having impact on service design thinking. Due to its multidisciplinary nature, service design can be viewed as a complement to a set of approaches rather than an established field on its own.

What distinguishes service design from most related fields is its design nature. Whereas many disciplines strive to provide logical tools and solutions to problems, service design embraces exploration and openness to serendipity and surprise, while providing systematic means for identifying problems, research, creation, testing, and implementation.

As an approach, service design aims to not only solve known problems, but to discover the most important problems to design for. Complex social problems are a focal point in service design. To take on this challenge, service design combines a plethora of disciplines to gather a holistic understanding of a system and its weaknesses. This holistic approach differentiates service design from many of the fields it incorporates – notably, in the context of this thesis, agile development. (Stickdorn and Schneider, 2011)

Stickdorn and Schneider (2011) describe service design as having a variety of goals and outcomes. Common expressed aims for service design include better services and interfaces, creation of experiences, and expectation management. The outcomes of a service design project may take various forms, depending on what is deemed central in terms of providing value. The outcome may be a combination of abstract organisational structures, operation systems and processes, service design experiences, concrete physical objects, or a combination of these.

To provide something actionable, different methods from adjacent fields are adopted and applied to design services. These service design methods can be loosely aligned along four dimensions (Tassi, 2009): activities, representations, recipients, and contents. This taxonomy underlines the fact that design methods operate on various different levels of design, address different types of knowledge and questions, and mediate between different parties. The methods also vary according to which of the four phases of design they are best suited for.

Stickdorn and Schneider (2011) present service design as having five core principles: service design is user oriented, co-creative, sequencing, evidencing, and holistic. Of these, sequencing refers to the sequential, dividable nature of services. The different stages of a service combine a whole that needs another kind of lens – a holistic one. Evidencing refers primarily to the physical dimension of services.

There have been case studies and research on the effectivity of service design tools in projects, no academic meta-analysis on service design thinking has been conducted as of yet (Stickdorn and Schneider, 2011).

This lack of evidence may render the approach challenging to communicate in corporate environments. Furthermore, there has been little explicit interaction between the fields of HCI and service design (Wild, 2008).

6.1.7 RODI: Design Pays Back

The benefits of design need to be formulated in order to communicate its worthiness to other fields. The benefits design brings may be classified in several different ways. Design can be said to have “hard” and “soft” benefits, referring to how easy they are to measure and how quickly they emerge. Return on investment (ROI) is a typical hard indicator for to what extent an investment will pay itself back and in what, supporting decision making. Return on design investment (RODI), not surprisingly, specifically measures the profitability of design efforts.

In the following, I will summarise some of the potential benefits, both hard and soft, based on Borja de Mozota (2003) and Schaffer (2004). I distinguish between internal benefits, or benefits that fall on the organisation using design, and external benefits, or benefits that fall on the customer or user of the service or product.

Internally, design improves the learning process, reduces design cycles, helps avoid building unnecessary functions, and expedites decision making. On a softer side, design brings about an user oriented philosophy, and fuels innovation through creative methods and enabling communication between different fields.

Externally, design improves communication with and between stakeholders, identifies potential segments & niches, reduces time to market, drives user orientedness, strategically differentiates products and services, allows to adapt to changing needs, and increases the overall quality of the product or service, driving sales. Lack of design may form a bottleneck, strongly restricting value provided by other functions. Lack of up-front design may also result in design debt, making later changes exponentially more resource-intensive to implement. Design reduces risk by answering known questions and unveiling unknown ones, helping avoid disasters.

Softer external benefits include improved means for finding contextual opportunities, overall external relationship, and sources of value.

More specifically, according to Nielsen (2008), a 10 % investment on usability engineering provides an improvement of 83 % in a website's business.

6.1.8 Design Strategy

Above, I have presented a brief view on the benefits of design. The question arises: How to introduce design into organisations that are founded on different paradigms? Best (2006) describes design strategy as the "effective allocation and coordination of design resources and activities to accomplish a firm's objectives of creating its identities, environments and offerings", which involves identifying and ceasing opportunities for valuable design contribution. (Note that this definition is quite distinct from the idea traditional business strategy that typically consists of values, mission, and vision.)

Borja de Mozota (2003) proposes that design be introduced into organisations gradually, responsibly, and deliberately. This requires learning from successive design projects, focusing design decision-making, design management on all levels, and management support.

Best (2006) suggests that design strategy be established by first describing the need, then developing proposals. The need should be communicated carefully, communicating with stakeholders to understand their various backgrounds and world views and conceiving opportunities for design, for example using different types of analysis, research, interviews, and scenario planning. Implementation can be done using tools, workshops, and guidelines.

Both Best and Borja de Mozota view design strategy as encompassing several organisational levels from operational to organisational and strategic. Schaffer (2004) talks about institutionalising usability in organisations, promoting a mindset shift from a typical functionality heavy to a progressive user-centered. To achieve this shift, he presents ways from hands-on design work to operations management, including trainings, staffing, processes, methods, and organisational restructuring. He also mentions executive insights, major failures, and exploration as possible triggers to a cycle of usability improvement. Schaffer emphasises that standards of any kind shoud not be developed unless clear plans and resources are in place for disseminating, supporting, and enforcing them.

All of the forementioned authors seem to lean on hierarchical organisational structures and position-based decision-making, not considering agile, low-hierarchy organisations such as Futurice. Furthermore, their discussion seems to be focused on physical products and original product development instead of contractor work and the design of services. While they may not be directly applicable in the context of Futurice or other agile and lean contractor organisations, lacking anything better, their points may form a noteworthy platform for design strategy work if applied critically.

6.2 AGILE SOFTWARE DEVELOPMENT

In this chapter, I will briefly describe lean and agile and some of their methods to shed light on the touchpoints between agile software development and design. I then proceed to connect agile with design and look for gaps.

6.2.1 Lean

Lean is an approach, a philosophy, or a toolset to organisational thinking. It provides a way to specify value, a sequence for value creation, and a way to conduct this sequence. Originating in car manufacturing industry, it views the organisation fundamentally as a production line, the product of which is value. Lean's other core concepts are waste, or any work that does not produce value; flow, or the seamless production of value; pull production, or listening to the next steps of a production line for demand; and *kaizen*, or incrementally reaching for perfection. Also, *kaikaku*, or radical improvement, may take place to complete realign a value stream. (Nicholas and McGraw-Hill, 1998; Womack and Jones, 2003)

Poppendieck and Poppendieck (2003) have formulated a framework for applying lean to software development that can be summarised in a set of principles: Eliminate waste, amplify learning, decide late, deliver as soon as possible, empower the team, and build integrity in. Building software is considered a learning process that can be nourished with feedback, iteration, and evaluating past experiences. Decision-making is postponed as far as possible to keep the door open for late changes and ensure trustworthiness of decisions. Quick delivery encourages feedback and learning from concrete results. Teams take responsibility of the quality of the project, and strive for an easily maintainable product. Also, a holistic viewpoint and a shared understanding of the quality of the product is promoted, allowing members to step out of their professional roles.

6.2.2 Agile

Agile is a group of software development methodologies, originating in the 2001 Agile Manifesto (Beck et al., 2001), a set of development principles established in search of alternatives for the mainstream waterfall process. In agile, the focal points are people, software, collaboration, and change, and it discourages documentation, fixed contracts, and accurate plans. Changes in requirements are welcomed as the outcome becomes more tangible along the process.

Value in agile is defined as the satisfaction of the customer. Satisfaction is created by continuously delivering working software, which in itself is the measure of progress. Projects are self-organising, built around motivation and empowered with trust and support. The whole project team is responsible for the end product. Seamless work is ensured with continuous cooperation between functions.

Agile can easily be misunderstood as a pretext for a laissez-faire policy, where best practices are rejected with the excuse of being too cumbersome, and autonomous teams end up self-organising haphazardly. Agile methods can, however, turn out to be quite intricate and as such, are not implement without effort.

I will give an overview on two agile methodologies, Scrum and Extreme Programming (XP), common at Futurice and around the world. Another technique, Kanban, is being experimented at Futurice as of this writing, but I shall leave it, along with other agile methods, out of the scope of this thesis.

6.2.2.1 Scrum

Scrum is a widely used agile methodology for organising work. It features three roles: Work is carried out by fully autonomous, cross-functional scrum *teams*, which are facilitated by *Scrum Masters*. In addition, a *Product Owner* represents the interests of the customer, providing product requirements.

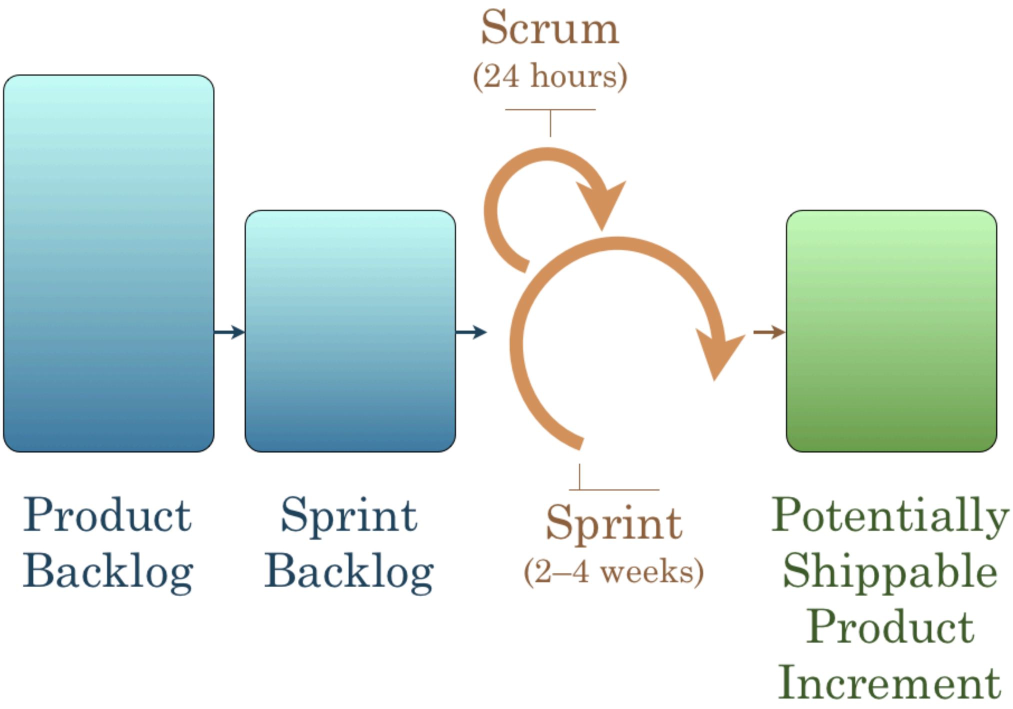


Fig. 3: The Scrum model (redrawn from scrumalliance.org).

Scrum involves short development cycles called *sprints*, typically two to four weeks each, and daily status meetings called *Scrums*. During a sprint the scrum team has daily standing meetings, or Scrums, where every team member shares what they have done yesterday, what they will do today and if there are impediments that would obstruct their work. The Scrum Master helps eliminate the obstructions and evaluates whether the Scrum process is followed. Scrum Master should have no formal authority or decision making role, but rather that of a servant leader, aiming to empower team members to be able to learn, grow, and remove obstacles on their own. The role of the Product Owner in Scrum is to decide and communicate a product vision based on stakeholder interests. (Schwaber & Beedle, 2001)

Development tasks are listed in a *product backlog* that withholds all features and functionalities the developed system is visioned to have. These are presented in the form of user stories and prioritized by selected criteria. The backlog evolves throughout the project as deliverable requirements change. For each sprint, a *sprint backlog* is assembled of product backlog items. The scrum team is responsible for estimating the work, deciding what can be done during each sprint, and executing against that. (Budwig, Jeong, and Kelkar, 2009; Schwaber & Beedle, 2001)

Scrum differentiates between the design and implementation of the user interface and the meta-design of product requirements as different layers of decision-making. The team operates on the UI layer, and product owner on the business logic layer. (Deemer et al., 2010)

Each sprint produces a potentially shippable product which acts as a reality check and exposes dysfunctional constraints. (Deemer et al., 2010) A sprint review is held after each sprint to go through what was done in the last sprint and what could be done better in the next one. After this, either the product is finished or the team plans the next sprint. (Schwaber & Beedle, 2001)

6.2.2.2 Extreme Programming

Extreme Programming, or XP, is an agile method somewhat similar to Scrum. The project is carried out in two-week iterations, with a number of customer-selected user stories developed during each. A responsible similar to Scrum's product owner acts as a spokesperson for the end user. The customer provides requirements and feedback about the project, leading to changes in the next iterations' requirements and user stories. XP highlights just-in-time (JIT) decision making as requirements are not done up front but in parallel with implementation. As changes to requirements are to be expected, long period of gathering requirements is seen as waste. (Martin, 2002)

Specific agile tools and techniques, such as burndown charts, pair programming, and test-driven development, are typical to different agile methodologies, but I won't address them more closely in this thesis.

6.3 DISCUSSION: COMBINING AGILE AND DESIGN

The design approach has not yet been widely connected to agile software development; only recently have tentative attemps been made. The majority of existing design research seems to concentrate on the limited sphere of physical product development and big hierarchical waterfall organisations, contradicting with the fuzzy service design approach and the iterative agile approach.

Next, I compare the different fields of design to one another as well as to lean and agile, looking for similarities, dissimilarities, and gaps of knowledge.

6.3.1 Different Design Fields Conflict

Comparing the UCD process with that of service design (exploration, creation, reflection, and implementation), many views of UCD seem to lack the elements of creation. And indeed, UCD provides little tools for creating novel solutions, settling for spreading established best practices and utilising measured results.

Most of the time UCD also seems to assume that the design question be known already at the beginning of the design process. The ISO model takes the product or service concepts as a given. To put this in the context of Hyysalo's (2010) levels of design, UCD doesn't address the highest levels of design from the customer's perspective. Hyysalo's own model takes "product ideas" or "possibilities" as its starting point, restricting creative action in the start of a product project. Service design contradicts these approaches by incorporating a persisting criticism towards the design problem itself into its process and approach.

What differentiates service design from UCD and UX is its embedded urge to reflectively choose its communication and collaboration partners: Unlike service design, UCD and UX provide little means to communicate with a business customer, or combine user knowledge with business requirements. It has not been needlessly suggested that stronger communication between the fields of HCI and service design would benefit both as even this intra-design landscape seems uncharted (Giusti and Zancanaro, 2009).

6.3.2 Agile user-centered design

It has been acknowledged that even professional UCD practicioners rarely complete repeatable and systematic methodologies (Schaffer, 2004). This may be because of the complexity of required such minimum processes and the difficulty to follow them on one hand, and the incompatibility of such rigid processes to any organisational environment on the other. This complexity is highlighted when UCD should work as part of an agile development process. While a user-centered philosophy is common to both UCD and agile methods, the two approach it with different principles and rhytm.

Wärn (2010) rightly notes that agile methods, by themselves, lack means for ensuring usability of products. On the other hand, some cases report efforts to combine agile development methods with user-centered design. Several user experience professionals have pointed out that the old holistic way of designing software before even starting implementation is not optimal, especially in agile work environments, as a separate up-front design phase would take a significant portion of the time available to get the whole project done (Budwig, Jeong, and Kelkar, 2009; Gosper, Agathos, Rutter, and Coatta, 2011) – though Gosper et al. (2011) suggest, contrary to agile, that a full set of agreed-upon user stories should be ready before development and design can work in parallel.

The demand for a natural design–development parallel seems justified. Chamberlain et al. (2006) pave this road by suggesting that UX work should be integrated with agile processes by involving users; having designers, developers, and customers work together closely; synchronising production and testing of prototypes with development; allowing for up front design to formulate the initial set of requirements; and providing a cohesive project management framework for this integration.

Attempts in that direction have been made. For example, Budwig et al. (2009) suggest a separate design scrum team with its own product backlog and product owner, and would have the design team working one or two sprints ahead of the development scrum team. This approach, however, assumes a UX scrum team of several people. Patton (2008) presents the idea of having a designer attending the Product Owner role of an agile project to best make design decisions.

It should be noted that most of the forementioned cases take place in heavy organisations with big teams, and may not be applicable in middle-size companies with considerably smaller project teams. The discussion over best practices for integrating UX and development in a middle-sized software company is ongoing and likely to strengthen, and introducing service design in the mix will make it more complex still.

6.3.3 Agile and Service Design

There are several similarities between agile and service design: Agile methods are presented as appropriate techniques in uncertain environments, utilising built-in process mechanics to create understanding over the use of the product through iteratively developing an end product. Scrum and XP both drive user-centered thinking by revolving around user stories. Working iteratively and using a software prototype as hypothesis are also very design-like approaches (see Leinonen, 2010). While the rational model of design seems to align with the waterfall process of software development, the action-centric model of design has more in common with agile ways of working.

However, while agile presents a variety of techniques to bridge the chasm between a user story and its implementation, it does assume that the product owner has perfect or sufficient understanding of the users and the business. Agile provides ways of efficiently delivering something that has been externally defined, but it leans on discovering value through incremental trial and error. The process provides no specific means for generating these user stories, accessing user understanding or reaching understanding over a customer's business needs, nor do they take into account that user knowledge is transformed through iterative implementation and often discovered serendipitously (Hyysalo, 2009). In other words, agile methods focus on Hyysalo's (2009) system, user interface, and detail levels of design, providing little support for addressing the high design levels of society, business and social context.

Several challenges in combining these approaches exist. The waste principle, central in lean, is problematic in terms of design, because design has difficulties with quantifiably justifying its creative and serendipity-directed yearnings. As many design methods lean on experimentation in discovering unknown unknowns (instead of just making known unknowns known), they are easy to interpret as waste, and, as such, drop out of the process.

To start a development project, some definition of the product to be developed must be defined at any rate. Agile development may be one feasible way to build understanding over the requirements. However, more important than the development process itself is the level of fidelity. Agile development is easily understood as starting from producing code, while the right fidelity to start with might be for example paper prototyping, or business modeling.

Looking at the agile approach from the perspective of service design's four phases (see Figure 1), exploration and creation are completely neglected, or handed over to the product owner role, and the development team activity focuses strongly on implementation and, partially, reflection over that implementation. Introducing the first two phases more strongly and integrating them with the rest of the process, then, could provide surprising value.

To summarise, research over combining service design with agile development seems scarce. This merger is made more difficult by two things: Different design fields contradicting to some extent; and agile methodology partly conflicting with service design thinking. Simultaneously, however, strong similarities exist, giving hope of these fields being able to formidably enrich each other.

7 DESIGN CHALLENGE: INTRODUCING SERVICE DESIGN INTO AN AGILE SOFTWARE CONTRACTOR

Above, I have reviewed current literature, research, and discussion over different design and agile topics. It seems evident that little experience and research exists over combining service design with agile development, therefore providing fertile ground for both further research and practical experimentation.

In this thesis, my design challenge is: **How to introduce service design methods into Futurice, an agile software contractor?** More specifically, what steps should be taken to bring about this introduction? And more broadly, how to introduce service design methods into any agile software contractor?

In order to answer this challenge, I follow Best (2004) and Borja de Mozota's (2003) loose guidelines for the organisational introduction of service design as presented by Stickdorn and Schneider (2011). I look for frameworks for Futurice to create value in software projects using service design methods, reflecting on and learning from cases in which UX has been combined with agile development. I make informed guesses (see Kurvinen, 2007), striving for actionable and practical steps that reflect Futurice's culture on the long and uncharted path of becoming a hybrid design–software company. To conclude, I evaluate the proposed actions and consider how they can be applied in contexts other than Futurice.

To maintain a focus tackling this multidisciplinary and multilayered challenge, I will have to leave some interesting aspects outside the scope of this thesis. I will not directly address Futurice's business strategy, organisation structure, skill pool or individuals' skills, recruitment practices, sales, or leadership – all of which would be reasonable lenses for the challenge of design strategy. Neither will I address the feasibility of individual design methods, nor their compatibility with agile software development processes.

8 METHODOLOGY

In order to come up with means to introduce service design into Futurice, a thorough understanding of the company culture – attitudes, knowledge, and ways of working – was required. Much of this information is tacit, invisible and difficult to grasp, but essential in order to determine the best ways to introduce service design into the organisation. To gain a holistic understanding over this complex issue, I utilised a variety of methods, many of them user-centered and discussion-based.

I approach the company culture through Rousseau's (1995) Layers Culture and Normative Contract model, which describes culture to construct in several different layers: fundamental assumptions, values, behavioral norms, patterns of behaviour, and artifacts. I built the ethnographic research on my own experience as an employee of the company. I started work in May 2011 and went through the company's introductory program, including training sessions and interviews. I have since participated in several projects. I analysed a set of company artifacts, utilising previous efforts to make tacit knowledge explicit and gain a view over the company's commonly accepted values and aspects deemed central.

I conducted interviews, focus group discussions, and workshops, including people from different functions. I created a set of key questions to support the discussions, keeping them thematic to allow for taking surprising paths. In these discussions, I brought up issues and research I've outlined in my literature review above. At some of these discussions, I created visualisations to work as boundary objects over relevant aspects of Futurice, aiming to challenge and validate assumptions, focus and incite cross-disciplinary discussion, trigger a priori mental models, and help reveal unknown issues.

The interviews and workshops took many forms – in fact, it's safe to say that each of them was different, as I modified the (loose) template between every session. Typical themes revolved around the interviewee or workshop target group's role: "How would you define design?" "What's the Futurice project model like in your eyes?" "What does sales mean to you?" In addition to answering preformulated questions, I expected these methods to reveal surprising elements about the company culture.

Between discussions and workshops, I formulated basic models for the measures for service design introduction, utilising and adapting practices from theory and cases from the fields of design strategy and agile user experience design to Futurice's specific cultural context. These models were then iteratively and co-creatively assessed and developed further, finally taking the form of practices and physical artifacts at the UX team's working space.

My pre-theory is that in Futurice's projects, high level design, or meta-design, is mostly unstructured and does not utilise design expertise nor design methods.

8.1 POSSIBLE SHORTCOMINGS

At Futurice, being a UX team member, I'm viewed as a designer despite my background in language technology and computer science, which may affect how I am viewed as member of the community. On the other hand, being a designer, I may have interpreted my environment from a design-centered perspective. Both of these issues may have affected the ethnographic research conducted.

An alternative approach for this co-creative, discutative approach would have been to gather more quantitative data, collecting and quantifying elements from past projects. While it could have more specifically revealed shortcomings in current practices, such an approach would, however, be more limited in creating new solutions bottom-up that would fit the culture.

Also, regarding the discutative approach I took, I opted for quantity instead of scientific rigour, aiming for several cycles of interaction that build on each other, instead of for example meticulously litterating and analysing individual interviews. This approach, while surely providing a better subjective cultural understanding, may prove to stagger before positivistic methodological criticism.

To monitor the effectiveness of the proposed measures, a longer iterative period of alterating between practical use and redesign would have been ideal, but given the timeframe for this study, only one evolutive iteration could be incorporated.

9 FUTURICE: MAPPING THE ORGANISATION CULTURE

In this chapter, I explore Futurice as a company, basing on my personal experience as a Futurice employee, company artifacts, interviews, focus group discussions, observation, and workshops. Conducted observations, interviews, group discussions and workshops are listed in Appendix 1.

I evaluate Futurice mainly from the viewpoint of design utilisation, seeking for and pointing out room for improvement. This task turned out surprisingly challening in a middle-size, low-hierarchy organisation, in which knowledge is abundant but fragmented and, at times, contradicting. Typically, the interviews took their own course, revealing new insights into the culture and ways of working of the company.

To facilitate discussion, I produced two visualisations: a descriptive project life cycle diagram, and a UX-centered organisation map. I provided the visualisations in a very crude, hand-drawn format to communicate the indecisiveness of the models and as such encourage discussion, participation, and modification. These models served as useful boundary objects for communicating what role design functions have in projects and where any bottlenecks for design to provide value may lie.

9.1 OVERVIEW

Futurice Ltd is a Finnish medium-size software company. Founded in 2000 by a handful of technology students, the company initially worked on mobile technologies. Later, Futurice built Koneboxi, a photograph sharing website similar to Flickr. Koneboxi was later sold as the company reached a pivot point and started focusing completely on contractor business.

During the past years, Futurice has been growing strongly both in terms of figures and personnel. As of this writing, the company employs 130 people. In 2010, turnover was at 9,5 million €, and a current estimate for 2011 is 12 million €, keeping growth at 26 %, the forecast for 2012 matching at 25 %.

Futurice has offices in Helsinki, Tampere and Berlin, with circa 90, 20, and 20 employees respectively. As of this writing, a fourth office is tentatively being set up in London. It is worth noting that being younger, smaller and differently balanced, the Tampere and Berlin offices operate differently from the Helsinki headquarters. In this thesis, I mainly focus on the Helsinki office. There, alongside several development business teams, there is an inhouse UX team, founded in 2009.

Futurice describes itself as agile and lean, constructing its organisation around their central concepts such as empowerment, transparency, and elimination of waste. There is a relatively low hierarchical structure, with regular employees, project managers, and team leaders forming roughly three distinct levels of hierarchy.

Futurice provides its customers with software on demand, delivered in the form of projects as a service in collaboration with the customer.

Futurice's core expertise and projects revolve around mobile and web technologies and solutions. Being a contractor, the ways Futurice's project teams work are strongly dependent on customers' ways of working. Customer needs, and projects along them, vary from clearly defined, well understood problems to complex, open questions. Value is thus created in a variety of ways ranging from efficient production to creation of novel concepts. Typically, a new customership is started with a small project to build trust, and bigger projects are built on that.

9.2 COMPETITION

Futurice's closest competitors are other midsize software companies, especially Finnish ones such as Reaktor and Luxus, with which Futurice competes in efficiency, quality, trustworthiness, and customer satisfaction. Big software companies such as Accenture, Tieto, and Logica are also competitors, although less direct, as they orient towards bigger projects with a strong waterfall way of working.

Somewhat surprisingly, creative houses, such as Fjord, Idean, Palmu, and Nordkapp, also form close competition. Customers often prioritise more pure creative agencies over Futurice when it comes to design. Futurice may still be considered as a design partner, as having technical competence under the same roof and having to manage a smaller network of subcontractors is clearly seen as a benefit. However, creative agencies drive past in design expertise, design portfolio, and design product offering; Futurice's development company image may render it difficult to view it as the best design partner; and a lack of deep design relationships with the client all work for the benefit of competitors.

9.3 STRATEGY

An important perspective to leadership and culture is what is focused on and measured. At Futurice, a principal focus is on EBIT (earnings before interest and taxes) to measure overall company performance, and as EBIT's subfactor, employee utilisation rate, describing the share of billable hours of employees and thus seen as a reliable measure of individual profitability. Overall employee satisfaction and customer satisfaction are also regularly measured. At weekly meetings, teams also follow members' project progress and billability, as well as workload, availability, mood, and satisfaction.

In a strategy workshop, the company has formulated goals for 2015. Numeric indicators for these include 300 employees, 3000 followers on Facebook and Twitter, and 10 key accounts. Five top goals were recognised as: reaching an ultimate way to do software; structurising sales; forming a systematic cycle of improvement; accelerating company-wide learning; and increasing Quality Assurance inclusion to projects to provide value.

While service design and UX were not included in the five top goals, UX was recognised as a key value-adding function. Moreover, the goal of 10 key accounts demand a deepening of existing and future customer relationships into strategic partnerships and promotion of communication heavy ways of working, inviting a service design approach and methods. Furthermore, service design may play a central role in structurising sales, and the introduction of service design may enhance the communication capabilities of software development work at Futurice.

9.4 CULTURE

By culture, I refer to the shared attitudes, values, goals, and practices that characterise Futurice and define patterns of action. Understanding Futurice's culture is important for evaluating how a measure proposition would be received. In mapping this cultural knowledge, surprising and unforeseen factors were discovered.

Great Place to Work Institute listed Futurice as second best place to work in Finland in 2011, and 17th in Europe, a recognition strongly based on employee satisfaction, in which culture plays a key role. Futurice's commitment to agile and lean principles clearly reflect in its cultural foundations. At a strategy workshop, the Futurice culture was crystallised as ideally being open, transparent, trusting, empowering, client-centered, and people-centered.

At Futurice, imposing rigid processes is generally shunned as teams are believed to be able to find their best ways to work together on their own, and learn from the process. There is a number of exceptions however, increasing as the company grows in size. New employees go through an introduction period consisting of trainings and interviews. Most teams have weekly meetings in a specific format, and regular mentoring and development discussions take place.

Employees are encouraged to learn. A montly FutuFriday event features volunteer presentations and workshops that address anything employees wish to bring up and learn. A culture of giving and receiving feedback is encouraged and maintained through presentations, stickers and posters. Feedback is used to reward with recognition, and as a means for learning. The UX team, for instance, has quick weekly sessions to evaluate and comment others' designs. Also, a culture of understanding the whole is enforced by encouraging employees to “ask why.”

Transparency is a principle that’s often brought up to guide decision-making. Weekly financial meetings addressing company status are open to all employees, and the same numbers are mailed to all and shown on corridor screens.

Futurice employees are empowered to take as much responsibility as they want and need to get things done. Employees are granted the power to make the decisions that most affect themselves, and remove any obstacles on their path without external assistance. For instance, all permanent employees are equipped with a company credit card and permitted to use common sense to purchase anything they need to get their work done. While this power nourishes ownership, it also comes with responsibility to weigh decisions against the company’s 3 × 2 philosophy: When making decisions, employees are expected to balance the interests of the *customer*, the *people* working at Futurice, and the company *numbers* (3), both *now* and in the *long term* (2).

Futurice’s culture is shown and constructed by visible artefacts the company produces. These artefacts include posters, documentation, and training materials, but also more ephemeral elements, such as room names.

Plenty of posters are used to communicate and remind of commonly approved best practices, often conveyed in a list format, and sometimes giving concrete examples. The selection of the topics for these posters may be understood as important, commonly accepted factors of what defines good work. Topics include listening, lean principles, customer relationship maintenance, and backend setup process. Apart from one-way information channels, walls are used also as co-creation forums. Many of the Helsinki office rooms are named after Helsinki's Linnanmäki Amusement Park's recreational devices, implying that work should be fun.

9.5 PROJECT ANALYTICS

According to client feedback from 21 Futurice projects, successful projects commonly have the following properties: The customer understands technology (71% success rate), the project is of large size (72%), Futurice and the customer have a common expectation of the project (69%), the schedule is relaxed (100%), and the project had no unknowns (83%).

38% of the projects have failed, however. Top reasons for failures are as follows: The project contract is fixed price (70% failure rate) ; the project is small in size (60%); and the expectations Futurice and the customer turn out to differ from one another (66%).

While design isn't featured prominently in either direction, introducing design as an intermetiading element can provide means for clarifying unknowns and sharing expectations and thus help projects succeed.

The failures imply that in design cases it's crucial that the customer understand design. Achieving this understanding presents a challenge as design professionals in leading and sales positions are scarce. A strong design understanding would also be required on the part of Futurice's sales to communciate design value.

9.6 VALUE PROPOSITION

Futurice makes effort in branding itself not exactly as a software house, but a partner with an emphasis on speed and quality boosting functions. This shows in the company's slogans "We make it easy, we make it happen."; "A software development boutique, not a code factory."; "High-end software boutique"; and "Your rapid development partner."

Though software development is Futurice's core activity, the company's external communication shows an intended shift in emphasis towards strategic partnerships, service business and holistic design thinking. The UX element is prominently featured on the company web page and sales presentations. On the company website, the communication emphasis is on concepting, design, and consultation. Noticeably Futurice's offerings consist of services and expertise that "help develop your digital business." In an offering visualisation, software development is presented as a particle of "digital strategy," which consists of several parts (see Figure 4), the other parts being, as of October 2011, digital strategy, concepting, service design, UX design, and lifecycle. (Note: This offering has been modified as of February, 2012.) UX design is presented as separate from strategy, concepting, and service design, taking place after them and before software development.

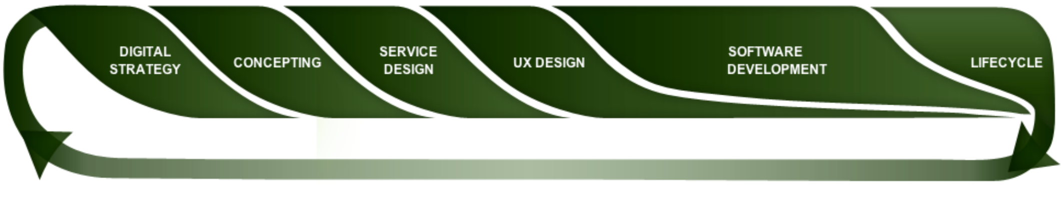


Fig. 4: Futurice's offering as of October, 2011, as presented on futurice.com.

Despite this communication focus, in an interview, a sales person described customers as “typically seeing Futurice as a 150-man digital sweatshop,” referring to strong emphasis on development, which UX may or may not be a mechanical part of.

In practice, proposals are negotiated with the client. Typically projects revolve around software development projects, possibly with added work, such as UX design. Recently, however, there have been some pure service design opportunities, reflecting Futurice's growing expertise and experience in the field as well as shifting emphasis more strongly towards UX work in sales.

Sales-wise there is a big challenge of Futurice lacking actual service design products, service design professionals, experience in service design methods, and a portfolio of reference cases to demonstrate service design expertise.

9.7 SALES & CUSTOMER RELATIONS

At Futurice, sales people are also technological experts. Interviewed sales representatives strongly agreed on was that trust is a principal element on several levels: in deepening a strategic relationship, being able to understand customer’s business, and selling work in a format that provides best value. The customer relationship was seen as even mutually therapeutical, helping both sides understand their capabilities and business better. To achieve this depth, deep mutual trust is required as perhaps the most important single defining factor of a customer relationship.

In interviews, contradicting views on sales principles were expressed. Others view optimal sales as activity that prioritises the customer’s good over the company’s own profit, e.g. offering the best solution even though Futurice would not be directly involved, describing it as “selling with integrity.” A project manager, however, saw that “best value is brought to the client by maximising Futurice’s own benefit first.”

Both views however recognised that unknowns can damage partnerships. They both rely on the view that sales is a service profession that aims to understand the customer’s problems and business and as such create value by itself.

In building trust with the customer, both sales people and customers identified past work, open communication, and questioning of assumptions. In all these aspects, design presents gamechanging tools.

Customer attitudes towards design work vary greatly, so any frameworks for supporting design sales would have to be situationally adaptable. Deeper trust also increases capability to include design work in projects more strongly.

Recently, Futurice has taken first steps in introducing strategic selling methods that focus customer communication with key questions and seek to acknowledge assumptions and unknowns.

9.7.1 UX sales

Currently, several factors obstruct the utilisation of value-adding design methods in service projects. Perhaps the most crucial of these is a lack of design knowledge and understanding amongst other functions than the UX team. Chiefly, the understanding the customer has over design strongly defines the nature of the project as a customer only buys what they deem valuable. Even in major IT companies the value of design may not be recognised, and a feature heavy approach remains the principal paradigm, showing in outsourcing as well.

Another problematic resides in a company’s sales personnel, as they form the main channel between the company and the customer especially in the formulation of projects, typically defining project scopes, timelines, deliverables, methods used, and skills required. Furthermore, design work may be restricted by the limits of design knowledge and understanding within the project team.

A major obstruction to including a sufficiently strong design approach into a project is a lack of proof over its return on investment. Reference work may be used to communicate examples of design impact, but numeric data over design’s impact on the overall quality of the resulting product or service is lacking.

Lastly, resource limitations restrict design work. Typically this means the amount of time that can be used on design, though it is also possible that at a given point in time, design demand over a specific field, such as graphic design, exceeds the amount of design expertise available, such as graphic designers.

9.8 AGILE PROJECTS

Work at Futurice is organised in small customer project teams, typically operating under business teams. An anomaly in this setting is the UX team, from which UX professionals, each working on one or two projects at a time, are requested to join projects. The project teams are gathered in an ad hoc manner in the offering stage by sales representatives, depending on availability and chance. In this sense, the UX team is a reactive resource pool, not a proactive change agent.

While Futurice prizes itself as an agile and lean company, project teams normally only utilise parts of agile methods, rarely doing it by the book. For example, a project team might use a Scrum table to visualise project flow and make decision making concrete, but lack a Scrum Master or sprint reviews, or alter their implementation, often discarding many of Scrum's benefits in the process (Deemer et al., 2010). Some agile practices, such as pair programming, are utilised, while others, such as test-driven development, are uncommon, resulting in a testing heavy period towards the end of the project. What results is that descriptively projects take waterfall form both within an iteration and in terms of the big picture of the process.

In a workshop, it turned out that the people in charge of developing practices at Futurice emphasised defining, finishing, and locking in place sub-wholes of a project within a sprint. While the goal of implementing a potentially shippable product aligns with Scrum, viewing the implemented parts as hopefully immutable differs from it (Deemer et al., 2010).

Additionally, many of Futurice's business cases may start off as pure concept design with a 'go – no go' treshold before any development takes place. This means that Futurice would benefit of ways for competing in the field of pure service design.

9.9 UX AT FUTURICE

There is already a generally positive attitude towards design functions at Futurice. Futurice leaders talk of the company as a representative of a "New School" of building usable, experience-rich, valuable software user first. In the company's sales materials, UX is described as "baked into development”, and there is a soft internal recommendation of including 10 % of UX work in projects, along the lines of Nielsen (2008).

While, in interviews, the value of UX was recognised both in terms of customer satisfaction and end product quality, how UX is understood at Futurice varies however depending on person and context. In projects, some project managers saw UX primarily as building wireframes and beautifying the user interface, sometimes jokingly refered to as “pixie dust.” Many interviewees regretted having no accurate conception on what design work or design training consists of.

Despite its recognised benefits, a general experience was that including UX in sales propositions paradoxically resulted in a lower rate of deals. On the other hand, it was partly believed that quality Futurice is often selected as partner largely because of quality UX references.

9.9.1 UX Method Tool

UX professionals typically work as the only UX responsible within a project team. Combined with a lack of processes, the UX professionals tend to have their own ways of working that relies mostly on a “genius design” mentality of creating user interfaces with little external input, instead of user oriented or co-creation techniques. Furthermore, as the UX team is physically co-located, the project team’s UX member may work at the UX team space and not at the project team. UX related tasks may be mixed in the project’s Scrum table along with development stories.

At the Futurice UX team, a tool for selecting user centered methods in different project situations has been proposed. This framework presents a set of six dimensions of project context, these being: low–high resources; work–entertainment; mature–new; simple–complex; positive–negative customer attitude towards design; and low–high documentation need. A set of methods have been placed along these dimensions. The methods included are interviews, participatory design, observation, questioonaire, prototype, artifact analysis, and usability testing. The tool gives a general recommendability of the tool in a particular situation.

While the tool visualises decision-making, it underestimates the professionals' ability for contextual decision-making by formalising it. It does not provide for learning and iteratively collecting information for the use of these methods in projects to support productisation of approaches, nor does any of the methods provide boundary objects for achieving mutual understanding of the service with the customer. In other words, the tool largely functions in the same low levels of design as agile methods, and do not bring the higher levels into focus.

9.9.2 Service Design at Futurice

When I started inquiring people about their attitudes and experiences with service design, they expressed unfamiliarity with the concept. Even members of Futurice's UX team were not familiar with service design and, prior to workshops, expressed reserved attitudes towards it. Unfamiliarity with the concept seemed to support the thought that service design would have little to offer to Futurice projects.

Methods used for reaching strategic understanding of a customer's business were limited to workshops, presentations, and benchmarking. A typical customer workshop often is a brainstorming session, with the aim of grouping ideas, generating user stories, and discovering user viewpoints, creating common understanding. Still, project definition greatly relies on proposal-centered negotiations, and there seems to be vast room for service design methods to energise customer relationships.

10 INTERMEDIATE RESULTS: OPPORTUNITIES FOR INTRODUCING SERVICE DESIGN AT FUTURICE

Above, I have mapped out Futurice's organisation culture using different methods. In this chapter, I analyse these findings from the perspective of introducing cultural and process change towards a stronger service design approach.

10.1 REQUIREMENTS FOR CHANGE

Futurice's cultural context sets requirements for any introduction of change: While constantly learning and receptive to change, not any change will be easy to bring about. Therefore, based on my cultural exploration, I present requirements for introducing cultural change to Futurice to serve as a foundation for a design solution.

Rigid top-down processes should not be considered first, as introduction of even light processes, such as the weekly meetings, face resistance and require plenty of energy to push through, monitor and maintain. In the best situation, change builds on existing practices and gathered experiences, and aligns with dominant philosophies. If not, these philosophies would need to be remodeled.

Involvement should be a result of understanding and accepting the proposed value and ease of this acceptance. Ease of acceptance can be achieved through incorporating new practices into existing ones.

Ownership over novel concepts and participation should be nurtured, perhaps through gathering support for the concept before bringing it to the whole company, or co-creating the concepts with key people. Furthermore, radical change would be more difficult and risky to embark on, especially if comparable results can be achieved incrementally.

Openness is an important dimension. At Futurice, transparency is a core value, and as such should be incorporated in and promoted by any novel solution.

Understanding and ownership can be achieved through involving individuals in the inception of concepts. Ownership can also be nourished through continued participation. This implies making the solution open in such a way that individuals can contribute to the solution and adjust it even after it has been revealed.

Change should start small, and measures should be iteratively evaluated and constantly adjusted. With our case of design strategy, the UX team is in a central role, as their work is most affected by emerging requirements.

10.2 OPPORTUNITIES FOR SERVICE DESIGN INTRODUCTION

The cultural mapping of Futurice reveals many aspects in which service design may prove highly valuable. First, while Futurice's development and UX functions cater well for low levels of design, there were almost no signs of systematic high level design. Based on interviews, observations, and group discussions, meta level design is generally done *ad hoc* with the customer and haphazardly connected with actual project work, leading to limited customer and user understanding. The use of service design methods could facilitate communication with the customer and provide better uderstanding over the customer's business. To conform with agile development, customer communication should also focus on priorisation of user stories when running in parallel with development.

Another surprising problem is that Futurice, despite its profile as an agile house, executes agile methodologies such as Scrum only half-heartedly. Based on observations, interviews, and workshops, actual practices utilised some Scrum elements, but again, an *ad hoc* approach seemed more prevalent. In this thesis, I shall not directly address this issue. However I interpret it as a symptom of a general difficulty of implementing practices which needs to be taken into account when designing for solutions.

In the sales interface, trust emerged as the principal element of quality, and it is built with good communication. Service design methods and tools can be used to direct and focus communication, reveal unknowns and keep the customer-oriented design of products and services on a right fidelity.

Futurice's UX function leans strongly on "genius design", or utilising professional assumptions and design architectures over actual user testing, resulting in limited user knowledge (Hyysalo, 2011). While this knowledge may be very cost effective and valuable for the product or service, design decisions are often more difficult to communicate with the customer if it isn't backed up with actual user behaviour data (Kurg, 2005). Also, a culture of genius design deflects to a certain extent any intention to introduce systems or methods. This reluctance was shown in workshops and interviews.

Service design methods may provide the lacking user knowledge: Not only are the methods partially same than in UCD, when service design is used as a lense through which projects are looked, these methods may be cognitively easier for reluctant designers to reach for. An introduction of service design methods would also cater to an existing need as Futurice already attends pure service design cases, which may launch partnerships or even emerge from amidst development projects.

The UX team also turned out to be a passive resource repository, a trait that may be linked with the genius design mentality. As such, the UX function may keep itself substantially distanced from the client's business. A stronger service design focus could help expand customer communication to this level.

The agile methods used at Futurice assume perfect knowledge over the user and the customer's business, providing no means for achieving this understanding. Service design methods should be integrated into the customer communication and development process to enhance high level design of products and services.

A stronger design specialisation combined with inhouse development expertise would boost Futurice's competitivity against pure service design houses. Having service design under the same roof with UX and development facilitates project work, communication and management, thus providing a competitive edge against multiple providers. Though service design and UX operate at different levels of design in projects, UX expertise in service design projects is a benefit Futurice already is armed with.

However, Futurice has no actual service design products to facilitate selling and negotiations, and help build trust with the customer. A focused service design effort would help gather experience on service design methods and construct service design oriented product offerings.

To summarise, service design seems to multilaterally provide credible solutions to Futurice's challenges. The question is, how to best execute the introduction of service design into project work and the customer interface. My solutions follow in the next chapter.

11 RESULTS

Above, I have presented some current literature and research on design and agile development. I have critically explored Futurice's organisation environment using interviews, focus group discussions, workshops, observation, and artifact analysis. As a result of this process, I have found several opportunities for improving an agile software contractor with a service design approach, described in the previous chapter. Building on the understanding gained through these discussions and co-creation, I present a set of interlinked steps to answer my design challenge: **How to introduce service design methods into Futurice, an agile software contractor?**

First, I introduce a **UX team strategy**, or a set of issues for the team to keep in mind in terms of development. This serves to establish a common support over the team's general direction and clarify its role within the company.

Second, I introduce a novel role of the **User Experience Product Owner**, or UXPO, a role within an agile project to holistically take responsibility over the design process, method selection, and high level design of a project. This role serves to elevate the UX function's role as auxillary to and overlapping with sales and consultation, as well as to fill in for the lack of high level design in agile methods.

Third, I present a **Service Design Method Toolkit** to support the use of service design methods in projects in an iteratively learning manner. The toolkit introduces the field and methods of service design to the UX team members and helps build a service design portfolio and products.

11.1 UX TEAM STRATEGY

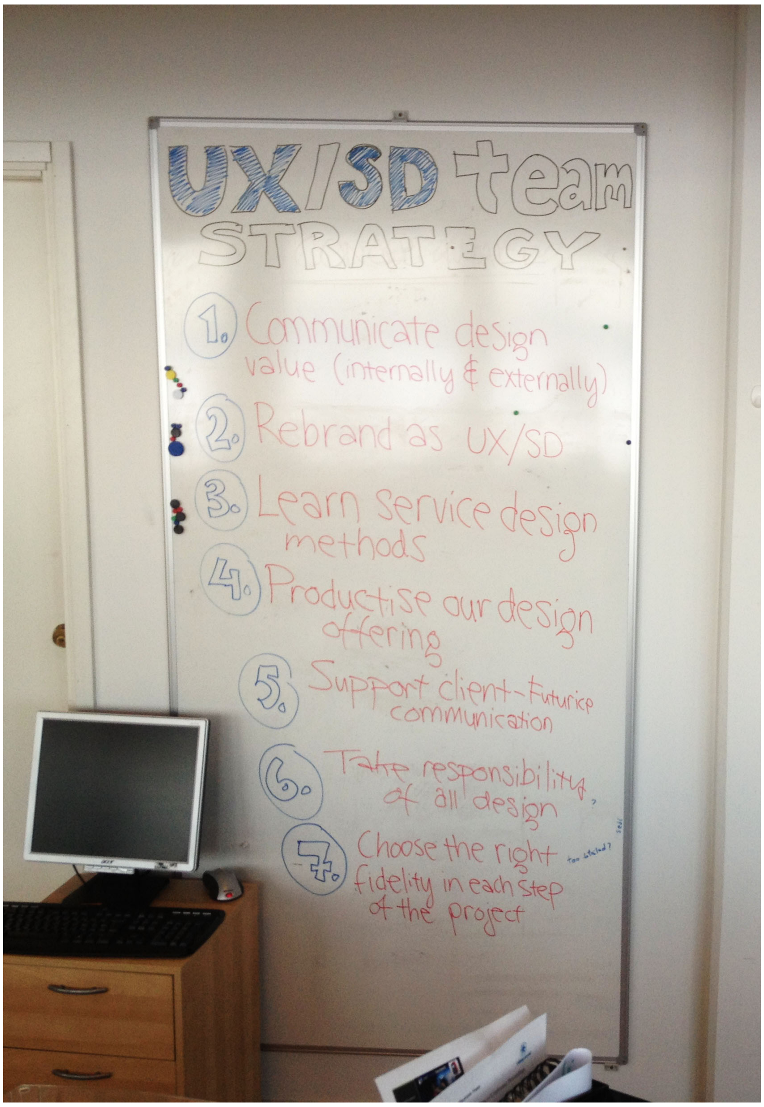
At a workshop, the UX team members agreed upon a set of strategic issues, listed and briefly explained below. Though inspired by Schaffer (2004), these points were created collaboratively to ensure the commitment of team members, and as such, are specific to Futurice. In another environment, a different set of issues may be prioritised, uncovered in co-creation with team members; however, they, too, are likely to concentrate on the same spheres: design function's role in projects; internal and external communication; and competence development.

1. **Communicate design value internally & externally.** The value of methods and processes should be efficiently communicated by designers: internally to sales, proposals, and developers; and externally to customers. This requires active involvement in sales and decision-making over overall processes and project contracts.

2. **Rebrand as UX / SD.** The commitment to service design also needs to show externally to support internal and external communication.

3. **Learn service design methods.** Try out new methods and ways of working. Learn from shortcomings. Productise successes. Share what you learn.

4. **Productise our design offering.** Construct our design experience and ways of working into products to facilitate the communication of design value.


UX/SD team
STRATEGY
1. Communicate design value (internally & externally)
2. Rebrand as UX/SD
3. Learn service design methods
4. Productise our design offering
5. Support client-Futurice communication
6. Take responsibility of all design
7. Choose the right fidelity in each step of the project
too detailed?

5. **Support client communication.** Help achieve understanding over the customer's business. Create boundary objects, visualisations, and documentation to avoid and agree upon conflicting assumptions.

6. **Take responsibility of all design.** The UX team should proactively take responsibility for all usability, user experience, and service design done at Futurice – even the projects nobody asked us to do. This mentality drives the team to actively become sensitive to where its input would provide the best overall value.

7. **Choose the right fidelity in each step of the project.** Evaluate and ensure that work is done at suitable levels of design at given phase of a project.

The focal points are written with a marker on a whiteboard at the UX team's working space. This is to keep the most important viewpoints in focus, as well as provide an open, easily modifiable element for discussion and critical analysis. The points are an internal tool for the team, but can later be used to communicate design strategy for the rest of the company.

11.1.1 Evolution

As of its inception, the strategy wall has invited a little discussion. Confusion and disagreement over some formulations was expressed. Later on the strategy wall took the form of sizeable Post-It notes due to a spatial rearrangement and a need to be able to edit individual list items and rearrange them fluently. The wall remains in the team space, reminding team members of mutually agreed focal points, and has been adopted as a frequent discussion item in monthly meetings.

11.2 UXPO ROLE

I propose a novel project member and customer relations role of the UX product owner (UXPO). The UXPO is a generalist role with the principal holistic mission of driving both UX and concept quality of a service, and removing barriers for and communicating the value of UX/SD work and decisions. To accomplish this, the UXPO utilises their mastery over UX processes and service design methods to evaluate the benefits of using them in different stages of the project, and supports sales in communicating the value with the customer.

The rationale behind the UXPO role leans on three main issues. Firstly, energy and dedication is required to keep important questions in focus. Like with the Product Owner role in Scrum, it is reasonable to believe that an individual with an implicit responsibility helps ensure this focus. Second, domain-specific expertise is required to effectively communicate design value between stakeholders.

Third, design functions benefit from a direct communication channel with the customer to iteratively validate the product concept and requirements. And last, cases as well as Scurm theory suggest that it is beneficial to grant design-level decision making power to a single authority to maintain the effectiveness of project work.

The UXPO should be seen as a sales support role. Relating to the third issue of client communication, an UXPO should be present at customer meetings as early as possible during the negotiation process to be able to combine design process and method expertise with business understanding.

As the UXPO role is novel and there are, to my knowledge, no cases of its application, extra caution should be taken when implementing this measure. The UXPO role should be introduced gradually and piloted with motivated individuals with the adequate set of skills and experience. Its benefits and shortcomings should be closely evaluated in recurrent retrospects with the project teams as well as other UXPOs, and the role immediately adjusted upon signs of trouble.

Alternatively, the UXPO role can initially be distributed to a pair or a small team in order to ensure that the most important aspects are covered, to continuously reflect the theory behind the role against practice, and to collaboratively learn from doing together.

11.3 SERVICE DESIGN METHOD TOOLKIT

I hosted and facilitated a workshop with some of the UX team members with the aim of collecting a service design toolkit, a collection of service design methods deemed useful for service design work in Futurice's context, selected from amongst an abundance of service design methods.

The methods the toolkit provides can be applied front up to create initial requirements, and during development to adjust them as knowledge about the product or service increases.

I formulated an initial strategic service design framework for Futurice in the form of a method toolkit and an evolving experience bank. In the initial inception phase a general understanding over service design methods within the UX team was reached, and the service design method toolkit was co-creatively constructed. From there, the toolkit proceeded to a heuristic, iteratively evolving state of gathering experience and references from real customer cases.

I had assembled a wide library of methods beforehands mainly from Stickdorn and Schneider (2011) and IDEO method cards (Stout, 2003), and continued the set with individual methods from additional sources (see Appendix 2 for a full list of methods). For efficiency, I filtered some of the most heavily user-centered methods out of the assortment prior to the workshop, as most UX team members are familiar with many user-centered methods but still don't use them. The toolkit's goal is not to force any methods the team members have no motivation to use. The methods were selected according to how useful they felt to individuals and how motivated they felt to use them in a future project.

An alternative approach would have been to have the toolkit contain as many methods as possible, with the aim of providing the best possible tool for any given situation. This approach, however, could work against itself by baffling the designers with a crushing amount of options. Such an approach could work better in a society where everyone already knows and is experienced with all or most of the methods. Designing for an introduction of methods, however, a limited set should be better suited. As a bonus, selecting the methods collectively from amongst a multitude of methods increases general knowledge over several methods outside the selected set.

11.3.1 The Toolkit Is a Set of Methods

The toolkit contains a handful of methods. This set of methods was chosen according to people's initial perception of the method and their motivation to use it. We considered using Hyysalo's (2009) criteria for method selection, but discarded the criteria as an unnecessarily top-down and rational approach – it's more important that the tool *feels* useful to and motivates its users. And, indeed, this approach seems to have worked: From a large body of options, each participant found methods they saw interesting and novel, and would like to try out themselves in their current and future projects.

Constructing what information the toolkit provides, a handful of angles were chosen. Experiences can be attached to the methods, such as projects in which they have been used, who executed the method, value created, unknowns unveiled, surprises encountered, learnings regarding the method itself, and so on.

Having selected the set of methods, we grouped the methods in sets according to their use, level of design, and information using them yields. We started off with the taxonomy of activities, representations, recipients, and contents provided by Tassi (2009), but soon felt that this taxonomy wouldn't be useful enough for our purposes, and abandoned it altogether.

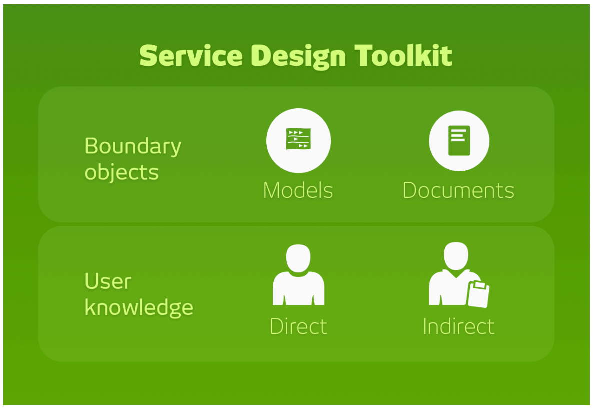
We instead divided the methods into two main groups according to their nature. The first main group was formed of methods that focus on attaining user knowledge. We further distinguished – crudely along the lines of Hyysalo's (2009) taxonomy of sources of user knowledge – two subtypes of user knowledge methods. The first subgroup consists of methods that haul direct user knowledge from the actual behaviour of users, and the second comprises of indirect methods that rely more on heuristics and professionals' assumptions and ability to empathise with the user.

The second main group of methods was identified as boundary objects – means of representing, learning about, and transforming knowledge to resolve the consequences that exist at a given boundary (Carlile, 2002). These representations function as focal points for discussion and decision making, helping achieve a common understanding with the customer. They also give an overview on central issues, supporting discussion and decision making by helping keep it on high enough a level. This is done by visualising and concreting patterns, unknowns, assumptions, and decisions.

We further divided the boundary objects into two subgroups depending on the outcome of the method: models and documents. Models are more visual representations that give an overview of the service from relevant perspectives and help identify risky assumptions. Documents typically take a less visual form of presenting key questions and storing the answers to ascertain common understanding over the goals and parameters of the service with the customer.

11.3.2 Included Methods

The representation *models* selected for the toolkit were the Business Model Canvas, Service Blueprint, Flow Analysis, Customer Journey Map, and Expectation Map.



The selected *documents* are Personas, User Stories, Five Whys, Be Your Customer, and Co-creation.

For gathering *direct user knowledge*, we selected the following methods: Rapid Ethnography, A Day in the Life, Card Sort, Interviews, and Participatory Development.

For *indirect user knowledge*, the methods are Artifact Analysis, Try It Yourself, Empathy Tools, Experience Prototype, and Paper Prototyping.

11.3.3 The Toolkit Is an Artifact

Physically, the toolkit is a two-dimensional piece on the wall at Futurice's UX team's working space, where it can be easily accessed and altered by anyone. The methods it contains are displayed in groups, visualising the differences in utility and thinking.

The toolkit provides no formal process to choose the methods in particular situations, though the descriptions and sample use cases attached

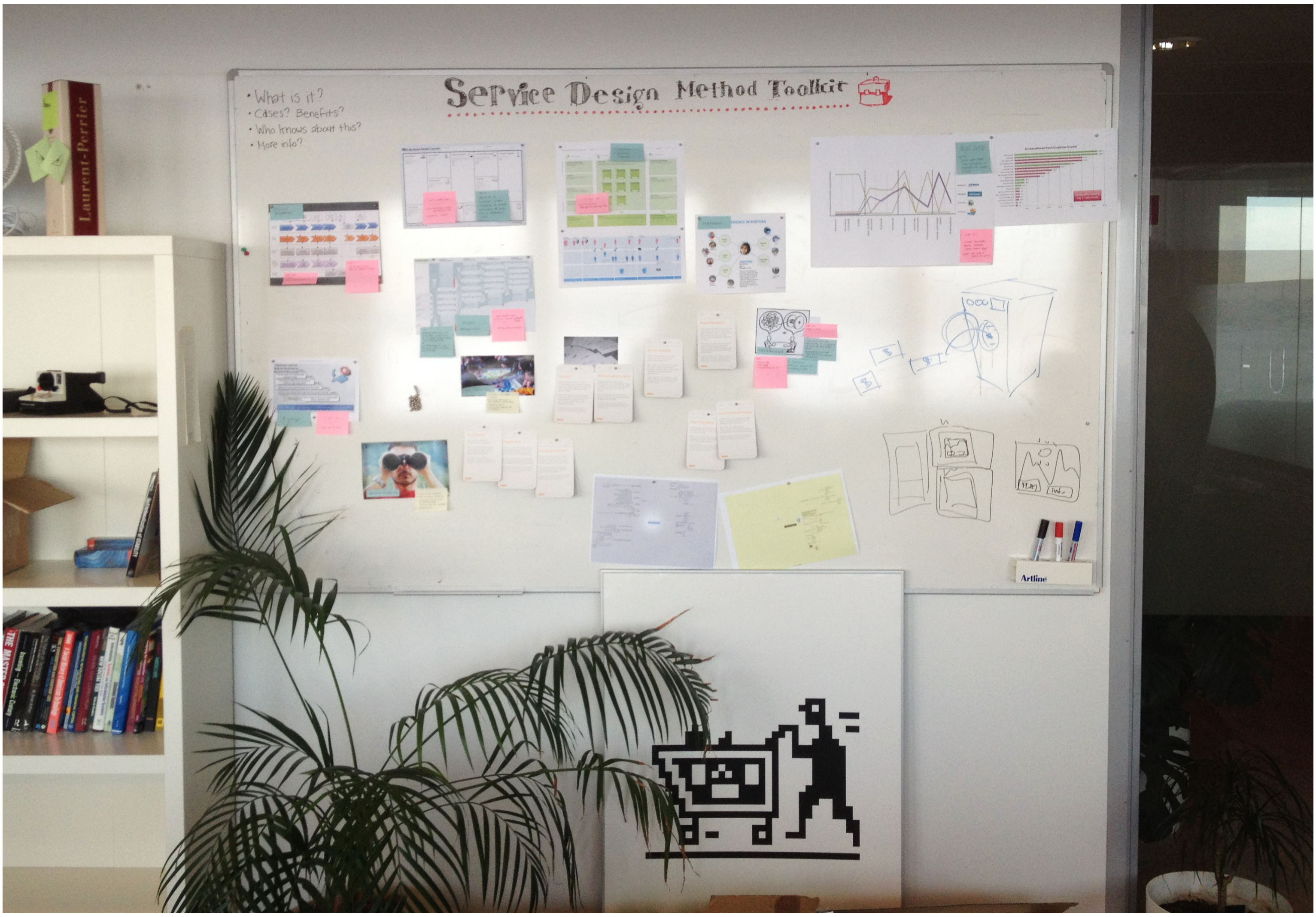
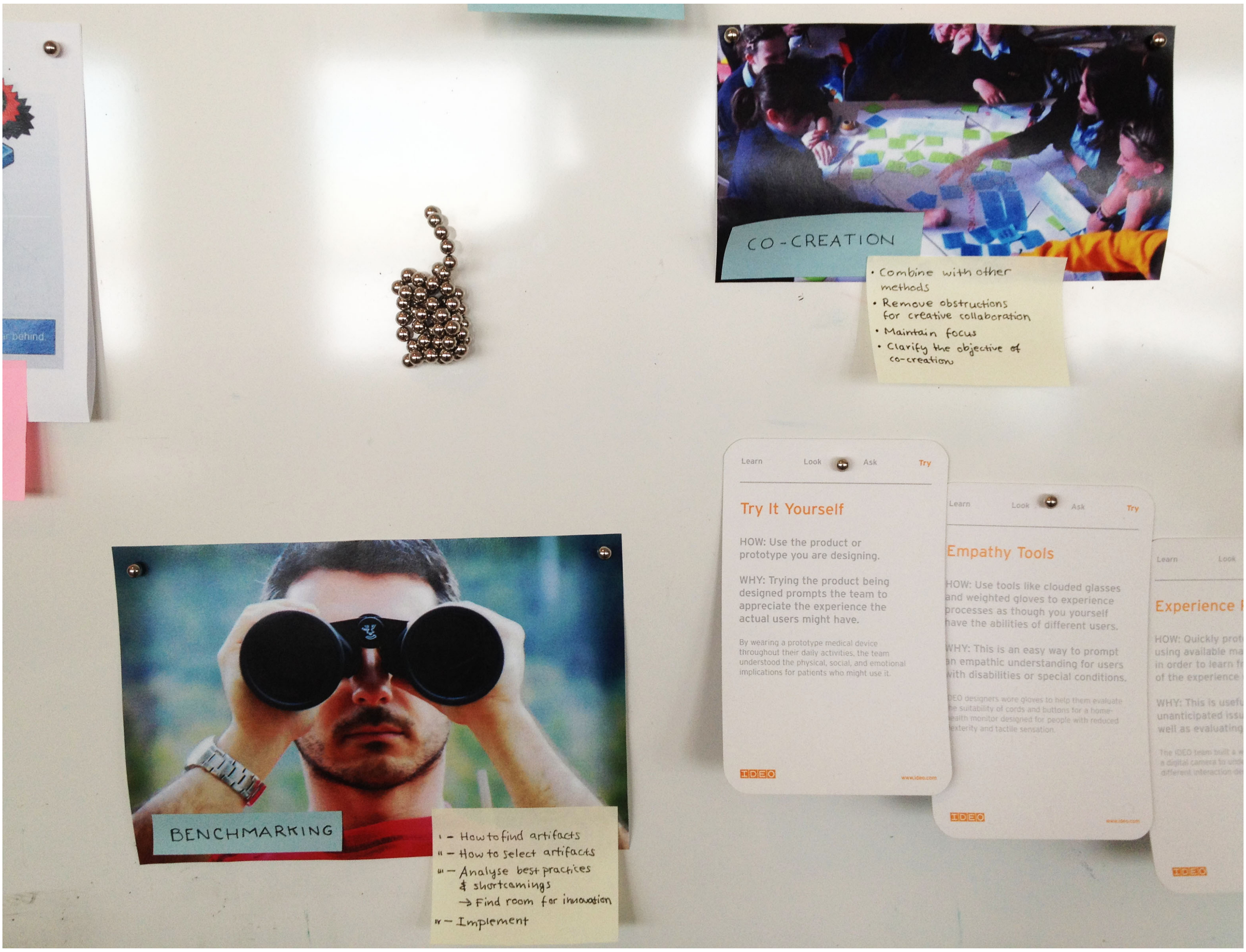


Fig. 5: The Service Design Method Toolkit on a wall in Futurice UX/SD team's premisses.

to the methods facilitate contextual decision-making. It is however in no conflict with Futurice's existing tool for project metric based method selection.

The toolkit presents each method as answers to a set of key questions: What is the method – what is it useful for? In which projects have we used it? Where to get more detailed information on the method – tacit information, literature, links?



- Combine with other methods
- Remove obstructions for creative collaboration
- Maintain focus
- Clarify the objective of co-creation

I – How to find artifacts
II – How to select artifacts
III – Analyse best practices & shortcomings
→ Find room for innovation
IV – Implement

Try It Yourself

HOW: Use the product or prototype you are designing.

WHY: Trying the product being designed prompts the team to appreciate the experience the actual users might have.

By wearing a prototype medical device throughout their daily activities, the team understood the physical, social, and emotional implications for patients who might use it.

Empathy Tools

HOW: Use tools like clouded glasses and weighted gloves to experience processes as though you yourself have the abilities of different users.

WHY: This is an easy way to prompt an empathic understanding for users with disabilities or special conditions.

IDEO designers wore gloves to help them evaluate the suitability of cords and buttons for a home-health monitor designed for people with reduced dexterity and tactile sensation.

Fig. 6: The Service Design Method Toolkit, detail.

11.3.4 The Toolkit Evolves Heuristically

Once established, the toolkit evolves heuristically. Experience is gradually accumulated on the toolkit board's method cards. Should a method not gather experience, it's an indication of either valuable work not being sold, or a method not being as valuable as previously thought. The toolkit currently withholds 20 methods, which is too many to productise each, so dropping out methods without replacing them does not present a problem.

Moreover, the format of the toolkit is subject to constant evolution. For instance, the *models–documents–direct–indirect* typology may prove clumsy or unhelpful. The set of questions posed by the toolkit may be expanded and refined. Extra elements may be introduced to communicate aspects that prove helpful.

The performance of the toolkit should be regularly updated and discussed. I recommend taking it as focal point in weekly or monthly UX team retrospectives.

11.3.5 The Toolkit Has Several Goals

The service design method toolkit has several main goals. First, it acts as a means to focus project thinking into the most important aspects. The main questions also help achieve a common understanding of design work. Furthermore,

Second, the toolkit supports the UX team members' learning on service design methods. This goal is furthered by promoting a cognitive focus on service design; encouraging learning, experience sharing, and experience storing and making it tangible; and providing a framework for communicating design activity value. Furthermore, it also supports communication of design method options and service design value with sales professionals, customers, and UXPOs.

Third, the toolkit is a tool for producing referrable experiences, gradually building the company's design portfolio and helping build service design products. As an added bonus, it recognises design work by visibly attributing designers.

11.3.6 Evolution

The toolkit seems to have been adopted as a fluent part of everyday work – not only by UX/SD team members but project managers and sales people, too. The wall seems to serve as a reminder and give a useful overview of Futurice's expertise with different methods. The assortment of methods has been filled in with details and methods from outside the library of methods I initially assembled. IDEO method cards have been slowly replaced by case examples of our own.

The initial division of methods into categories has dispersed as not useful enough. The amount of methods seems to be small enough to be browsable without further categorisation, regardless of the type of need. This apparent browsability of the content invites the question, could the depth of the content be further increased?

12 DISCUSSION

In this thesis, I have presented a set of actions for Futurice as steps to introduce service design into the business and processes of the company to provide design-driven value in customer projects and customerships.

12.1 POSSIBLE SHORTCOMINGS

While the measures I propose seem to have a lot of potential, they aren't guaranteed to work perfectly and right away. In this chapter I explore the various challenges that remain regarding my approach as a whole as well as individual proposed measures.

Perhaps foremost is the design approach's general conflict with the company strategy. While UX is regarded at the high level as an important strategic spearhead, service design, as of yet, is not – and it may be questioned whether it should be, as such a direction would be something of a deviation from the current strategic core of software production. Hopefully these suggestions will bring the benefits of service design into strategic discussion at Futurice.

A more abstract concern is that of the possibility of conflict of world views. Should Futurice expand its expertise with strong design thinking, will it result in a deepening of the chasm between design and development in a multilateral value environment? To this I have no other answer than to keep this perspective in mind as Futurice's culture grows and diversifies.

A clear limitation of the measures I have presented here is their specificity to the particular context of Futurice. Furthermore, they take place in a dynamic environment and in the midst of an abundance of other measures aiming to improve the company's capacity as a strategic partner, increasing the difficulty of measuring the benefits.

Relating to the specificity of the context, Futurice proved to be not as agile a software house as it could be. This aspect inevitably weakens the applicability of the proposed means in more strictly agile environments.

While I have taken a wide perspective on design possibilities at Futurice in order to holistically grasp interdependecies, the proposals remain to be tested in practice and evaluated against their performance. Whether they bring the benefits they are designed for remains to be seen, and continuous evaluation needs to be conducted to follow their success and adjust them at signs of turbulence.

Despite the width of my approach, I have left out of the scope of analysis several issues that might have proved significant. These include recruitment, organisational structure, and underlying values, and they should be taken into account as the proposed frameworks are tested in practice and fundamental experience over them is accumulated. I have also leaned more strongly on producing artefacts and providing forums in the form of interviews, presentations, and workshops, than seeking out and empowering individual champions to drive the change. While these heroes have seemed to emerge out of the process, and I myself will try to act as one, more attention could be directed towards and ways could be found to highlight this issue.

12.1.1 UX Team Strategy

Introducing a strong service design spearhead is something of a multilateralisation of company strategy, deviating from a traditional "do one thing and do it well" approach. At Futurice, while UX has been recognised as a strategic spearhead, software development still is the core of the company.

Re-branding an agile software contractor for the customers as a hybrid service design + software development house that would excel in both will be substantially difficult, as even the existing UX offering has not been successfully communicated to all of Futurice's customers. Such hybrid companies are scarce, and as the company's value proposition becomes more complex, it also turns more labourious to communicate.

Introducing a strong service design paradigm into Futurice comes with challenges. First, Futurice's culture is engineering and business dominant. For a service design viewpoint to take root, the value service design brings to projects and partnerships requires that its value be understood widely and clarified within the company. A general positive, partly even enthusiastic, reception for UX facilitates this challenge.

Albeit substantial, these challenges seem possible to overcome with sufficient management and energy.

From a wider perspective, a worrysome point is strategic. A core message of the company is reliability in terms of providing hard value, communicated in taglines such as "No bullshit." Challengingly, there may be cases in which a service design measure does not create clearly measurable value. The new service design approach therefore requires a realignment of company message, values, and value proposition.

Furthermore, in these situations, a strong focus on customer satisfaction should be maintained.

Alternate solutions would include gathering and mastering a network of design professionals, forming strategic partnerships with design agencies, and rebranding the UX and Service Design team as its own company. These approaches would lack the crucial competitive benefit of binding design and development work tightly together and formulating joint offerings around that practice.

As to the artifact format of the strategy wall, being part of monthly meetings it seems to have been adopted into use fluently, though it will be difficult to measure how much value it will provide to the team and actual design work. The editability of the wall is both its strength and its weakness: While empowering team members to participate in discussion, strategic focal points hastily written on a whiteboard may feel too volatile to enthusiastically follow.

The strategic points may be different for other agile software companies, though I presume they will be likely to revolve around the same concepts of design function’s role in projects, internal and external communication, and competence development. Not having the most important team concepts on the wall would be because there is either something even more important to put on the wall or a very good reason for not having anything on the wall.

12.1.2 The UXPO role

The new role of the User Experience Product Owner does not come without worry, but poses the question whether the UXPO shoes can reasonably be filled by anyone, as the sphere of the and the pool of skills required are so large. Being novel, no existing experience can be utilised in implementing the approach. Furthermore, a plethora of Scrum authors warn against altering their methodology, claiming that the processes should be implemented meticulously. Introducing the UXPO role is exactly against that advice, though others, too, have altered Scrum to introduce design elements with good results.

Worrisomely, the role of the Scrum Master presents an ominous example, as it seems systematically improperly executed, though being equally or possibly even less challenging than the UXPO role. The gradual introduction of the UXPO role may help in finding an appropriate balance of skills and responsibilities.

Furthermore, although the theory behind the UXPO role would be beneficial, as a novel and unfamiliar concept it may be difficult to communicate and understand, and as such, cultural resistance may complicate its successful introdution. This effect can be mitigated with appropriate communication to ensure the teams and customers in projects with an UXPO understand and welcome the role, the rationale behind it, and the benefits it may bring.

Related to Futurice's evolving cultural environment, it seems clear that the company needs to recruit and train "T-shaped" professionals, or ones with a basic understanding of several fields in addition to professionalism in a single field – be the measures I propose executed or not, though in particular this question arises regarding the UXPO role. In this thesis, I provide no tools to address this challenge.

The deep rooting of the UXPO role may also have strong implications for recruitment, organisational structure, culture, and values, which I have not addressed in this thesis, but require attention in the long run.

If designers are to affect business, they need to learn to communicate with business people. The UXPO role I propose in this thesis underlines that besides business language, communication skills and metaskills such as project work and management skills need to be combined with design understanding for designers to be able to truly work as mediators between world views – and as such influence multidisciplinary projects in multiple levels and phases of design.

I have not provided specific steps as to the introduction of the role in agile processes, enabling the idea to be left hanging in the air with the danger of becoming just an internal discussion point for the UX/SD team.

Also, the timespan available for this thesis was not long enough to enable any reasonable observation of the successfulness of the UXPO concept. For now it remains just that – a concept for intra-team thinking. Further steps could be taken in the direction of making this role more visible within the company and communicating it to other functions.

While, lacking such observation, it is difficult to reliably say whether the UXPO concept should be implemented in other agile houses, it seems apparent that agile models require such a function in one way or another to be able to fluently and professionally work with the higher levels of design.

12.1.3 Service Design Toolkit

While the introduction of the service design toolkit aims to incentivise the usage and exploration of high level design methods by bringing them visibly into the workspace, the result may not be as wished for.

Even in its intended lightness, it may simply not be enough to lower the treshold of experimentation enough. Furthermore, strongly limiting the amount of methods included in the toolkit may exclude useful methods, making it too restricted for the scope of project types Futurice needs to cater for.

These worries, however, are best confirmed or dismissed through experimentation and learning from feedback.

Examples of alternatives for the toolkit approach would have been to produce a checklist of sorts; design and more clearly define a process; and introduce a touring service design champion to consult sales and project teams. The checklist option would not have provided the team with a limited set of methods, and the two other options conflict with the company culture as top-down and rigid.

The toolkit is probably the most complete and exportable solution of the three. Any agile software house may cheaply adopt it and modify it according to the specific needs and experiences in that company, and it will promote the use of beneficial design methods in right parts of projects, provide ideas, help collect experiences and knowledge and, eventually, support the productisation of valuable approaches.

12.2 CONTRIBUTION

If anything, this thesis should serve as underlining at how deep a level the design paradigm can operate within an organisation and its environment. I have shown that design as a methodology can effectively extend agile frameworks' focus, which is limited to the lower levels of design, to encompassing also the higher design levels of society and business.

For Futurice, the solutions I have proposed aim to help form deeper strategic partnerships through increased business understanding and produce ever higher quality digital services. The same goes for other software houses – and creative agencies, as well, who also benefit of increased business understanding, enhanced efficiency in projects. Their customers are more satisfied and benefit from the increased quality.

From an academic perspective, in this thesis I have pointed out a need for a stronger cohesion of the fields and methods of service design and management of software development, and suggests some means to bridge this gap. I have also pointed out a discrepancy between service design and UCD as working on different levels of design, and as such, these disciplines, too, need to communicate better.

For the design profession, this thesis underlines the value of multilateral expertise and metaskills in high level design.

12.3 FURTHER RESEARCH

To quantify the value brought by taking the measures I have proposed, several different factors could be examined. For instance, a mass of project data could be gathered to see whether the proposed approach brings measurable quality boost. Measured elements could include subjective views of teams on project work and outcome quality, customer satisfaction, and end user satisfaction. The effect of UX and service design on the depth of partnerships, customer satisfaction, and end user satisfaction could be measured.

In this research I have aimed for strong context sensitivity. As such, the measures may face difficulties in other contexts. Further research can be conducted by implementing similar measures in other contexts, both similar than and different from Futurice. For example, these methods could be used for bringing a service design house more strongly into software development. Also, fields other than design strategy, service design, UX, and agile methods may provide useful lenses to this topic, examples being operations management, sales theory, business strategy, change management, and new product design.

13 CONCLUSION

In this thesis, I have presented a set of interlinked steps to answer the design challenge of **introducing service design methods into Futurice, an agile software contractor.** I have used design methods – interviews, workshops, visualisations, and presentations – to discover potential for Futurice to use design to increase its competence as a strategic partner.

Based on the gathered cultural knowledge, literature review, and co-creative work, I have provided measures to fill this potential considering Futurice's agile culture. These measures are: A UX/SD team strategy wall; a novel role of the UX Product Owner (UXPO); and a Service Design Method Toolkit.

While my intention has been to formulate the measures to be taken to be modular, they all reflect an existing demand for Futurice to transform itself and its working philosophy from a software contractor into a deep strategic design partner. I expect these measures to improve Futurice's capabilities for design work in a way that is natural for the company in terms of culture and domain, while encouraging constant evolution and learning.

14 REFERENCES

K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, et al.: *Principles Behind the Agile Manifesto*. http://agilemanifesto.org/principles.html, Agile Alliance, 2001. Retrieved on Dec 27, 2011

M. Budwig, S. Jeong, K. Kelkar: *When user experience met agile: a case study. Proceedings of the 27th international conference extended abstracts on human factors in computing systems* (pp. 3075–3084). ACM, 2009

K. Best: *Design Management: Managing Design Strategy, Process and Implementation*. Ava Publishing, 2006

B. Borja de Mozota: *Design Management. Using Design to Build Brand Value and Corporate Innovation.* Allworth Press, 2003

P. Carlile: *A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development*. Organization Science, 2002

S. Chamberlain, H. Sharp, N. Maiden: *Towards a framework for integrating agile development and user-centered design. Extreme Programming and Agile Processes in Software Engineering*, 143–153. Springer, 2006

J. Chapman: *Emotionally Durable Design: Objects, Experiences & Empathy*. Routledge, 2005

P. Deemer, G. Benefield, C. Larman, B. Vodde: The Scrum Primer. http://www.scrumprimer.com/, 2010. Retrieved on January 4, 2012

M. Engwall, P. Magnusson, C. Marshall, T. Olin, R. Sandberg: *Creative Approaches to Development: Exploring Alternatives to Sequential Stage-gate Models*. Fenix Research Program & Stockholm School of Economics, 2001

L. Giusti, M. Zancanaro: *What we can learn from services in order to design services*. FIS'2009 Proceedings of the Second Future internet conference on Future internet, Springer-Verlag Berlin, 2010

J. Gosper, J. Agathos, R. Rutter, T. Coatta: *UX design and agile. Communications of the ACM*, 54(1), 54. ACM, 2011

ISO: *ISO 9241 part 210: Human-centered design for interactive systems*. International Organization for Standardization (ISO), 2010

S. Hyysalo: *Käyttäjä tuotekehityksessä. Tieto, tutkimus, menetelmät.* Otavan Kirjapaino Oy, 2009

S. Hyysalo: *Käyttäjätieto ja teknologian sosiaalinen muotoutuminen.* In *Ihmisen ja tietokoneen vuorovaikutus*, Gaudeamus, 2011

P. Kroll, P. Kruchten: *The Universal Rational Process Model Made Easy: A Practitioner's Guide to RUP*. Addison Wesley, 2003

S. Kurg: *Don't Make Me Think: A Common Sense Approach to Web Usability*. New Riders Press, 2005

E. Kurvinen: *Prototyping Social Action*. Gummerus Printing, 2007

E. Law, V. Roto, M. Hassenzahl, A. Vermeeren, J. Kort: *Understanding, scoping and defining user experience: a survey approach.* CHI '09 Proceedings of the 27th international conference on Human factors in computing systems, ACM, 2009

T. Leinonen: *Designing Learning Tools. Methodological Insights*. Aalto University School of Art and Design publication series A 111, 2010

S. Moeller: *Characteristics of services – a new approach uncovers their value*. Journal of Services Marketing, Emerald Group Publishing Limited, 2009

J. Nicholas, I. McGraw-Hill: *Competitive Manufacturing Management. Continuous Improvement, Lean Production and Customer-Focused Quality*. 1998

J. Nielsen: *Usability Engineering*. Morgan Kaufmann, 1994

J. Nielsen: *Usabitlity ROI Declining, But Still Strong*. Useit.com, 2008. Retrieved on Dec 28, 2011.

D. Norman: *The Design of Everyday Things*. Doubleday Business, 1990

J. Patton: *Twelve emerging best practices for adding UX work to Agile development.*

http://www.agileproductdesign.com/blog/emerging_best_agile_ux_practice.html, Agile Product Design, 2008. Retrieved on Dec 27, 2011.

M. Poppendieck, T. Poppendieck: *Lean Software Development – An Agile Toolkit*. Boston: Addison-Wesley, 2003

D. M. Rousseau: *Psychological Contracts in Organizations: Understanding Written and Unwritten Agreements*, Thousand Oaks, CA: Sage Publications 1995

E. Schaffer: *Institutionalization of Usability. A Step-by-Step Guide.* Addison Wesley, 2004

K. Schwaber, M. Beedle: *Agile Software Development with Scrum*. Prentice Hall, 2001

J. Stickdorn, M. Schneider: *This is Service Design Thinking: Basics, Tools, Cases*. Book Industry Services, 2011

W. Stout: *IDEO Method Cards*. William Stout, 2003

R. Tassi: *Service Design Tools*. http://www.servicedesigntools.org/, Service Design Tools, 2009. Retrieved on November 11, 2011

K. Ulrich, S. Eppinger: *Product Design and Development*. McGraw-Hill Education Singapore, 2007

R. Verganti: *Design-Driven Innovation.* Harvard Business Press, 2009

P. Wild: *HCI and the analysis, design, and evaluation of services. Proceedings of the 22nd British HCI Group Annual Conference on People* *and Computers: Culture, Creativity, Interaction – Volume 2*. British Computer Society, 2008

J. Womack, D. Jones: *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Free Press, 2003

F. Wärn: *Käyttäjäkeskeisen suunnittelun ja ketterien menetelmien yhdistäminen prosessimallien näkökulmasta*. Aalto University, 2010

15 APPENDIX

15.1 INTERVIEWS AND WORKSHOPS

All interviews and workshops took place in 2011. Altogether there were one observation, six interviews, and nine workshops / group discussions.

UX day	Jun 21
Interview, HR specialist	Sep 1
Client panel discussion	Sep 2
Project kickoff meeting observation	Sep 7
Interview, UX designer	Sep 7
Interview, HR specialist	Sep 8
Sales discussion	Sep 15
Interview, product owner	Sep 20
UX group development discussion	Sep 20
UX group discussion	Sep 27
Interview, project manager	Sep 29
Process workshop	Oct 3
UX Strategy workshop	Oct 3
Interview, developer	Oct 3
UX Strategy workshop	Oct 6
Scrum workshop	Oct 9–10
Service Design workshop	Oct 28

15.2 LIBRARY OF DESIGN METHODS

To choose the set of methods to include in the Service Design Method Toolkit, I assembled a library of methods. From this library, I pre-selected a variety of methods for a Futurice workshop to choose from. I've assembled the library from Stickdorn & Schneider 2011, IDEO method cards, servicedesigntools.com, and other sources such as *Harvard Business Review*.

- Design drivers
- Visual mental imagery
- Parallel design
- GOMS (goals, operator, methods, and selection rules)
- Human Processor Model
- Keystroke level modeling
- Card sorts
- Tree tests
- Ethnography
- Heuristic evaluation
- Usability inspection
- Meeting: users + developers + human factors people evaluate step by step
- Consistency inspection
- Activity analysis
- Inquiry methods
- Task analysis
- Focus groups
- Questionnaires & surveys
- Rapid prototyping
- Remote usability testing
- Thinking aloud

- Discount usability testing
- RITE (Rapid Iterative Testing and Evaluation)
- Subjects in tandem
- Component-based usability testing
- Cognitive walkthrough
- Benchmarking
- Fantasy X
- Meta-analysis
- Personas
- AT-ONE workshops
- Stakeholder maps
- Service safaris
- Shadowing
- Customer journey maps
- Customer journey canvas
- Contextual interviews
- Five whys
- Cultural probes
- Mobile ethnography
- A day in the life
- Expectation maps
- Idea generation
- What if...
- Design scenarios
- Storyboards
- Desktop walkthrough
- Service prototypes
- Service staging
- Agile development
- Co-creation
- Storytelling

- Service blueprints
- Service roleplay
- Customer lifecycle maps
- Business model canvas
- Anthropometric analysis
- Affinity diagrams
- Activity analysis
- Cognitive maps
- Behavioral archaeology
- Behavior sampling
- Behavioral mapping
- Be your customer
- Camera journal
- Card sort
- Cognitive task analysis
- Collage
- Competitive product survey
- Conceptual landscape
- Cross-cultural comparisons
- Draw the experience
- Empathy tools
- Error analysis
- Experience prototype
- Extreme user interviews
- Flow analysis
- Fly on the wall
- Foreign correspondents
- Guided tours
- Informance
- Long-range forecasts
- Narration

- Path to participation
- Paper prototyping
- Personal inventory
- Preduct next year’s headlines
- Quick-and-dirty prototyping
- Rapid ethnography
- Relationship mapping
- Scale modeling
- Scenarios
- Scenario testing
- Secondary research
- Social network mapping
- Still-photo survey
- Surveys & questionnaires
- Time-lapse video
- Try it yourself
- Unfocus group
- Word-concept association
- Historical analysis
- Ten types of innovation