Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

Professional MBA
Entrepreneurship & Innovation

Design Methods for Innovative Software Products and Services

A Master's Thesis submitted for the degree of
"Master of Business Administration"

supervised by
Prof. Dr. Marc Gruber

Kristina Dimitrova Stoitsova

1353474

Vienna, June 22, 2015

Affidavit

I, **KRISTINA DIMITROVA STOITSOVA** , hereby declare

1. that I am the sole author of the present Master's Thesis, "DESIGN METHODS FOR INNOVATIVE SOFTWARE PRODUCTS AND SERVICES", 57 pages, bound, and that I have not used any source or tool other than those referenced or any other illicit aid or tool, and
2. that I have not prior to this date submitted this Master's Thesis as an examination paper in any form in Austria or abroad.

Vienna, 22.06.2015

Signature

Abstract

The main goal of this master thesis is to make an overview of contemporary design approaches that exist for the development of innovative software products and services. The aspect of product design that is discussed in this work is the user-facing design (aka usability, user experience). This text does not aim to cover technological, marketing, production, operations or other aspects of the product development lifecycle.

This analysis provides a general discussion of the creative process of design and introduces the concept of a "frame" for addressing the main design paradox. It reviews various design approaches – User Centered Design, Activity Centered Design, Data Driven Design, Systems Design and Genius Design. A framework for assessing and comparing them is introduced to help the reader evaluate which one would best fit a certain context. The User Centered Design approach is analysed in greater detail. The document provides a description and comparison of five main User Centered Design methods – Participatory Design, Contextual Design, Design Thinking, Empathic Design and Lead User Method. The structure of the processes for applying them in new software product or service development is presented. A real-life case study about the application of Design Thinking at SAP is included to capture the practical aspects of applying design methods to software projects.

In practice, this analysis can be used:

- As a reference list of options for applying design approaches and methods, specifically for the purpose of developing innovative software products and services

- To inform a decision which design approach and method would be most effective in a certain context

- To provide references for further analysis of related aspects – human-computer interaction, user experience, software development lifecycle, software teams design

Contents

Abstract iii
Contents iv
List of Figures v
List of Abbreviations v
1 Introduction 1
1.1 Purpose and Research Questions 1
1.2 Motivation 2
1.3 Disposition 2
1.4 General Discussion of Design Challenges 3
2 Design Approaches for Innovative Software Products and Services 5
2.1 User Centered Design (aka Human Centered Design) process 6
2.2 Activity Centered Design (a.k.a. Usage Centered Design) 7
2.3 Data Driven Design 10
2.4 Systems Design 13
2.5 Genius Design 15
2.6 Comparison of Design Approaches 16
3 Drill-down into User Centered Design Methods 23
3.1 Participatory Design 23
3.2 Contextual Design 25
3.3 Design Thinking 33
3.4 Empathic Design 35
3.5 Lead User Method 39
3.6 User Centered Design Methods "At a Glance" Comparison 44
4 Case Study of Applying Design Thinking at SAP 48
4.1 Case Study Selection Process 48
4.2 Historical Background 48
4.3 Organizing for Applying Design Thinking 49
4.4 The Design Thinking Team 50
4.5 The Problem 50
4.6 The Process 50
4.7 In Hindsight 53
5 Summary and Learnings 54
6 Bibliography 58

List of Figures

Figure 1: Conceptual procedure for applying Usage Centred Design process 9
Figure 2: Conceptual Components of a System 14
Figure 3: UCD "Profile" 19
Figure 4: ACD "Profile" 19
Figure 5: Comparison between UCD and ACD 20
Figure 6: DDD "Profile" 20
Figure 7: Comparison between UCD and DDD 21
Figure 8: Contextual Design Process 27
Figure 9: Example of a Flow Model in Contextual Design 28
Figure 10: Example of a Cultural Model in Contextual Design 28
Figure 11: Example of a Sequence Model in Contextual Design 29
Figure 12: Example of a Physical Design in Contextual Design 29
Figure 13: Example of an Artifact Model in Contextual Design 30
Figure 14: Example of Storyboards in Contextual Design 31
Figure 15: Example of User Environment Design in Contextual Design 31
Figure 16: Example of a Paper Prototype in Contextual Design 32
Figure 17: Empathic Design process 37

List of Abbreviations

ICT - Information and Communications Technology

UCD - User Centered Design

HCD - Human Centered Design

UX - User Experience

UI - User Interface

GUI - Graphical User Interface

HCI - Human Computer Interaction

R&D - Research and Development

POS - Point of Sale

HPI - Hasso Plattner Institute

1 Introduction

This master thesis is about design methods, their characteristics and their relevance in specific contexts with regards to developing innovative software products and services. Developing innovative software products and services is an important piece of the whole picture of entrepreneurship and innovation in today's world due to the critical role that the ICT (Information and Communications Technology) industry plays in business as well as in society. There are multiple aspects of design – user experience design, technological design (i.e. software architecture and technology stack), design of the product development lifecycle, organizational design optimized for innovation, etc. This master thesis focuses on product design with an emphasis of user experience.

There is extensive literature that describes a specific design approach or a specific design method or the particular techniques that can be utilized at every stage of a design method application. However, there is a general lack of literature that puts all of those in one "big picture" representation. Also some of the methods related to a certain design approach are seemingly very similar and it is not trivial for a person, who does not have broad experience with any of them, to understand the differences and to figure out in what contexts one would be more effective than the other.

This master thesis aims to provide an overview of the design approaches, bringing them together and making a comparison. It then zooms into one of those approaches – UCD (User Centered Design) and gives a more detailed description of the main design methods available, with some of the techniques that can be put into action in their application. The goal is to give a description, classification and references for further research.

1.1 Purpose and Research Questions

In a world where companies' success and even survival depends on their capability to come up with and develop innovative products, the topics related to designing those novel products and services becomes more and more important. This master thesis tackles the following research questions:

- Which are the ways to approach innovative software products/services development and what are the specific methods that can be applied?
- How should one decide which approach and methods to use? What are the criteria that one should take into consideration when choosing an approach and a specific method?

- Should those methods be applied in their purist form or are combinations and/or modifications of methods appropriate? What is the general practice in the industry?

1.2 Motivation

I got inspired to work on the topic of applying design methods for innovative software products and services by the experience and pressing challenges that I have in my current role as a Director of R&D (Research and Development) in UI (User Interface) at a large software company that produces software for virtualization, datacenter management and cloud services. My direct area of involvement is the GUI (Graphical User Interface) of products, which is the part of the product that drives to a large extent the UX (User Experience) with the product and hence, the level of customer satisfaction with it. I feel the urge for my team to be able to innovate and drive the GUI of the product in a direction that will increase adoption and customer satisfaction and I believe that a good systematic view over the design approaches and design methods available to software development organizations is crucial for the success of any manager and team in a similar situation.

1.3 Disposition

The master thesis consists of four charters. In the current charter *Introduction* I discuss the goal and focus of the master thesis, define the research questions, touch upon my personal motivation to investigate this topic and do an introduction into the general problem that design tackles.

The next chapter, *Design Approaches for Innovative Software Products and Services*, dives into the topic of the general approaches to the design of software products and services. I discuss five main directions – UCD (User Centered Design) a.k.a. HCD (Human Centered Design), Activity Centered Design a.k.a. Usage Centered Design, Data Driven Design, Systems Design and Genius Design. Lastly, I propose a framework for presenting those design approaches in a uniform and visual way. I also make an attempt to compare them based on ten criteria, to visually illustrate the differences among them and to suggest some combinations based on the principle of complementing their strengths for solving a problem.

The chapter *Drill-down into User Centered Design Methods* zooms into the UCD approach and focuses on the main and most commonly used design methods. It provides a description of five major design methods – Participatory Design, Contextual Design, Design Thinking, Empathic Design and the Lead User Method. The process of applying each of the methods as well as the main assumptions and important considerations are described.

In the chapter ***Error! Reference source not found.*** I refer to several examples of application of esign methods at well known, industry-leading companies and discuss the issue of purist versus adapted design methods that those companies apply to facilitate their own innovation process as well as the innovation process of their ecosystems.

1.4 General Discussion of Design Challenges

In the last years terms like "Design Thinking" have gained broad popularity and have become overused and overloaded to such an extent, that some advocate against using the term "Design Thinking" at all. In order to overcome the risk of putting too vague definition of the design domain, it makes sense to start with a general discussion of the design challenges. Discussing those challenges and the design reasoning process will help us define the basic design problem as well as the high level process of coming up with solutions.

One way to look at the design reasoning process is to put in the context of formal logic (Dorst, 2011). In formal logic we use the basic equation *Equation 1: Basic Equation of Formal Logic* to analyze different settings of knowns and unknowns:

Equation 1: Basic Equation of Formal Logic

WHAT (thing) + HOW (principle) = RESULT (observed)

In **deduction** we have the 'what' (object, service, system) and the 'how' (operating principle) as a known and we are able to deduct or predict what the 'result' would be, so the basic equation transforms to the one presented in *Equation 2: Deduction Equation.*

Equation 2: Deduction Equation

WHAT (thing) + HOW (principle) = ?

For example having a system of a sailing boat, sea and certain wind conditions, by knowing how the boat is operated and how it interacts with the wind conditions, we can predict where the boat would be in a given point in time.

In **induction** we have as a given the 'what' (object, service, system) and the 'result' and we can induce or suggest what the 'how' (the working principle that when applied to the WHAT produces the RESULT) might be. The basic equation transforms to the one presented in *Equation 3: Induction Equation.*

Equation 3: Induction Equation

WHAT (thing) + ? = RESULT (observed)

Through induction we come up with a hypothesis of a working principle that could explain the observed behavior. For example having a sailing boat and observing its movement at sea, we can induce some working principles of how the boat is operated and interacts with the wind. This is a creative act and is core to scientific "discovery". It explains how hypotheses are formed.

The basic equation in design is similar, but focuses on the aspired value rather than the result, as shown in *Equation 4: Basic Design Equation.*

Equation 4: Basic Design Equation

WHAT (thing) + HOW (principle) = VALUE (aspired)

The basic reasoning pattern in productive thinking is **abduction** (Dorst, 2011). Abduction 1 is to have the 'how' and the 'value', but to be missing the 'what' (service, system). It is related to conventional, "closed" design problem solving. It is represented by Equation 5: Abduction 1.

Equation 5: Abduction 1

? + HOW (principle) = VALUE (aspired)

In contrast, in Abduction 2 at the start of the problem solving process we only know the value we want to achieve, but we don't know 'what' to create (service, system) and we don't know what 'working principle' to use. This is more related to conceptual design and is represented by Equation 6: Abduction 2

Equation 6: Abduction 2

? + ? = VALUE (aspired)

The response to Abduction 2 is the development or adoption of a "frame". A frame is the general implication that by applying a certain working principle, we will create a specific value. The term "framing" is used within design literature for the creation of a (novel) standpoint from which a problematic situation can be tackled. The concept of framing implies that IF we look at the problem situation from this viewpoint and adopt the working principle associated with this position, THEN we will create the value we are striving for. In solving the parallel creation of a thing and its way of working, the most logical way is to start backwards, from the only "known" in the equation – the value and then to adopt or develop a frame.

Deduction and induction are used in analysis, Abduction 1 is used in conventional problem solving, while Abduction 2 is used in design. The creation and use of frames is inherently related to Abduction 2. Frame creation is done not necessarily by addressing the core design paradox head-on, but by looking at the broader problem situation and searching for surrounding themes that

can foster the creation of new frames. In later analysis of design methods we will see how the use of frames is applied in solving particularly novel and complex design problems.

2 Design Approaches for Innovative Software Products and Services

In order to give focus to our analysis of design methods, we are going to look more specifically at the UX or 'usability' aspect of design and put an emphasis on usability. According to (ISO9241-11, 1998) 'usability' is defined as the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." The system itself is the object (WHAT), while the UX of a product or services is a representation of the "working principle" (HOW) of a system. The goal of the designer is to make sure the designed system with its UX achieve the aspired VALUE.

A simple way to test usability of a product or service by (Rubin & Chisnell, 2008) is to check whether "the user can do what he or she wants to do, the way he or she expects to be able to do it, without hindrance, hesitation, or questions." According to (Rubin & Chisnell, 2008) usability objectives are:

- **Usefulness** – the extent to which the product/service enables the user to achieve their goals, and an assessment of the user's willingness to use the product at all.
- **Efficiency** – a measurement of how quickly the user's goals can be accomplished accurately and completely (usually measured by measuring time)
- **Effectiveness** – a measure of the extent to which the product behavior matches the user's expectations and how easy it is for the user to use the product to achieve their intents
- **Learnability** - user's ability to operate the system to some defined level of competence after some predetermined period of training (which can also be zero training). Also, refers to ability for infrequent users to re-learn the system.
- **Satisfaction** (likeability) – captures user's perceptions, feelings and opinions of the product, usually captured through both written and oral communication.

Various UX literature sources suggest different approaches to designing a software product or service. There are five main approaches that have significant importance: UCD (User Centered Design) aka HCD (Human Centered Design). Activity Centered Design aka Usage Centered Design, Data Driven design, Systems Design and Genius Design.

Each of the design approaches is discussed in detail in the following sections of this charter.

2.1 User Centered Design (aka Human Centered Design) process

One of the widely discussed problems of innovation, especially in the area of ICT where innovation is often driven by technology push rather than market pull, is the risk that the product will only match the needs of a handful of people. This gap between the world of designers and users (Muller, 2002) is what UCD attempts to fill in by involving end users heavily during the whole product development process and especially in the fuzzy front-end of designing novel products and services. UCD puts the user in the center and focuses on user's needs and goals. The designer's role is to translate them in an appropriate design. Because of this UCD is good when designers are working in domains that are unfamiliar to them. As the core concept is empathy with users and the center of focus is on people using the product, the risk of not meeting their needs or creating an un-usable product is to a great extent mitigated.

Per (ISO9242-210, 2010) "*Ergonomics of human-system interaction: Human-centered design for interactive systems*", there are 6 principles of UCD:

- The design is based upon an explicit understanding of users, tasks and environments
- Users are involved throughout design and development
- The design is driven and refined by user-centered evaluation
- The process is iterative
- The design addresses the whole user experience
- The design team includes multidisciplinary skills and perspectives

Of course UCD is not a "one size fits all" solution to bringing in usability into a product or service. Even though it heavily involves users in the design and development of products and thus ensures that their voice is incorporated into the design, one must take into consideration that it is often hard for people to generate and even evaluate new product ideas. An illustration of this is Henry Ford's famous quote "*If I had asked people what they wanted, they would have said faster horses.*" Also identifying the "right" users might be a challenge.

Core **analysis** tools used in User Centered Design are: personas, scenarios and user stories/use cases.

Personas are fictional characters that represent a group of users. "*An imaginative description of a user archetype which is based on the real research and observations*"

Scenarios "*describe how a product or a system would be used by personas in the real life*". Scenarios include information about goals, motivations, actions and reactions.

User story/use case is a simple template for one single goal that the user wants to achieve with the product.

Interestingly criticism to the UCD method comes from sources like Donald Norman, one of the gurus in the field of usability, design and cognitive science, and one of the pillars of UCD. (Norman D. A., 2005) had a very controversial article on the shortcomings of UCD and why Activity Centered Design (which we discuss in the next section) is superior. Norman questions the corollary of UCD that technology should adapt to people, rather than people adapting to technology. He gives numerous examples of products that were not designed with the principles of UCD in mind, but are nevertheless used by billions of people of different background, education and skills in a common way (e.g. the automobile and multiple everyday objects like cameras, typewriters, sporting equipment, kitchen utensils, watches and clocks). He advocates for the idea that the reason why all those devices work so well is the fact that they were designed with a deep understanding of the activities they were meant to help be performed. His belief is that successful devices are the ones that fit gracefully into the activity that they are designed to support. In a later clarification to that article ((Norman D. A., 2005) elaborates on the "*limited view of design*" that UCD has developed due to focusing on "*page-by-page, screen-by-screen analysis*" and largely ignoring user activities in their entirety. He advocates that the focus should be less on personas and scenarios and more on designing for activities, focusing on tasks, building task flows and especially designing for negative scenarios, thinking about what happens when the user goes out of the "happy path". The core controversy seems to be that *"To the Human-Centered Design community, the tool should be invisible, it should not get in the way. With Activity-Centered Design, the tool **is** the way."*

2.2 Activity Centered Design (a.k.a. Usage Centered Design)

Usage Centered Design is a term coined by software developer and Professor Larry Constantine and Lucy Lockwood that focuses on user intentions and usage patterns. It is very similar to activity-centered design (ACD), which is also concerned more with the activities of users but not the characteristics of users.

ACD analyzes users in terms of the roles they play in relation to systems and employs abstract use cases or task analysis. It derives visual and interaction design from abstract prototypes based on the understanding of user roles and task cases (Constantine & Lockwood, 1999). This design philosophy explores what people do, without focusing on people themselves per se, which differentiates this approach to User Centered Design.

ACD is largely based on formal, abstract models such as models of interaction between user roles, UML workflow models and task case and role profiles. Using abstract modelling is a key differentiator with UCD where designers use realistic personas, scenarios and high-fidelity prototypes. (Constantine, Human Activity Modeling: Toward A Pragmatic Integration of Activity Theory and Usage-Centered Design, 2009). The major differences with UCD are described in *Table 1: Comparison between UCD and ACD*

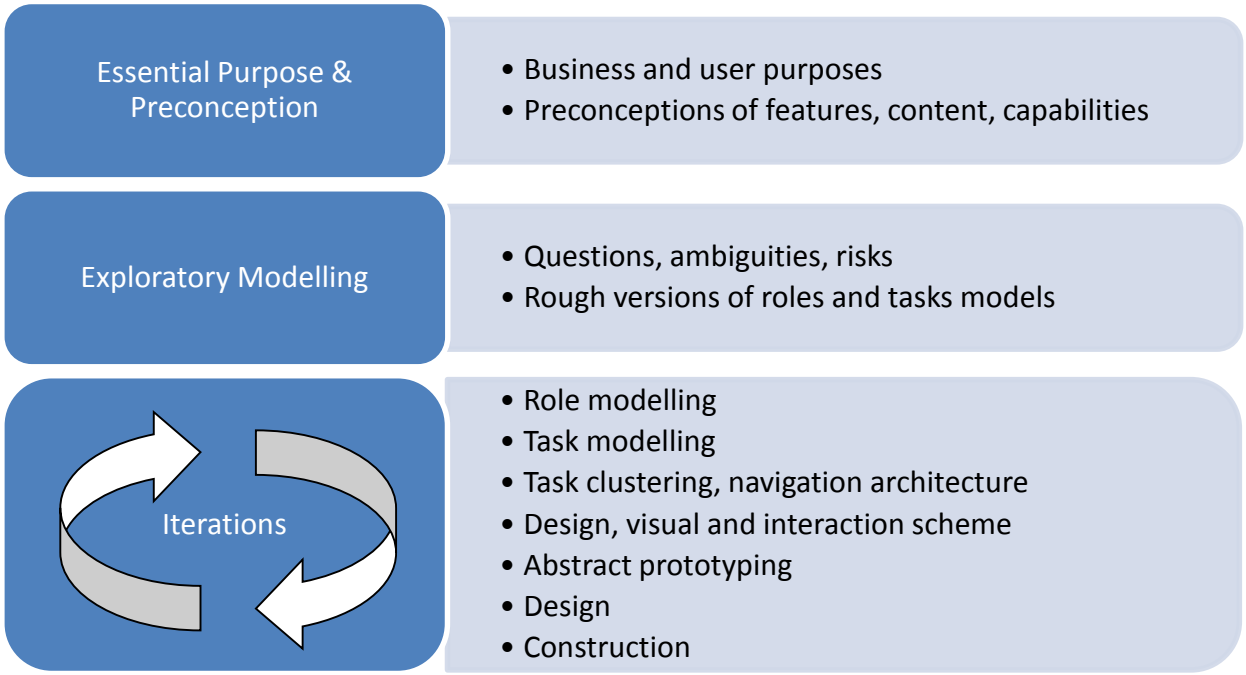
Table 1: Comparison between UCD and ACD

UCD (User Centered Design)	ACD (Activity Centered Design)
Focus is on users: user experience and user satisfaction	Focus is on usage: improved tools supporting tasks accomplishment
Driven by user input	Driven by models and modeling
Substantial user involvement (user studies, participatory design, user feedback, user testing)	Selective user involvement (explorative modeling, model validation, usability inspections)
Design by iterative prototyping	Design by modeling
Highly varied, informal, or unspecified processes	Systematic, fully specified process
Design by trial-and-error, evolution	Design by engineering

The characteristics discussed above make ACD good for intense, focused, complex activities; refining task flows; making actions more efficient.

As a highly systematic process, there is a conceptual procedure for applying ACD. Procedures may vary between different methods that fall into the broad ACD category, but all of them follow a common conceptual workflow. The workflow has a few preliminary steps in which business and user goals are defined, anticipated features and capabilities are identified and prioritized, risks and open questions are defined and rough role models and task models are built. Those preliminary steps are followed by (potentially) multiple iterations during which a subset of the role and task models are worked out in further detail, navigational architecture is defined, visual, interaction and aesthetic design schemes are created, UI design is created for selected interaction contexts and portions of the UI are programmed and implemented. The workflow is described in *Figure 1: Conceptual procedure for applying ACD process.*

Figure 1: Conceptual procedure for applying ACD process



During the **Essential Purpose and Preconception** definition step the main goal is to define the user goals and assume key features and capabilities of the system. Key stakeholders (managers, users, developers) should be involved in framing the purposes of the application from the business perspective and from the point of view of external users. Business and user goals can be prioritized, however they are treated as a wish list at this point and not as finalized requirements.

During the **Exploratory Modeling** step the team aims to identify questions, ambiguities, and risks. This is done by sketching out rough versions of roles and tasks models that will be developed in more refined form later. The goal is to reveal missing information.

After the preliminary steps described above, multiple **Iterations** are run to get to a complete product. During each iteration the following activities take place:

- **Role modeling** – during the first iteration all user roles are identified and prioritized. Then an initially targeted subset is selected to work out the next steps with. During following iterations, role models are reviewed and revised if necessary, and a next subset is selected.
- **Task modeling** – during the first iteration an inventory of all tasks is built, they are prioritized and an initial subset of tasks is selected. In each subsequent iteration, a new subset of tasks is selected and worked out.
- **Task clustering** – during the first iteration all tasks are grouped by affinity and a draft of the overall navigation architecture is created. The **navigation architecture** specifies how the

overall UI is organized into interaction contexts, collections, and groups, how these are presented to users, and how users navigate among these. For example the navigation architecture might specify that parts of the UI will be organized in tabbed notebooks, each tabbed notebook has an action bar at the top, which gives the user access to different actions on the data presented within the tabbed notebook, data represented in each tabbed notebook is organized in a "master-details" mode, etc. During following iterations the navigation architecture is revised as necessary.

- **Design –** during the first iteration a draft of the **visual and interaction scheme** is created. The visual and interaction scheme is a sort of an abstract style guide. It briefly describes the recurrent visual elements that will be used throughout the design as well as the common layouts, visual arrangements and templates that apply to various interaction contexts. For example, the schema might specify a basic layout grid with a modified tree-view in the left-most column for primary navigation and a set of view controls across the top for secondary navigation. At a more detailed level it might specify that a distinct color should identify all editable fields, which can be edited in place by double clicking. In later iterations the visual and interaction scheme is revised and refined.
- **Abstract prototyping** – during this step a content model is developed for selected interaction contexts that support selected subset of tasks. In each iteration a new portion of tasks is selected.
- **Detailed design** – during this step a detailed UI design for selected interaction contexts is defined
- **Construction –** during this step designed portions of user interface are programmed into a working product.

2.3 Data Driven Design

In DDD (Data Driven Design) designers are creators of options for users and users are sources of behavioral data. The focus is on options that are preferred by users. DDD is good for existing designs, incremental improvements, fine-tuning. It is not good for "big picture" re-thinking. The key to applying DDD is gathering data and applying data to inform future design decisions.

One of the popular approaches to data gathering and feed-back into the design is **A/B testing**. A/B testing is a randomized experiment with 2 variants, A and B (Kohavi & Longbotham, 2015). The A and B options are the control and the treatment in a 'controlled experiment' i.e. a setting where the focus is on one key variable of interest and all other aspects are kept the same. The use

of control helps eliminate experimental error and experimental bias. For example, in testing a drug on animals, the animals are split in two groups. The drug is only given to one group. Both groups are kept under exactly the same conditions so that any consequence of the drug on the animals can be attributed specifically to the drug, rather than an unrelated other characteristics of the environment.

In the development of software products or services A/B testing is used to compare two options for a (user experience) design, where often one of the options – the control option is the current design and the treatment is a proposed future design. As an example A/B testing can be used in web design to compare the effectiveness of two designs according to a specific variable of interest.

The first step in conducting an A/B test is identifying the variable. For a web site it can be anything from conversion rate (in SaaS web apps where users originally sign up for a trial with the goal to be converted to paying customers), session time, click rate (for news and media web sites which want users to click on ads or sign up for paid subscriptions). It is important that the two options are shown to similar users at the same time in order to eliminate other variables impacting the results. The option that performs better against the selected variable is selected as the future option to show to all users. For example if a certain design proves through A/B testing to generate more conversions than the current design, it is selected as the future design.

Examples of large companies applying A/B testing to online services are: Amazon using A/B testing for evaluation new home page designs, moving features around, testing new recommendation algorithms, changing search relevance rankings (Kohavi R. a., 2004), Google for making data-driven decisions on UI design and algorithms (Tang, Agarwal, Meyer, & Deirdre, 2010) and Microsoft for testing innovations and new designs at multiple sites (Kohavi R. C., 2009).

At its heart A/B testing is a quantitative method (as it captures success metrics for alternatives, without providing any insight on why the results are as they are) as well as behavioral method (as it reflects users' behavior). There are multiple sources of data – analytics, A/B tests, surveys, intercepts, benchmarks, scores of usability tests, etc. that are similar in nature. However most of those provide numeric, big data (where the quantity is pervasive). In order to make sense of the quantitative data, we need to understand its qualitative characteristics (Pavliscak, 2015). Another way to look at the quantitative vs qualitative data problem is to explore the difference between '**thin' data** and '**thick' data** (Krenchel & Madsbjerg, 2014). 'Thick' data is rich, contextualized information that you get by knowing the motivations, background knowledge of people when they take actions, in order to understand causality. 'Thin' data is data generated by capturing

traces of user actions and behaviors – what they clicked, how much time they spent, what they bought. It can be collected by devices (phone GPS, health wearables data, cookies in a browser, etc). It is most useful when the party making the experiment has a high degree of familiarity with the context in which the observed actions or behavior makes sense or when they are able to imagine and reconstruct the motivations for user actions.

There are common misconceptions about DDD (Pavliscak, 2015):

- The myth that 'thin' data is enough - actually adding 'thick' data to the mix is key to understanding what is going on.
- The myth that data (especially big data) is objective truth - actually there are common biases that apply when analyzing and interpreting big data – bias of the researcher, bias of the one sponsoring/initiating the research. Also the environment of gathering the data can influence the data – e.g. user studies often put the user in an environment that is different than their natural environment – like in the case of usability labs.
- The myth that just the size and velocity of the data is important – while taking into consideration the variety of the data and building valid categories to further track and investigate gives a more realistic understanding of the data
- The myth that the purpose of data is to prove hypothesis – while it is more about discovering patterns and trends and (iteratively) improving
- The myth that data kills innovation as it is backward looking, not suggesting of future trends, tactical rather than strategic and in many cases it seems to just skim the surface and not revealing of motivations, expectations, perceptions and emotions – while the key thing is how data is used, having in mind that most of those characteristics are indeed true
- The myth that there is a specific "right" way to use data to inform design – while the interpretation of data and the application of the gathered insights is what drives the useful (or harmful) application of data.

An online podcast talk between O'Reilly's director of market research Roger Magoulas and Arianna McClain, a senior design researcher at IDEO, provides a powerful example about story telling through data and illustrates a human-centered approach to DDD (McClain, 2015). The podcast is about how IDEO brings quantitative and qualitative data together. Arianna tells her own story of getting into data – starting during the time she was getting her PhD and was in her first biostatistics epidemiology class, in which she had to deal with an enormous government dataset that would cause her computer to crash whenever she tried analyzing it. She explains the

IDEO approach which is to stay human centered during their data journey. When the IDEO team works on projects, they use a lot of interviews in the initial stages when they need to design questions to ask people in order to gather big quantitative data. Also when analyzing data and once they have segmented data, they try to find representatives of those segments to understand better the nuances behind the segments. She told about a project for designing financial services where there were 2 segments – young entrepreneurs and seasoned business owners. If the team had taken the approach to just look at the quantitative data they would have ignored the seasoned business owners. However by combining the quantitative data with some qualitative data (interviews with some of those seasoned business owners) they were able to identify "hacks" that those business owners were using and the IDEO team was able to design for those hacks. Her conclusion is that quantitative and qualitative metrics are very interdependent – quantitative is meaningless without qualitative, but at the same time if you are only designing based on qualitative research, you are not understanding at scale what those insights mean.

Another example is revealed in a case study about Google actually going away or at least lowering the priority of DDD, which seems more suitable for incremental improvements and more into alternative design approaches that are more suitable for radical innovation, in the 10X "moon shots" style of Larry Page (Bohn & Hamburger, 2013) . The case study presents the journey that Google made in the years after 2011 from a company that makes strictly data-driven decisions about UX design to a designer-driven design decisions. The emphasis is on the role that Google CEO as of 2011 and co-founder Larry Page played into transforming the design approach at Google. They start off by mentioning the anecdotic fame that Google had about relying strongly on A/B testing to do some incremental improvements like the "41 shades of blue for links" (Divine, 2009). Jamie Divine, an ex-design-lead at Google shares thoughts on Google's approach to using data in driving design decisions - "data eventually becomes a crutch for every decision, paralyzing the company and preventing it from making any daring design decisions." He concludes "I'm thankful for the opportunity I had to work at Google. I learned more than I thought I would. … But I won't miss a design philosophy that lives or dies strictly by the sword of data". The case study then describes a more recent change in direction that Larry Page started when he took full control as a CEO in 2011, which led to multiple nice examples of consistent, good looking apps that Google released after 2011 – YouTube, Google +, Gmail, Maps, etc.

2.4 Systems Design

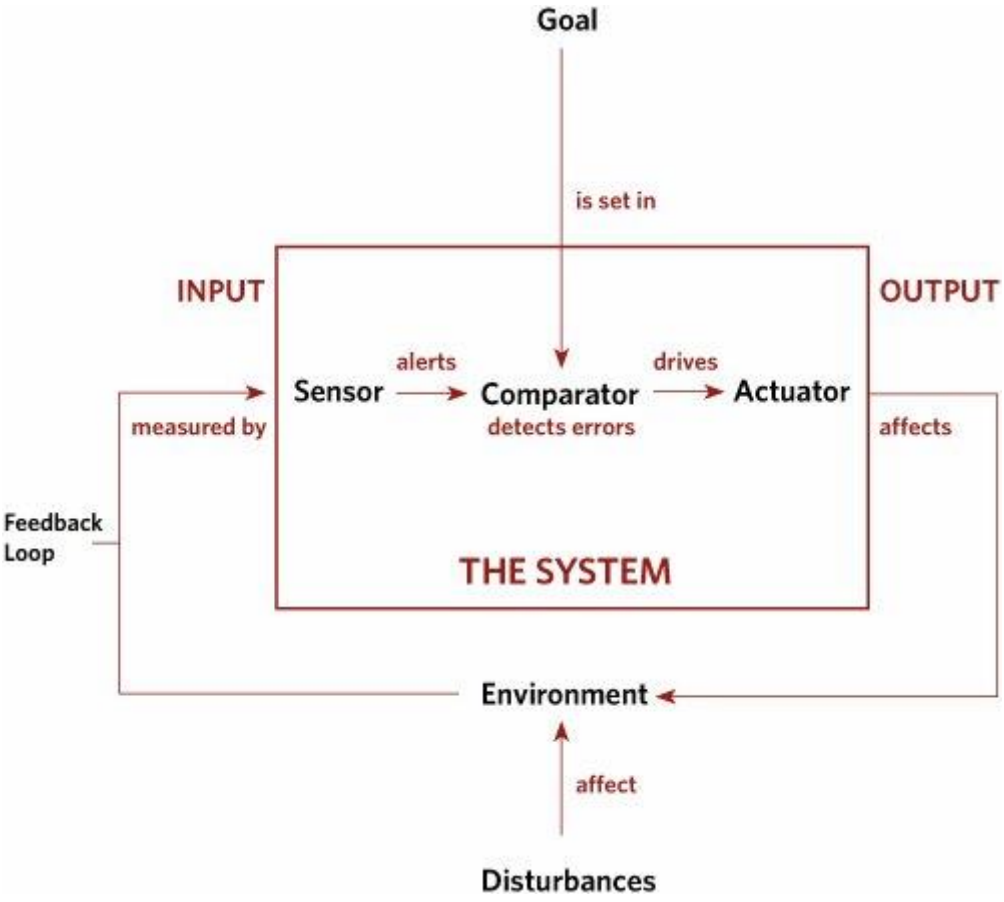
In Systems Design the focus is on the components of a system. Users set the goals of the system. Designers make sure all the parts are in place. Systems design is good for large scale designs,

design of systems of systems. It is applicable when a large team is tasked with designing a complex system and it is not a good fit for small projects. It is very analytical in nature.

The system can be more than a software product or service and even more than a computer system. It can involve people, devices, machines and objects. The complexity of system can range widely. The focus is on the context in which the design problem exists, which is also the context in which the system will be used. Systems design does honor user goals, through translating them into system's goals, but the role of the user themselves is de-emphasized. (Saffer, 2006).

The main components of the Systems Design are defined at the high level as a sensor, comparator, and actuator. The context of a system is called "environment" and it is very important to be able to draw the border between the "system" and the "environment" clearly. A conceptual system is presented in *Figure 2: Conceptual Components of a System* (Saffer, 2006)

Figure 2: Conceptual Components of a System



A classic example of a system is the heating system of a house. The **goal** of the system is to maintain a certain temperature at the house. The system goal is derived from user goals and describes the relationship between the system and the environment. The **environment** is the context, the surroundings in which the system "lives". In the example with the heating system of a

house the environment is the house. The **sensor** is a thermostat with a thermometer that is able to detect changes in the environment, in this case temperature changes. Changes in the environment themselves are called "**disturbances**" and can be expected (i.e. in expected range) or unexpected (i.e. in an unexpected range). The **comparator** is the element of the system that is responsible for comparing the current state of the environment against the goal. Every difference between the current state and the desired state is considered an error that needs to be addressed. The comparator can be a simple computer that compares the information about the environment coming from the sensor (i.e. the current temperature) against the goal (i.e. the desired temperature). The **actuator** is the component of the system that is responsible for making changes to the environment. It gets a command from the comparator, whenever the comparator identifies a discrepancy between the input coming from the sensor and the goal. In our example, the actuator is the boiler – the heating system unit that makes heat. **Controls** are mechanisms for making changes to the system (not to the environment) – e.g. setting the desired temperature as a goal.

The main questions to ask, when applying a systems design approach are (Dubberly, 2006):

- For this situation, what is the system?
- What is the environment?
- What goal does the system have in relation to its environment?
- What is the feedback loop by which the system corrects its actions?
- How does the system measure whether it has achieved its goal?
- Who defines the system, environment and goal and who monitors it?
- What resources does the system have for maintaining the relationship it desires?
- Are the resources sufficient to meet the system's purpose?

Systems design is suitable when tackling complex problems and can be an overkill in designing simple systems by solo designers. By setting a predefined set of components of a system, it provides a very structured and analytical design methodology.

2.5 Genius Design

Genius Design focuses on the skill and wisdom of the designer. Users are a source of validation (often via usability testing). Designer is the source of inspiration. Genius Design is suitable for rapid projects and gives a possibility to get a "purer" vision for the product and opens a possibility for a radial jump. However it should be used by experienced designers and requires lots of domain knowledge. This design can be drastically wrong as well.

Genius Design is an approach which relies heavily on the talent and decisions of the designer. This does not come to say that it ignores users. However it does not directly involve users in the design process, but rather than that relies on the designer's understanding of user wants and needs. This design approach works best with experienced designers, who have a lot of past projects to draw from. Sometimes it can be helpful if the designer qualifies as a user as well. However there is a caveat related to the fact that the designer has much deeper knowledge of the internal workings of the system than end users will have and thus they can fall into the trap of neglecting the need to create a mental model for the end users. (Saffer, 2006)

An example of Genius Design practice is Apple, where most of the new products are designed without the involvement of users. There are positive and negative examples, where the iPod is on the positive side, while the Newton, one of Apple's first handheld devices, is a negative example, measured by products' success on the market place.

The advantages of Genius Design compared to other approaches is that it is easy, fast and flexible. It enables designers to follow their instincts and innovate more freely. However the excessive reliance on designer skills can turn into a drawback when the designer lacks experience or just has the wrong instinct.

2.6 Comparison of Design Approaches

Based on the above descriptions of the main characteristics of design approaches, we now want to put a more systematic view of them. Getting back to the Purpose and Research Questions, we have already started answering the question "Which are the ways to approach innovative software products/services development and what are the specific methods that can be applied?" At this point we have only looked into general approaches and we will further expand the answer to this question in the next section *Drill-down into User Centered Design Methods*. Our next step would be to make an attempt at answering the question "How should one decide which approach and methods to use? What are the criteria that one should take into consideration when choosing an approach and a specific method?"

In this chapter, we introduce a proposal for a framework for comparing the five main design approaches. We define 10 different criteria and we build a "profile" of the design approaches based on those criteria. For the sake of simplicity, we assume that all criteria have equal weight and a relatively equal contribution to defining the "profile" or the "visual representation" of a design approach.

In the first step, we define the criteria. Taking into consideration the main characteristics of design approaches and what really identifies them, one can conclude that the following criteria are the "Top 10" important differentiators:

- **Explicit involvement of users** (no involvement -> heavy involvement) – or to what extent the design team directly involves end users into the design process. One extreme is the Genius Design approach, where a lack of any involvement of users is acceptable as the focus is on the talent, insight and creativity of the designer. At the other extreme are the User Centered Design and the Activity Centered Design approaches, where users are directly involved not only as validators of designs, but in the "fuzzy front end" of the creative process.
- **Time of involvement of users** (never -> early involvement) – or when (if at all) end users are involved in the creative process. The continuum ranges from "never" to capture cases where the user is not a required participant in the creative process, goes through medium values such as "users are involved in late stages as validators of design" and gets to early user involvement.
- **Designer role** (translator of input -> creator) – this criteria captures the creative responsibility of the designer with regards to being a "sole creator" of the design and concepts
- **Level of complexity** (low -> high) – measures the potential of the design approach to cover complex systems, or produce changes that have non-trivial impact to the user experience
- **Process** (sequential -> iterative) – this criteria measures the nature of the process, which can be linear/sequential as one extreme or based on iterations and feedback loops
- **UX coverage** (partial -> holistic) – is the extent to which the design approach covers fully the interaction of the user with the system in the problem domain
- **Level of formalism** (informal/experimental -> abstract/systematic) – differentiates design approaches by the level of systematic approach and formalized structure to the process
- **Innovation potential** (incremental -> radical) – is the ability of the design approach to lead to a radical innovation as opposed to producing incremental improvements or changes
- **Data orientation** (quantitative -> qualitative) – measures whether the data gathered during the process of applying the design approach is quantitative as opposed to qualitative.
- **Risk of "getting it wrong"** (low -> high) – looks at the risk for the design approach to produce results which will not be successful on the marketplace due to not taking into considerations users' actual and perceived needs.

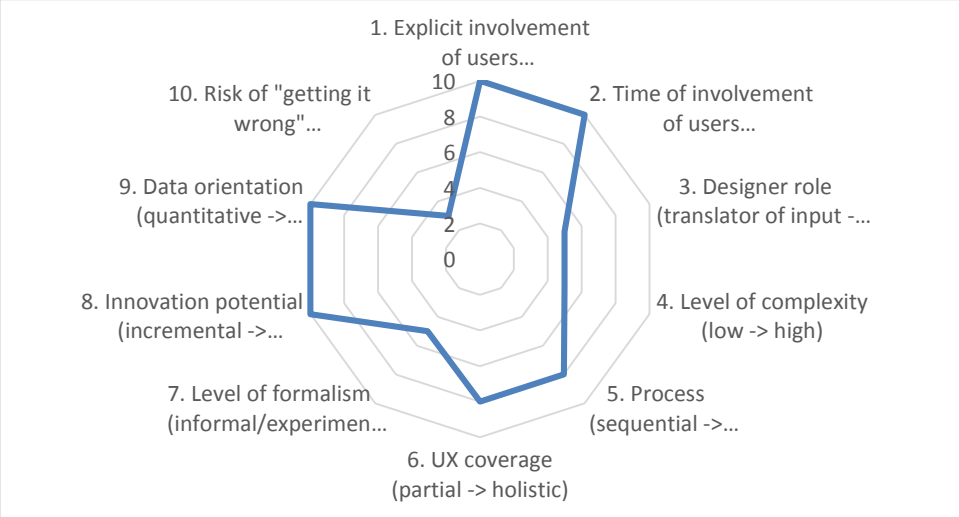
In our next step, we use the above criteria, and assess each approach against each of the criteria. *Table 2: Ratings of Design Approaches against the Criteria* presents the ratings of all design approaches against the criteria.

Table 2: Ratings of Design Approaches against the Criteria

Criteria	UCD	ACD	DDD	System Design	Genius Design
1. Explicit involvement of users (no involvement -> heavy involvement)	10	10	1	1	1
2. Time of involvement of users (never -> early involvement)	10	10	10	3	1
3. Designer role (translator of input -> creator)	5	8	3	10	10
4. Level of complexity (low -> high)	5	10	1	10	5
5. Process (sequential -> iterative)	8	8	5	1	1
6. UX coverage (partial -> holistic)	8	10	1	10	10
7. Level of formalism (informal/experimental -> abstract/systematic)	5	10	1	10	1
8. Innovation potential (incremental -> radical)	10	10	1	8	8
9. Data orientation (quantitative -> qualitative)	10	10	1	5	5
10. Risk of "getting it wrong" (low -> high)	3	5	1	1	10

Our next step is to use those ratings and create a “profile” or a “visual representation” of each design approach. Such representation will help us then to do further analysis, by visually comparing the design approaches and exploring some interesting patterns and outcomes. Such a common and uniform representation helps us bring the design approaches into one “big picture” where one can clearly see their differences and identify possible combinations that complement each other.The following figure presents visually the “character” of User Centered Design approach.

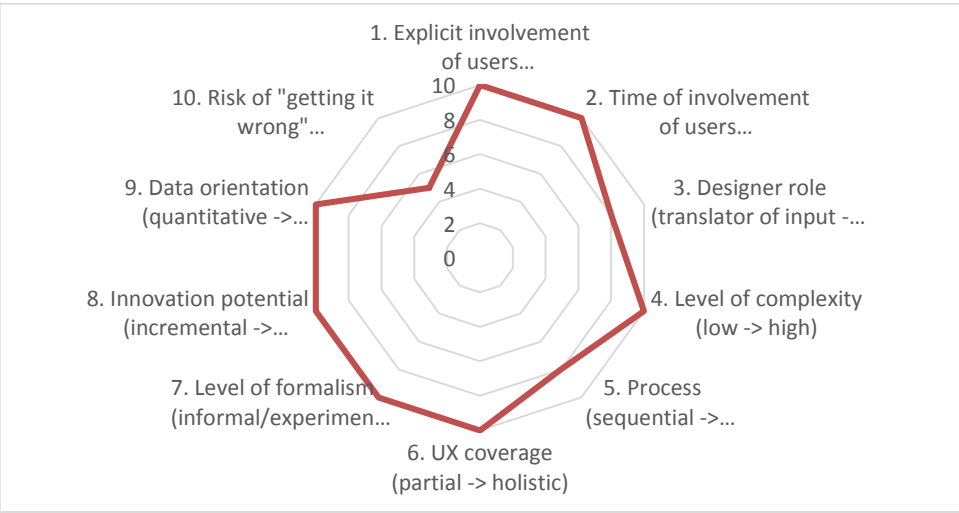
Figure 3: UCD "Profile"



As visible in the figure, User Centered Design is an approach that is very heavy on explicit and early involvement of users, where the designer role is more to translate user input rather than be a “sole creator”, where the process is very iterative, holistic, and experimental and takes into consideration qualitative data, thus minimizing the risk of low user acceptance. It has a low level of formalism and a rather complete coverage of the user experience. However the role of the designer is not too emphasized – designers are mostly translators of user needs. Furthermore the level of complexity of interactions that this design approach handles is not extreme.

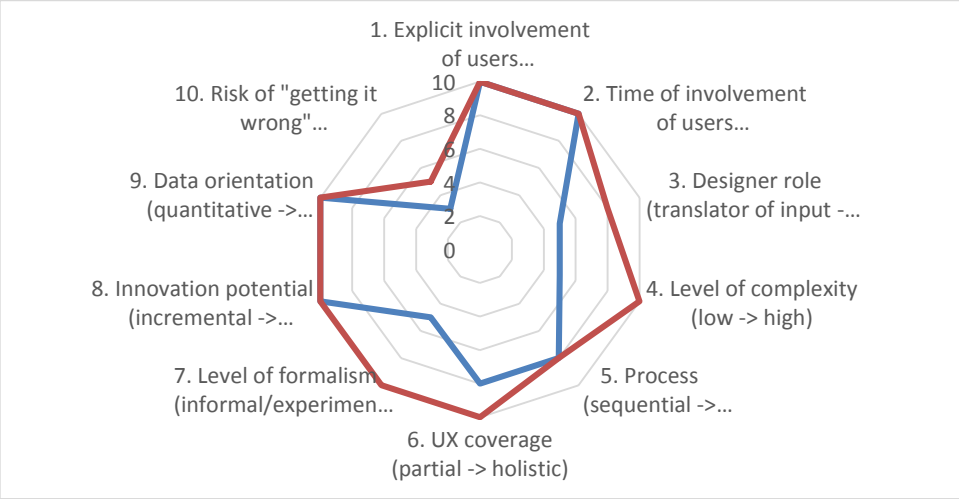
Figure 4: ACD "Profile" visually describes the characteristics of the Activity Centered Design approach – one that does not only involve users heavily and early, but is more systematic and formal, gives more creative pressure to the designer compared to UCD, looks even more holistically at the UX than UCD and has qualitative data orientation and a great innovation potential.

Figure 4: ACD "Profile"



A visual comparison between the two approaches described above (UCD and ACD) makes an obvious take away (*Figure 5: Comparison between UCD and ACD*): ACD is a more complete method that is not alternative to UCD, but adds some important characteristics (more holistic view through the additional focus on activities and formal and systematic process).

Figure 5: Comparison between UCD and ACD



Moving on to the Data Driven Design, another interesting outcome is revealed by *Figure 6: DDD "Profile":* Data Driven Design is very light on almost all of the criteria. It is very quantitative, does not have a potential to generate radical innovation, and involves users in a very indirect way (through collecting quantitative data about their behavior without explaining their motivations or emotions). Being based on “cold facts”, it eliminates the risk for making a wrong decision, especially about the local/isolated characteristics of the product being explored. However it does not help much in understanding the “big picture” or envisioning a radically new product.

Figure 6: DDD "Profile"

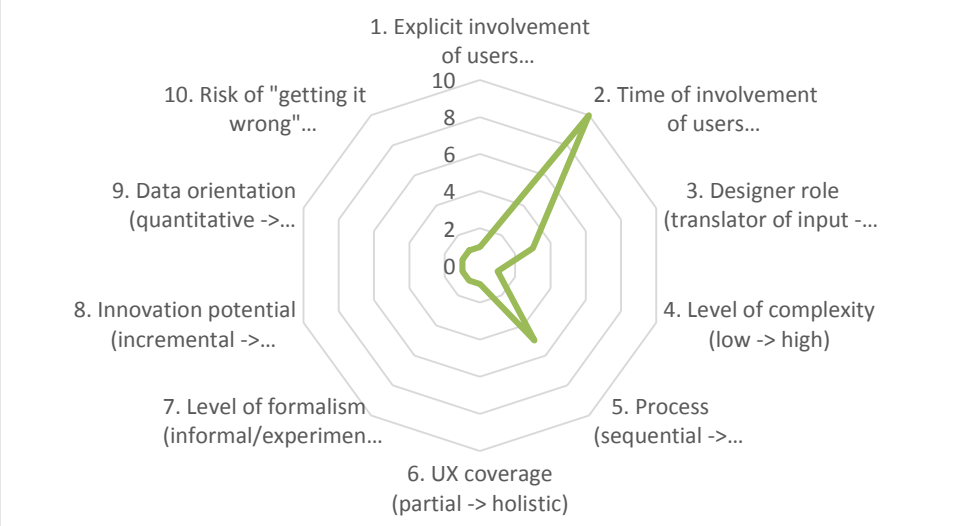


Figure 7: Comparison between UCD and DDD demonstrates the limitations of the Data Driven Design approach, compared to User Centered Approach.

Figure 7: Comparison between UCD and DDD

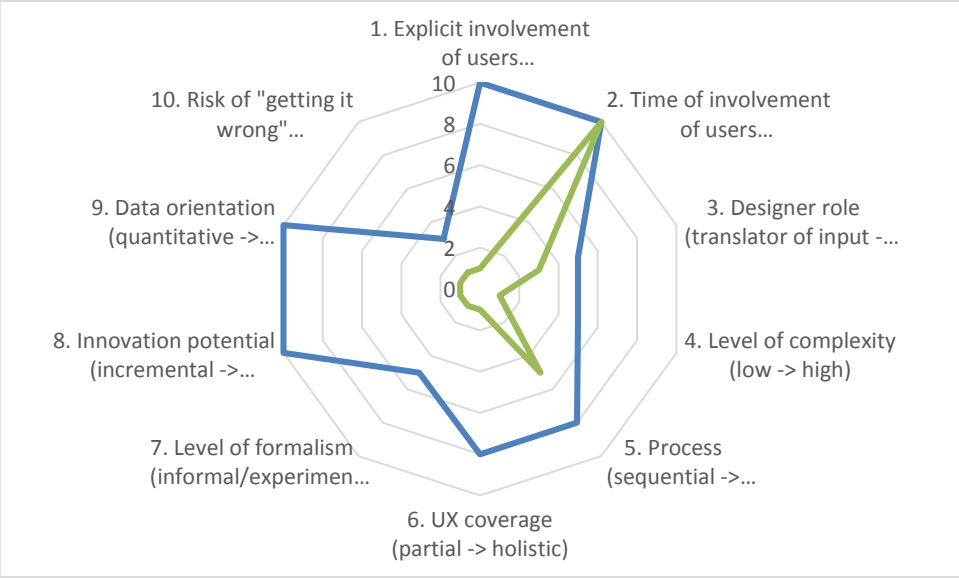
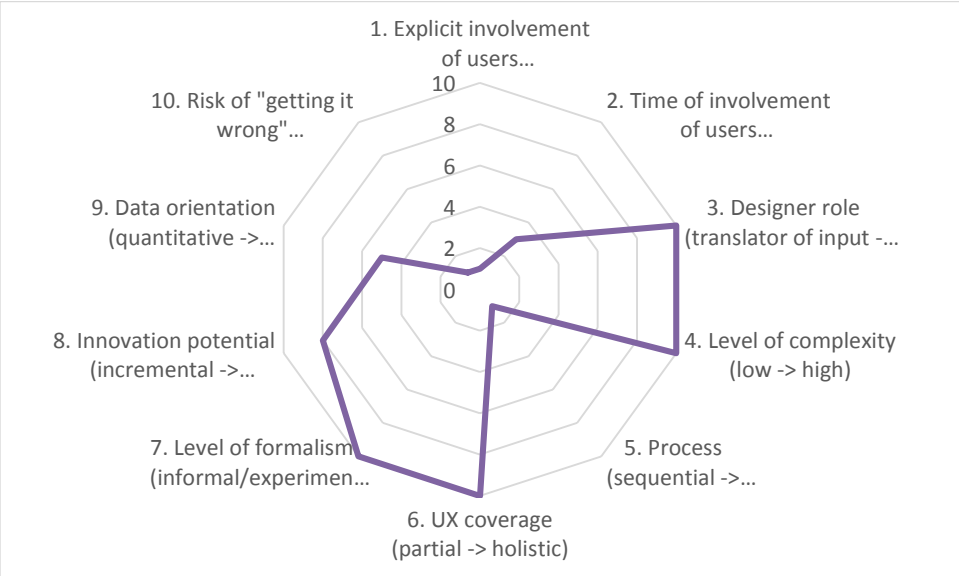


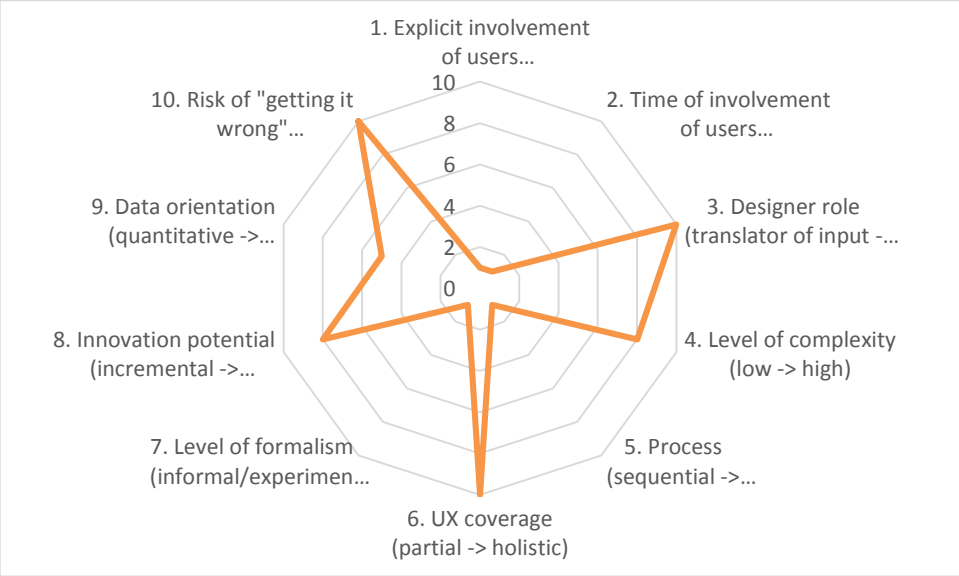
Figure 8: System Design "Profile" shows the characteristics of the Systems Design approach. Given that it is most applicable for designing “systems of systems”, when the level of complexity is very high and it is highly formal and systemic, the “inclination” towards criteria like “designer role”, “level of complexity”, “holistic coverage” and “level of formalism” is very heavy as opposed to almost no inclination towards involvement of users.

Figure 8: System Design "Profile"



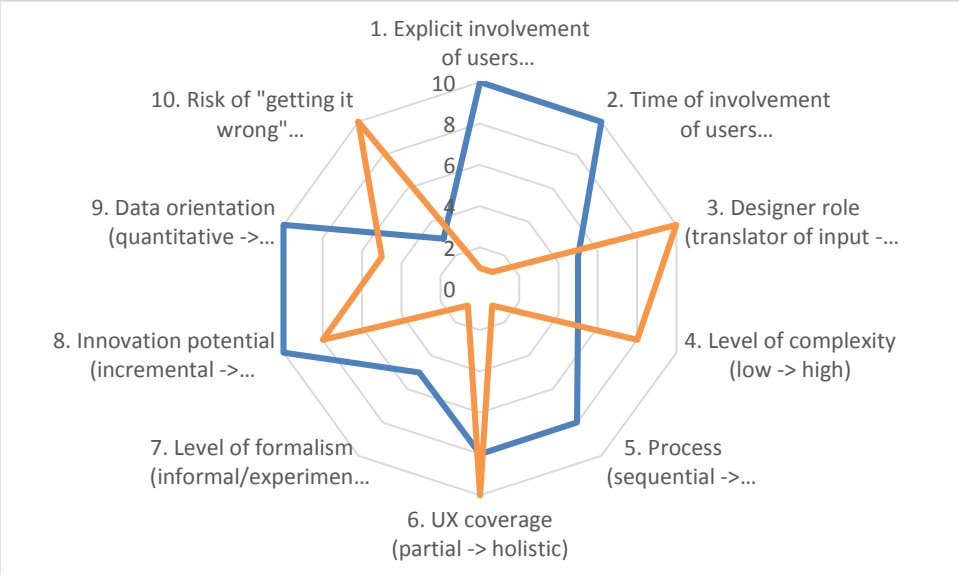
Another “outlier” that is very heavy on some criteria (e.g. innovation potential, emphasis on the role of the designer, potential for handling extremely complex interaction) and very light on other criteria (e.g. explicit involvement of users, level of formalism, data orientation or iterative process) is Genius Design, which “profile” is visualized in *Figure 9: Genius Design "Profile"*.

Figure 9: Genius Design "Profile"



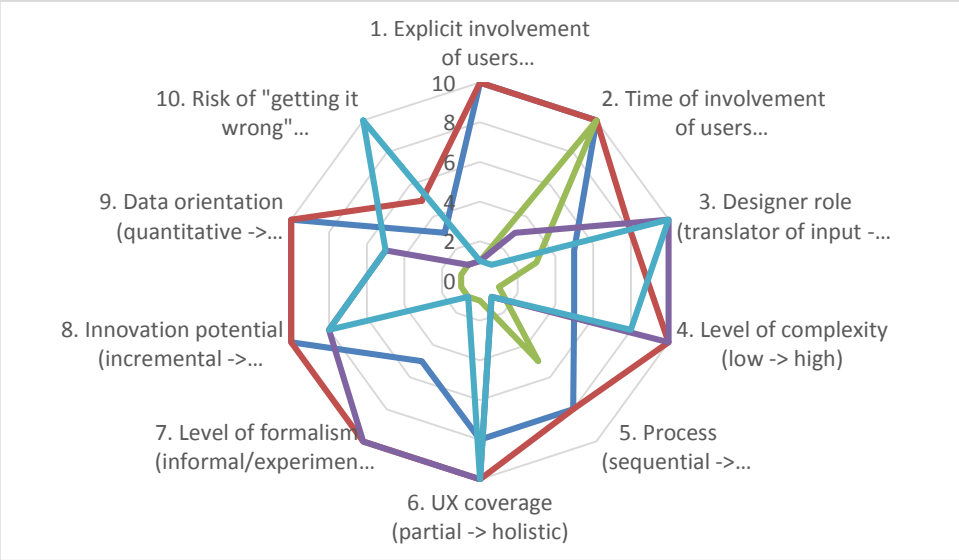
In answer to the third question in the Purpose and Research Questions section, one can use the visualization to explore some interesting opportunities for combining methods to fully cover the criteria. For example, a comparison between User Centered Design and Genius Design reveals that a combination of them can be very powerful to explore all dimensions of the design task – as illustrated by *Figure 10: Comparison between UCD and Genius Design*.

Figure 10: Comparison between UCD and Genius Design



Finally, *Figure 11: The "Big Picture" of Design Approaches* is an attempt at bringing all design approaches together in a dashboard-style view.

Figure 11: The "Big Picture" of Design Approaches



3 Drill-down into User Centered Design Methods

One discussion of different ways to apply UCD makes an interesting analogy with dancing done by dancing partners (designers and users), where in each dancing move one of the partners moves towards the other. Similarly to how dancing moves are combined and used together to form a dance, those design steps are not exclusive, they can be combined and used together. (Steen, Kuijt-Evers, & Klok, 2007) The following “dancing moves” are most often used in UCD: Participatory Design, Contextual Design, Design Thinking, Empathic Design and Lead User Method. Those “design moves” are further discussed in the following sections.

3.1 Participatory Design

Participatory Design has its roots in the early 1970, in Norway, when the industrial workplace was getting extensively computerized. Members of the Iron and Metalworkers trade unions advocated in favor of enabling workers to have bigger influence on the design and introduction of computer systems into the workplace. The process of introducing Participatory Design had political and democratizing effect. A special national codetermination agreement was founded to emphasize on the rights of unions to participate in the design and deployment of new workplace technology. (Winograd, 1996)

An example of a practical application of the Participatory Design technique is the UTOPIA project, which was jointly conceived by the Swedish and Danish researchers and the Nordic Graphic Workers' Union. The project end goal was designing a computer based toolset for graphic workers. The approach toward designing those was one of keeping control over process and methods with the skilled graphic workers, while enabling computer assistance for page makeup and image processing for newspapers.

The design philosophy, which was called internally "perspective" as described by Pelle Ehn, a primary participant in the UTOPIA: *"The tool perspective was deeply influenced by the way the design of tools takes place within traditional crafts... new computer-based tools should be designed as an extension of the traditional practical understanding of tools and materials used within a given craft of profession. Design must therefore be carried out by the common efforts of skilled, experienced users and design professionals. Users possess the needed practical understanding but lack insight into new technical possibilities. The designer must understand the specific labor process that uses a tool."* Ehn, 1992, p. 112, cited by (Winograd, 1996)"

Due to the complexity and tacit-ness of the context, novel ways of working together among the designers and the graphic designers had to be invented, in order to foster communication, mutual understanding and learning from each other.

"Requirement specifications and systems descriptions based on information from interviews were not very successful. Improvements came when we made joint visits to interesting plants, trade shows, and vendors and had discussions with other users; when we dedicated considerably more time to learning from each other, designers from graphics workers and graphics workers from designers; when we started to use design-by-doing methods and descriptions such as mock-ups and work organization games; and when we started to understand and use traditional tools as a design ideal for computer-based tools." Ehn, 1992, p. 117, cited by (Winograd, 1996).

While Participatory Design was gaining popularity and companies were realizing the importance of designing with the user rather than just designing for the user, Participatory Design researchers have created a variety of techniques to facilitate the communication of new technology possibilities to domain experts (craftsmen) and to help them get a feel of what it would be like to work with an envisioned system. Such techniques include low-fidelity and high-fidelity mock-ups and role plays.

Originally the Participatory Design movement proclaimed that user involvement is heavy on all dimensions, which could be used to measure user participation: directness of interaction, length of involvement during the design process, scope in the overall system and degree of control over

design decisions. This was largely due to the fact that Participatory Design was an embodiment of industrial democracy. However, recently and specifically in the development of software products, user involvement may not be that high in all of the dimensions.

3.2 Contextual Design

Contextual design is a UCD process, which provides structure and guidance throughout the whole process of designing a product or service. It shares some techniques with Participatory Design, but has different characteristics. It provides guidance how to gather user data in the field, how to interpret and aggregate this data, how to prototype product or service concepts and how to test them with users. At the core of the Contextual Design philosophy is understanding users' intents, desires and drivers and this is achieved through going out in the field, observing and talking to users. (Holtzblatt & Beyer, 2014)

Contextual Design is based on several disciplines – anthropology, psychology and design and is primarily used in designing computer information and IT systems. Parts of it have been used as field usability evaluation method (Mcdonald, 2006). Contextual Design has been applied to a variety of IT systems, including learning tools and technologies (Notess, 2005).

Contextual Design is based on a set of key principles which define its use as a design tool:

Principle1: System design must support and extend users' work practice. This principle is based on the belief that in order for a product or service to be easily understood, accepted, used and valued, it needs to be explicitly designed to support and extend its users' work practice. If it is not designed this way it will cause frustration, dissatisfactions and users will work around it. The meaning of "work practice" is somewhat broad and relates to the set of attitudes, behaviors, goals and intents that characterize a set of users in a particular environment. It is not necessarily related to "work" and can be associated with life practices like driving, buying, playing music, communicating with other people, etc.

Principle 2: People are experts in their work practice. However they are not able to articulate their work practice. This principle states that a major part of the knowledge that people apply in their workspace is tacit i.e. they are not consciously aware of it. Furthermore, there are a lot of seemingly unimportant details, workarounds, coping strategies associated with work practice that are actually very useful for systems design. Even though proficient in their everyday jobs, because of the tacit nature of their knowledge, people are unable to articulate all aspects of their work practice, especially when taken out of the context. This is where the main concept of

going into the field, into users natural context comes from and what drives the "Contextual" part of the name of the Contextual Design technique.

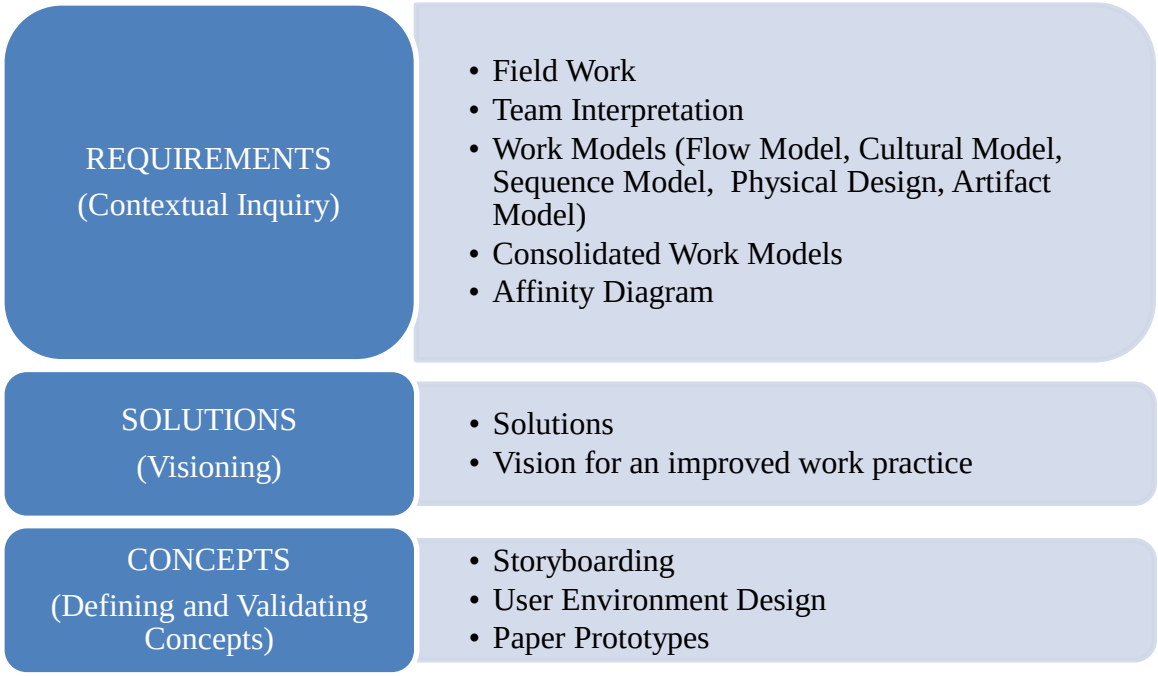
Principle 3: Good design involves users' participation. This is also the core principle of the user centered design approach in general. The Contextual Design approach calls for not just observing users in the context of their work practice, but also actively involving them in discussions, analysis and reflection of their values, intents and practices. The interview process itself does not use formal methods like surveys and advance prepared questionnaires. It resembles the master-apprentice relationship, where the interviewers strives to gain a deep understanding of the work process. The initial phase of gathering data is not the only point where users are actively involved. They also actively participate in the later phase of paper prototypes and short iterations on coming up with detailed designs.

Principle 4: Good design is systemic. This principle reflects the understanding that cohesiveness and consistency of design are key as design impacts the user through its entirety. The design of a mobile operating system carries certain visual characteristics and capabilities through all applications; the design of all elements of a car interior is driven by the same aesthetic and functional requirements; etc. The different artefacts in the Contextual Design process ensure coherence at different levels – the Vision ensures coherence of high-level direction, the User Environment Design ensures structural coherence, the storyboards ensure coherence at task level. This principle encourages designers to look holistically at the system design, rather than solving its different aspects as independent problems.

Principle 5: Design depends on explicit representations. This principle mandates the use of explicit, physical, tangible representations of ideas. Each technique in the Contextual Design process is associated with certain artefacts that facilitate doing the work, capturing the results and sharing and discussing with others.

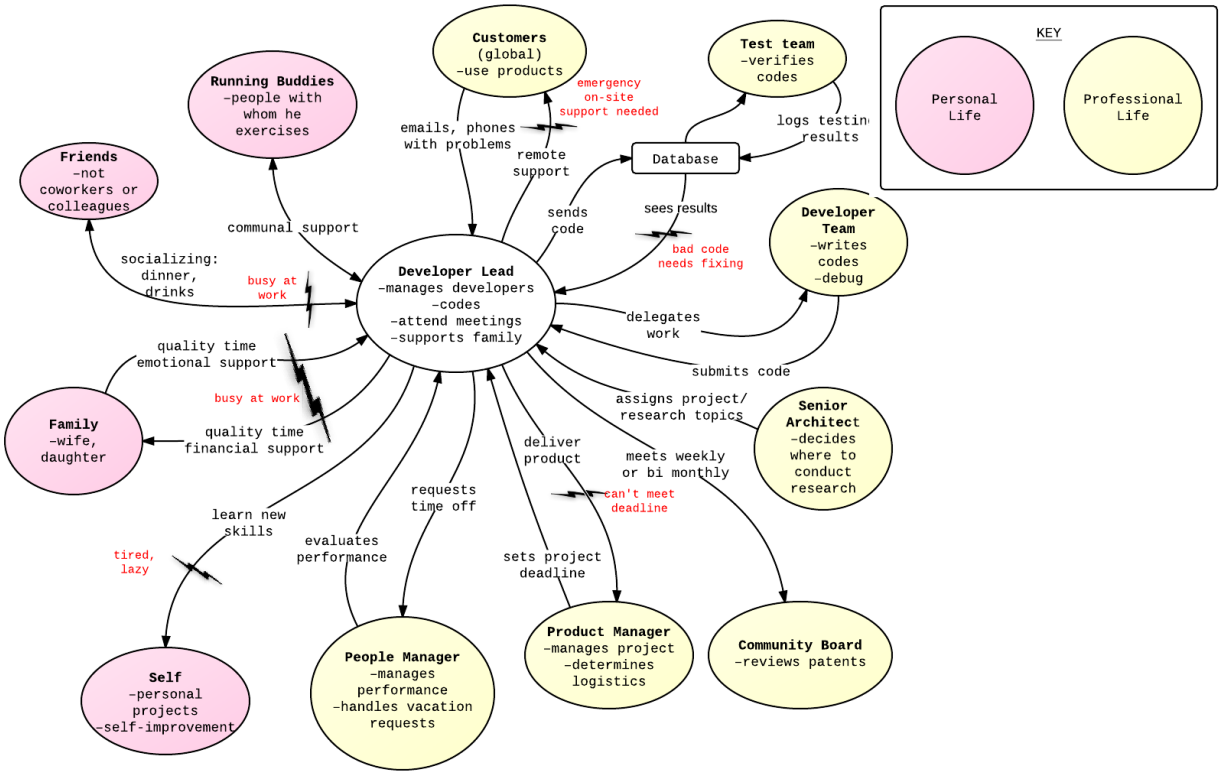
The Conceptual Design process is associated with multiple steps broadly grouped into two phases – the phase of defining requirements and solutions and the phase of defining and validating concepts. The steps in the phase of defining requirements and solutions helps characterize what customers do (i.e. what is important to them) and envisioning new ideas and direction. Customers are a broader group than end users, who directly use the product or service. Customers also include indirect users, managers of users, providers of funds, etc. The focus with Contextual Design is mainly on end users. The overall Contextual Design process is described in *Figure 12: Contextual Design Process*.

Figure 12: Contextual Design Process



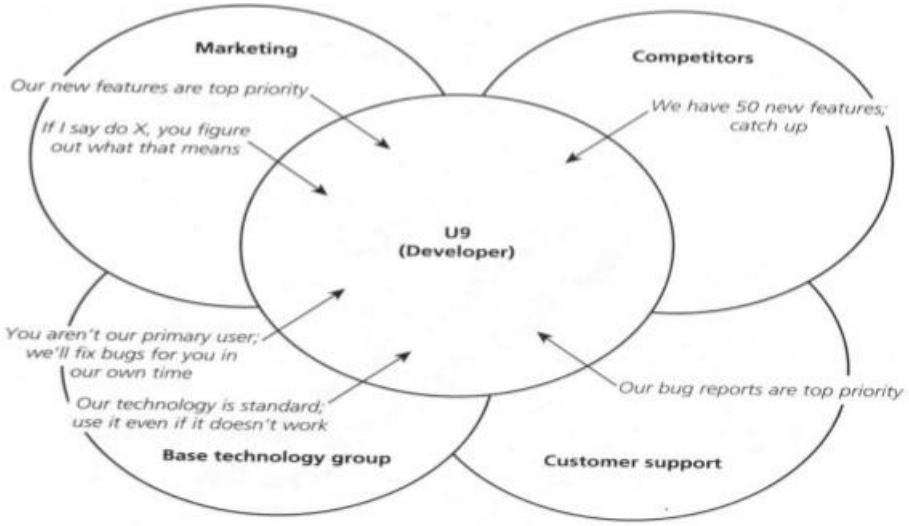
The first step in defining requirements and solutions is talking with specific users in the filed a.k.a. **Contextual Inquiry**. The Contextual Inquiry is an extremely important step to understand who the end users really are and what how they work on a day-to-day basis. This is done through asking questions about what the user is doing and why they do it as they are being observed doing it. It is not a questionnaire that is fixed and defined in advance. This inquiry process done in the field, or in the context of the user work practice is followed by a **Team Interpretation** step. The Team Interpretation session is where the cross-functional design team gets together to go through the inquiry results and distill the learning and insights. Everyone in the team is encouraged to share their interpretation of the data and the possible product, technical, marketing implications. This is how the design team builds a shared understanding of the users' world. Once the insights and learning have been discussed, they have to be captured in **Work Models**. Work models are formal descriptions of the work process through different perspectives by using diagrams. Different types of work models capture different aspects of the process. The **Flow Model** captures the coordination among individuals to achieve a goal. It reveals formal and informal groups, patterns, roles and responsibilities. An example flow model of the work of a developer lead is shown in *Figure 13: Example of a Flow Model in Contextual Design*.

Figure 13: Example of a Flow Model in Contextual Design



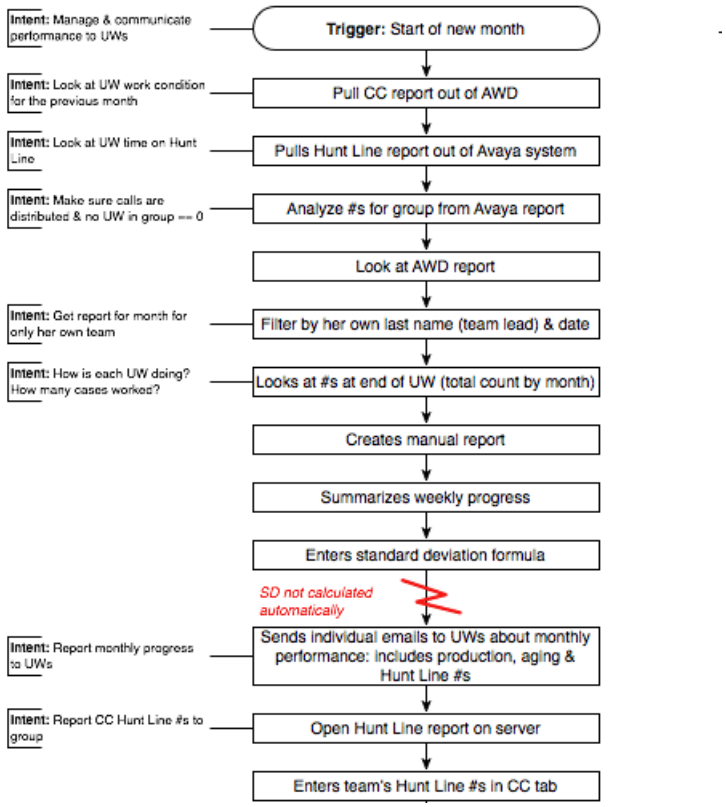
The **Cultural Model** captures culture and policy aspects that limit how work is done and also depicts workarounds to those limitations that users apply. The constraints that a developer meets are described in *Figure 14: Example of a Cultural Model in Contextual Design*.

Figure 14: Example of a Cultural Model in Contextual Design



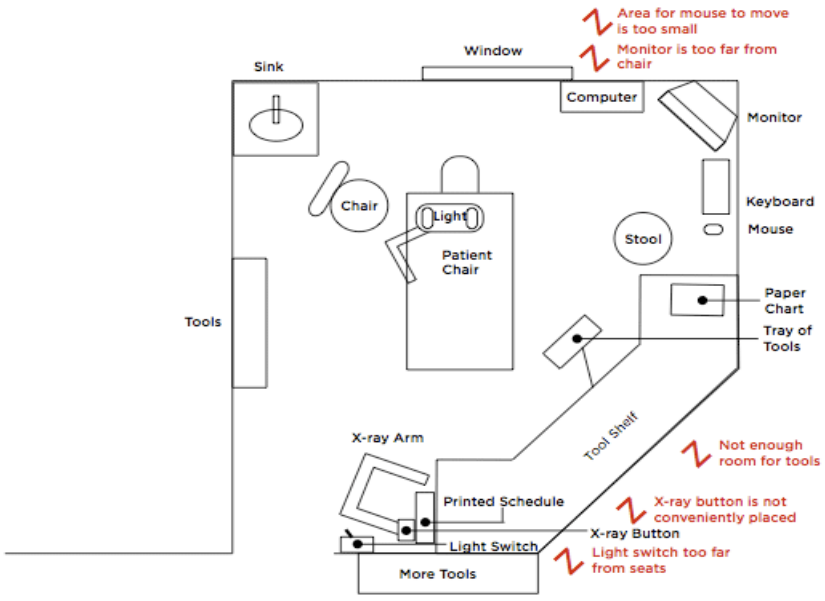
The **Sequence Model** captures the sequence of tasks that users do to accomplish their goals. It also captures different strategies as well as possible issues getting in the way (as described in *Figure 15: Example of a Sequence Model in Contextual Design*).

Figure 15: Example of a Sequence Model in Contextual Design



The **Physical Model** describes the physical environment in which users perform their work and how it is organized (see the following figure).

Figure 16: Example of a Physical Design in Contextual Design



Finally, the **Artifact Model** shows the various artifacts that are created and used during the users' work practice. The artifacts reveal the concepts related to the users' work. An Artifact Model is presented in *Figure 17: Example of an Artifact Model in Contextual Design*.

Figure 17: Example of an Artifact Model in Contextual Design

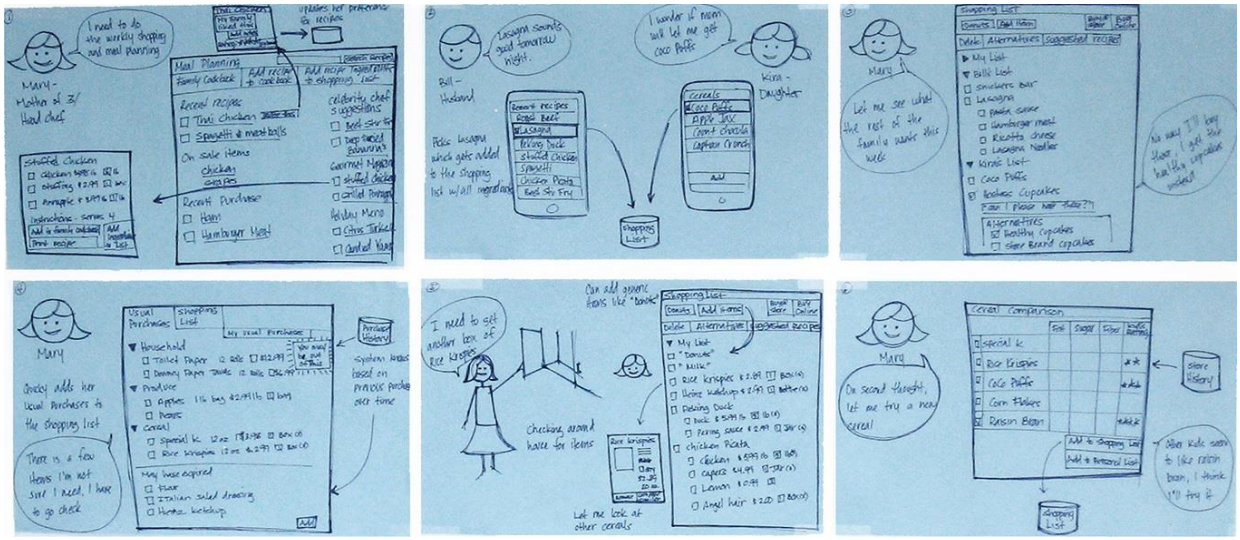
After different model diagrams have been constructed for individual users, two aggregations of the data are built: **Consolidated Work Models** and an **Affinity Diagram**. The consolidated work models are built through bringing together the different instances of each type of work model build for different users. The affinity diagram is a wall-sized hierarchical diagram that reveals the full scope of the problem. Another artifact of the process is **Personas** that are created using contextual data to create a more live image of the user. The caveat with Personas is that they need to be built with very rich data in order to be useful. If they are not based on rich data, they can be misleading.

All of the steps described above were related to gathering, consolidating, representing data about users and their work practice gathered in the field. Once all this is done, comes the design response to all this data – the **Visioning** step. During the visioning step, once the problem has been understood well, the focus of the design teams moves to defining solutions, using technology and business process design to make the work practice more effective and efficient. The **vision** is a high level description of a new work practice that sets a general direction without aiming to flesh out details. It enables the team to grasp the overall structure and direction of the solution and to continue working out the details in a consistent and cohesive way.

Contextual Inquiry and Visioning are related to gathering data and defining problems and solutions. The next phase is **Defining and Validating Concepts**. They take the vision as the basis for the new system and transform it from a high-level direction to a detailed description of the

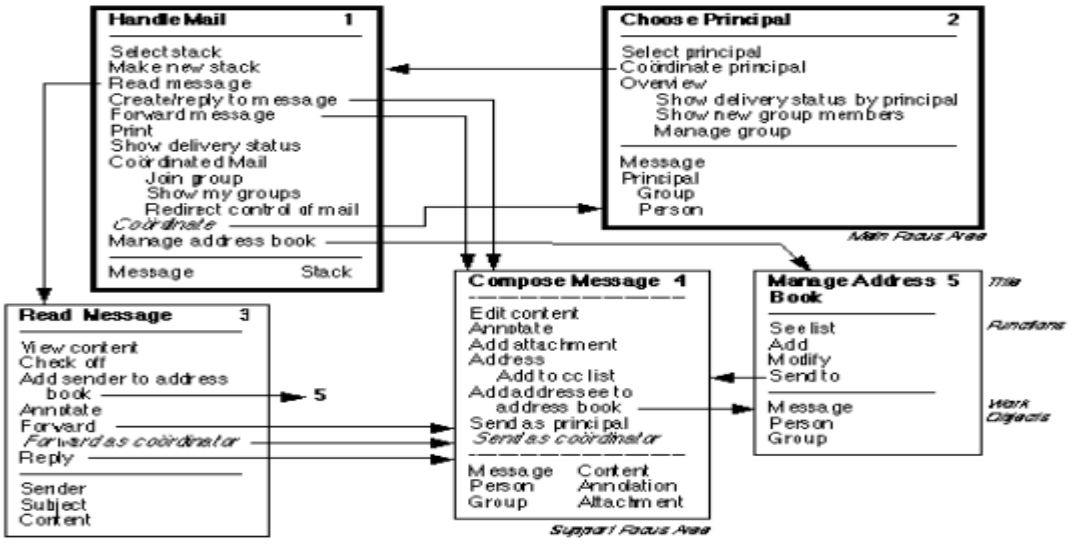
desired behavior, function and structure. The first step in this phase is the **Storyboarding** step. Storyboards are descriptions of how users will accomplish a specific task in the new system. It outlines the steps that the user and the system take as well as links between users or between multiple systems. Storyboards capture the individual tasks flow and ensure coherence between tasks, as described in *Figure 18: Example of Storyboards in Contextual Design*.

Figure 18: Example of Storyboards in Contextual Design



The **User Environment Design** captures the overall structure of the system and ensures that the system supports a natural flow of work (example in *Figure 19: Example of User Environment Design in Contextual Design*). It puts all parts of the system together, the functionality/capabilities each part provides and explains how the user navigates between parts of the system. The User Environment Design is not bound to any particular user interface design.

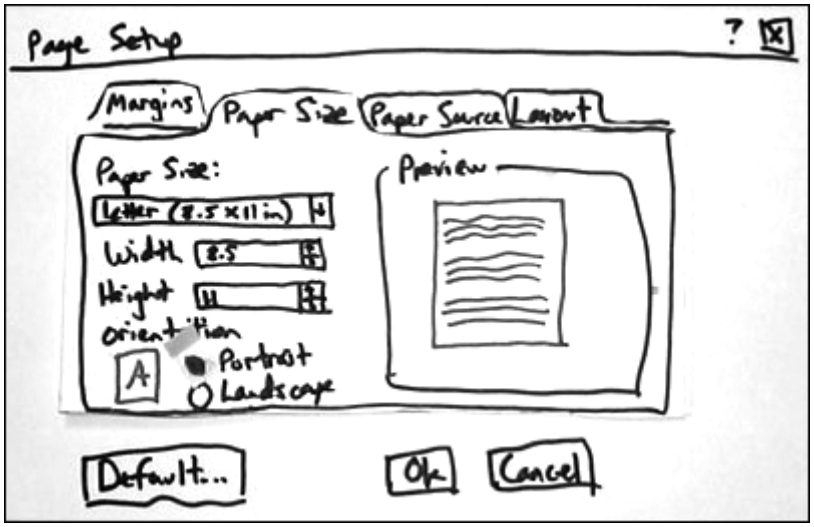
Figure 19: Example of User Environment Design in Contextual Design



As the Storyboards and the User Environment Design outline the design of the new system, before going to detailed designs or implementation, the generated solution needs to be tested with users. Testing needs to be cheap and quick to allow for multiple iterations to get to a detailed design.

The testing within Contextual Design is done through using **Paper Prototypes**. Paper prototypes are rough mockups of the system, produced with paper and pen, to describe the user interface of the system (as the example in *Figure 20: Example of a Paper Prototype in Contextual Design* suggests). Those are used again on user premises, to test the workings of the new system with actual users. Users use the prototypes as if they were a live system and uncover issues and gaps. Whenever such are identified, the design team works together with the user to redesign the experience. This is done to make sure that the user voice is heard best, especially when there are differences of opinions among design team members. Multiple iterations can be done with paper prototypes before the design is considered done and designers move on to sorting out the details of the design of the user interface.

Figure 20: Example of a Paper Prototype in Contextual Design



In a nutshell the Contextual Design method helps put structure in the initial phases of the design process, which is also the most creative and unstructured. It provides methods for understanding better users' work practice by observing and inquiring them in their natural context, consolidating the data, extracting learning and insights, representing them through various perspectives, coming up with a vision and solutions, creating structural and interaction design and testing those with users before going into implementation.

3.3 Design Thinking

Design Thinking is an approach that is so widely discussed in various fields (business process modelling, product development, service design) that has caused the term to be overloaded. For the sake of this analysis, we will refer to Design Thinking as the creative process of solving non-linear problems, where the analytical approach would not be effective. This creative process involves a specific skillset: both convergent and divergent thinking, prototyping and iteration, user research that is ethnographic and qualitative. This is opposed to analytical, quantitative, statistical methods.

Types of problems: (Treverton, 2007) claims there are two types of problems – puzzles and mysteries. Puzzles are problems that are easily solved if a particular piece of data is available. Mysteries however are more complex – data might be available, even too much data might be available, but that would not guarantee a solution to the problem as it involves an assessment of uncertainty. In a similar vein, (Liedtka, King, & Bennett, 2013) refers to the popular definition of a "tame" problem, analogous to the (Treverton, 2007) definition of a "puzzle" and a "wicked" problem that corresponds to the (Treverton, 2007) mystery. Tame problems are problems where there is a general agreement about the definition of a problem. There is usually relevant data available, a cause-effect relationship can be determined and we can successfully use liner processing with existing data to come up with a solution. In wicked problems, however, those conditions are not met. Often the stakeholders involved can't even agree on the definition of the problem nor on the solution. There may be abundance of data, but it is not clear whether or how any of it is relevant. The situation is often so fluid and complex that one cannot determine cause and effect. The way to check the validity of a solution is to try it out in practice. Wicked problems are human-centered. With wicked problems usually a deep understanding of people/users is required, while with tame problems a few people are involved in the problem definition or the solution. Wicked problems have a much lower level of agreement on the definition of the problem and a much lower level of confidence that the right problem is being solved than tame problems. Also with wicked problems the level of likelihood that past data could help predict the future is much lower than with tame problems.

Left-brain thinking vs right-brain thinking: Design Thinking is also associated with "right-brain" thinking, which is related to perceiving things as a whole, simultaneously, seeing the "big picture" as opposed to "left-brain" thinking which is sequential and analytical. (Pink, 2006) talks about the increasing importance of developing "right brain" skills and how the future will be driven by "right-brain" people. The idea in short is that as the left-brain skills are sequential,

analytical, linear they can be automated by computer algorithms, or outsources wherever the costs are lowest or replaced by more advanced tools. However right-brain skills are more holistic, creative, non-linear and "symphonic" and those are becoming the key differentiators in the business world today and going forward.

Design Thinking as a mindset: Design Thinking is considered not just a technique or a prescription of how to approach the creative part of the product/service development process, it is considered a manifestation of a specific **mindset**. A "learning" mindset that "promoters" (i.e. people who are motivated by aiming towards an envisioned future) exhibit. It is opposed to a "fixed" mindset that "preventers" (i.e. people who are motivated by preventing presumably "bad" things from happening) exhibit. It is a representation of a culture of fail-fast-to-succeed-soon attitude where small bets are done often, a lot of experimentation happens and the course is changed frequently to pivot through a dynamic and fluid environment. It is also a "solution-based" approach, as opposed to a "problem-based" approach. A solution-based approach to tackling a problem is one that attacks the envisioned future state, possibly simultaneously in multiple ways, rather than relying on de-constructing and analyzing the characteristics of the problem in order to get to a solution.

The attributes associated with~ Design Thinking are:

Ambiguity – this is related to the nature of the problems themselves (wicked or mystery problems) as well as to the possible solutions. During ideation no judgement is being exercised in order to prevent from the fear of failure to prevent from exploring a potentially novel and bold solution. "Out of the box" or non-traditional, wild ideas are welcome.

Collaborative – "everyone is a designer" is one of the strong beliefs in the process. People work in cross-disciplinary teams, end-users are directly involved in multiple stages of the process.

Curiosity – "learning mindset" is a key prerequisite to the success of the process. Fresh look on things, acceptance and willingness to understand things are unknown.

Holistic – paying a great deal of attention to the context. A lot of diligence is put to ensure the problem or goal at hand is not too narrowly defined.

Iterative – Design Thinking is not a linear process, but different steps can be done in parallel and revisited and repeated multiple times.

Tangible – "show rather than tell" approach when it comes to possible solutions.

The stages of the design thinking process are defined slightly differently between authors, but still there are consistent general phases and a set of steps in each phase. (Liedtka, King, & Bennett, 2013) Define 5 stages of the Design Thinking process.

Defining the problem – this is the stage when the problem statement is identified, the scope of the project is defined and a "design brief" is stated. Gavin Ambrose and Paul Harris also call this the "definition" stage in their book "Design Thinking".

Asking the "What is" question – this is the stage when the current situation is being analyzed. User research is done through ethnographical methods (observing and talking to users in their natural context), insights are extracted from this research and a design criteria is established to serve for validity checks for solutions. (Ambrose & Harris, 2009) refer to this stage as the "Re-search" stage.

Asking the "What if" question – this is the stage when the problem is re-framed in a way that triggers creative thinking and concepts, ideas and solutions start to emerge through different techniques like brain storming and co-creation. Those ideas and concepts are captured in a very rough, throw-away prototypes. (Ambrose & Harris, 2009) split this stage to "Ideation" and "Prototyping".

Asking the "What wows" question – this is the stage when a selection process is run and the list of generated solutions is shortened to several that meet the design criteria and would be later tested with actual users. (Ambrose & Harris, 2009) call this "selection" and "implementation".

Asking the "What works" question – this is the stage when one or several prototypes are tested in the marketplace and learning from the process is done to inform and improve future applications of the Design Thinking process. (Ambrose & Harris, 2009) call this the "learning" stage.

After getting into a certain stage, previous, earlier executed stages can be revisited and iterations can be run.

A specific example of applying the Design Thinking method at SAP Labs is provided later on in the Case Study of Applying Design Thinking at SAP section of this research. It is based on a live interview with a Design Coach at SAP Labs, who describes the application of the method in a project for optimizing POS (Point of Sale) process at retail stores.

3.4 Empathic Design

According to (Leonard & Rayport, 1997) the core of emphatic design is observation (how customers use products or services) that is conducted in the customer's own environment, in the

course of normal routines. Empathic Design is a way to identify (potentially critical) customer needs.

Traditional research models rarely lead to truly novel product concepts because of multiple reasons:

- Sometimes customers are so accustomed to given conditions that it doesn't even occur to them to ask for a new solution, even if they have needs that can be addressed, because they don't know it is feasible
- Often when there is inconvenience arising from the use of a product, customers tend to come up with workarounds and later on they become so used to using those workarounds that they forget the inconvenience that caused them to think of a workaround in the first place
- Natural bias of the respondent to standard market surveys to please the inquirer by providing expected answers, inclination to not reveal practices that they suspect might be deemed inappropriate
- Natural bias of the inquirer that they bring in inadvertently, but inevitably
- The natural paradox of standard research practices (interviews, surveys, focus groups) between the aim for quantitative results and the need to have open ended questions in order to fully capture the customer context.

Similarly to other UCD methods, Empathic Design involves observation of users in their **natural physical environment.**

The types of information that observation methods applied in labs can generate are - how easy/difficult it is for the user to work with the product, do they seem confused at any point, do they need to fall back to the product manual or is the product design self-explanatory, what are the unspoken assumptions that they apply when interacting with the product.

The types of information that can be generated through Empathic Design are:

- **Triggers of use** or the circumstances under which customers choose to use the product or service. If those do not fully match the expected circumstances or motivation, there might be an opportunity for innovation, product re-design or even entering completely new markets. Examples: using spray-on cooking oil for the bottom of lawn mowers.

- **Interactions with the user's environment** or how the product fits into the user's individual systems. Example – through observing first-time use of its personal finance management product

Quicken, the company Intuit discovered that owners of small businesses were using the product to run their business finances. This is something that could not be observed in a usability lab.

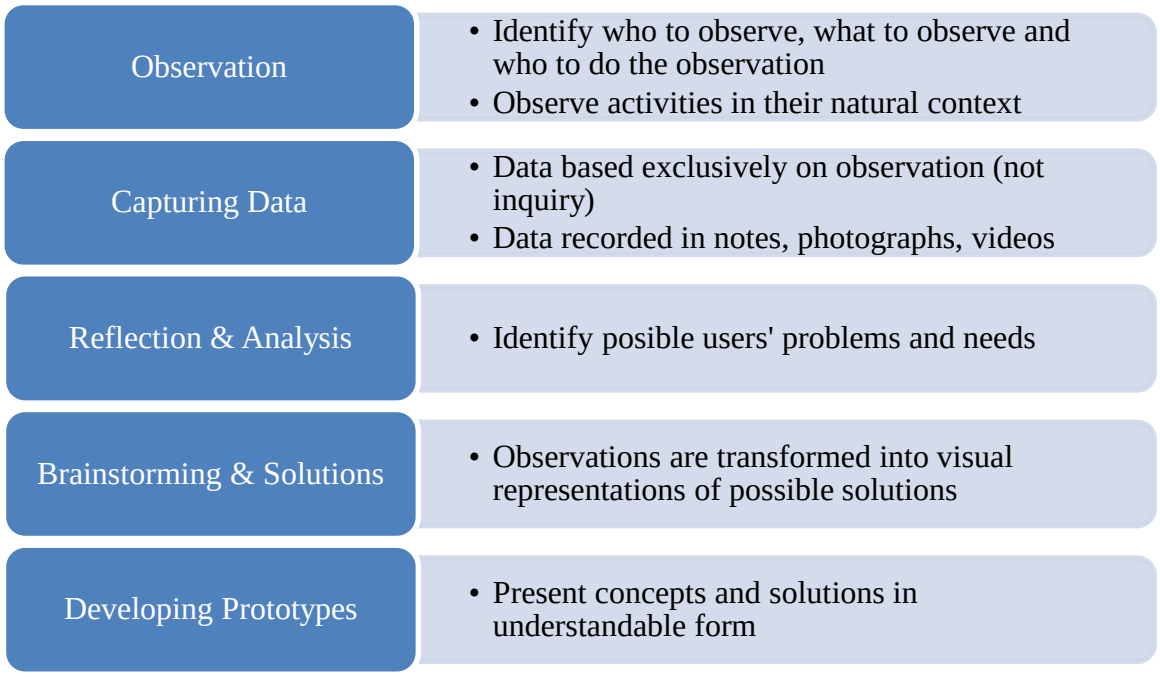
- **User customization** or how the customers re-design or tweak the product to better serve their own needs. Sometimes users combine several existing products to solve a specific need.

- **Intangible attributes of a product** or attributes that cannot be easily identified in a survey, but can bear a strong emotional charge. Example – environmentally friendly disks that clean clothes without detergent cannot gain a wide adoption due to the lack of "clean-clothes smell".

- **Unarticulated user needs** or problems that users meet in using products or services that they don't recognize as problems or don't articulate as they don't think those can be addressed. The combination of a designer who is also a user of a product can be very powerful as the same person combines unarticulated user needs with the knowledge how to fulfill those needs.

The process of Empathic Design consists of five steps described in Figure 21: Empathic Design process.

Figure 21: Empathic Design process



The challenge in the **Observation** phase is to identify who should be observed, who the observer should be and what should be observed. **Who should be observed?** It is not necessary that the person being observed is the customer. It may be that the person who should be observed is a customer of the customer, or a non-customer or even a group of people who collectively fulfill a task. **Who should do the observation?** There are multiple factors that can influence the information that an observer extracts from a situation – education, training, natural inclination. Thus

different observers observing the same situation may extract very different information from it. As people tend to focus on different aspects in a situation, the recommendation is to send out a small team of multi-functional specialists, where at least one member has experience in behavioral observation and at least one member should possess a deep understanding of the client company's organizational capabilities. The skills that are crucial to team members of an observation team are: open-mindedness, curiosity and observational skills. **What behaviour should be observed?** The key requirement is to observe activities in their natural form and context, as opposed to the artificial setting of a user laboratory or focus group.

Observation doesn't only make sense in the physical world, but also in the cyber-space as more and more tasks and transactions are done online. An example of online observation is a software vendor tracking which products are downloaded and installed and how those are used. Observation in the cyber-space can be very efficient as many people can be "observed" simultaneously. Also virtual discussion boards offer an easy way to get insight onto customers' feedback about a product and information about how the product is used and critiqued. However the techniques of Empathic Design do not translate directly from the physical world to the virtual world. The ability to gather lots and lots of data quickly does not automatically mean a true "observation". The Empathic Design philosophy is that data represents behaviour and to characterize consumers' behaviour based on data, a lot of analysis and reflection is required that cannot be fully automated.

In the **Capturing Data** phase data is gathered through observation rather than answers to questions. Sometimes open questions may be used to help the observer interpret user's actions, but most of the times data is gathered through visual, audible and sensory cues. Photography and videotaping are often used tools by observation team members as they capture subtle body cues (e.g. facial expressions showing surprise or confusion while attempting an action, spatial configurations, even things that the observing team doesn't notice while onsite). The key points in emphasizing on observation rather than **inquiry** are**:** people are often bad reporters of their own behavior; people tend to give answers that they think are right or desired; people are less likely to recall feelings about the intangible characteristics of a product or service while not in the process of using it; people stop noticing inefficiencies or inadequacies in the products they use after some time of using them; questions can be biased by the inquirer inner assumptions; questioning interrupts the natural flow of the user activities.

In the **Reflection and Analysis** stage of the Empathic Design process, possible users' problems and needs are identified and analysed. This stage involves not only the observation team members, but also other colleagues, who make their opinions only based on the data in front of them. This data consists of observers' notes, photos, videos, etc.

The next stage in the process is the **Brainstorming and Solutions** stage**.** This is the process through which analysed observations are transformed into graphic, visual representations of possible solutions. Brainstorming is simultaneously a creative process and one that is guided by clear rules: defer judgement, build on ideas of others, hold one conversation at a time, stay focused on the topic and encourage wild ideas.

Finally, in the **Developing Prototypes** stage prototypes are used to illustrate how possible solutions should look, function and be used. Prototypes enable the concept to be presented to other people outside the team in an understandable form. They are very concrete and as such can foster discussion and stimulate reaction from possible customers. Another practical form for testing a concept are simulations which may have a different level of techie-ness.

At the end of the day Empathic Design is about ideas still being guided by users, just in a way that users don't really realize.

3.5 Lead User Method

The Lead User Method was originally defined by Eric von Hippel in 1986 ((Hippel, 1994). He questioned the basic assumption that product innovations are usually developed by product manufacturers and presented a series of studies where the sources of innovation vary greatly and in some fields most of the innovations are actually done by users or suppliers.

Von Hippel's definition of a lead user has two components. Firstly those are users who experience an unsatisfied need that will be relevant in the general market place as well, maybe just years later. The important component here is the relevance of the need to other users as well. This is not always true with user innovation as innovating users are usually only concerned with satisfying their own need, rather than the needs of other users. On the opposite, manufacturers are always concerned with having a large market for their products. Secondly, lead users are users who would benefit substantially from obtaining a solution to their needs. This is the motivation that drives them to invest efforts and funds into finding a solution themselves.

One of the great benefits of the Lead User Method for designing new products is its potential to minimize the risk of failure in the marketplace that is a major issue with new product development. This results from the core definition of Lead Users and more specifically the requirement for relevance in the marketplace.

The steps in a lead user project (with a special focus on applying the Lead User Method in SMEs) have been described in a wiki developed as part of the "Open Innovation Project" (http://openinnovationproject.co.uk/). The underlying assumption in describing the steps is the willingness to lower the required methodical skills to apply the method as well as to lower the costs associated with applying the method in order to make it affordable to SMEs.

Phase 1: Preparation of the Lead User project

This phase starts with the **Lead User Project Planning step.** This steps is done by the project planning team, which has to combine sufficient marketing and technical knowledge that different team members bring to the table. In this step the project planning team first has to prepare the project plan, which defines the focus of the lead user study. The focus includes product or service area of interest, overall project objectives and resource requirements. The project/service of interest are defined together with the general types of markets and product/service application. The overall project objectives are the specific deliverables and goals of the project (what are the questions we want to get answers to, what is the market impact we want the innovation to get us or in general – how we will know the project is successful).

What follows is the **Selection of the Lead User Research team step.** In this step a team of 3-4 members is selected to deliver the research. This is a small enough group to be able to quickly move forward as a team, but also not too small that different perspectives would be lost. Having also a team that is strongly diversified by expertise and perspective can contribute to the creative process. Expertise in a specific area should be preferred over expertise in the area of research (as the acquisition costs of general area expertise is greater than the acquisition costs of expertise in the specific domain). The team members should also be able to think creatively, be open minded and accepting of new approaches and different perspectives from theirs and team players who would facilitate the joint creative process.

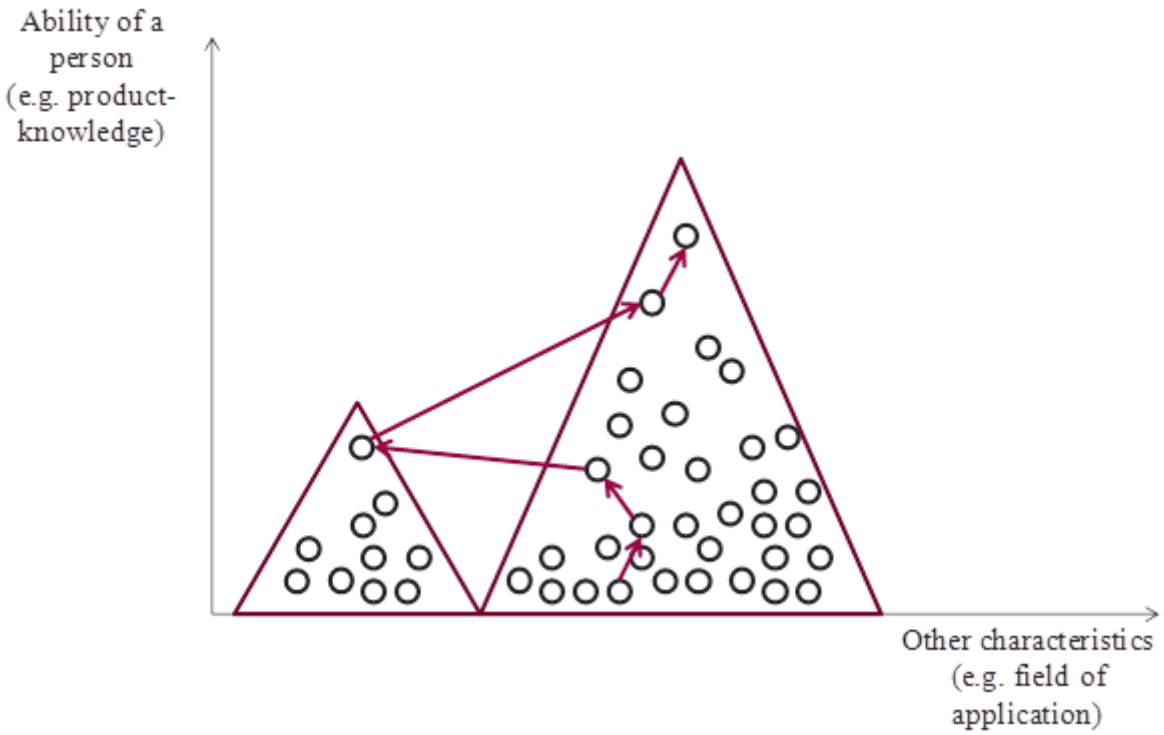
The last step of the initial preparation phase is the **Preparing the Lead User Research team step.** In this step the team gathers information, clarifies the expected outcomes, gets grounded in the project and creates a user research plan.

Phase 2: Identifying Trends and Key User Needs

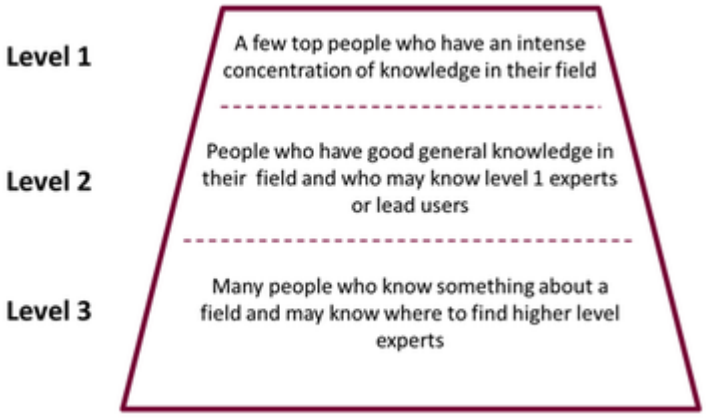
In this phase the team does a deep investigation of the market trends and focuses on emerging needs of users in the target markets.

The first step is the **Trend Identification step**. Trends can be identified through talking to market experts or following recent publications. While identifying trends the team tries to predict their likely impact on market needs for new products and services. Once those are defined, the team focuses on a specific user need area that they want to see addressed.

In order to identify emerging needs, the team has to interview relevant stakeholders with the purpose of **Identifying Lead Users**. In order to identify lead users, the team contacts experts and key users and uses specific techniques to get them help the team identify other users, who can potentially turn out to be lead users. One of the widely used approaches is the **pyramiding approach**. It is based on the principle of social recommendation. The basic assumption in the pyramiding approach is that people who have a strong interest into a certain area of knowledge are likely to know other people who know more than them about the area and look up to them for information. In the pyramiding approach a search for a user who has a certain knowledge or expertise starts with the virtual social network on the Lead User Research Team themselves. In the next step, every stakeholder who is being interviewed by the Lead User Research Team is asked for a recommendation of another stakeholder who would know more about the subject. This is a technique to expand the virtual social network. This process continues until the trend of new recommendations starts to go down. Once the social network is traversed, the interviewed users are clustered to identify which of them qualify to be Lead Users. (Figure from http://leaduserhandbook.eu/index.php?title=Leaduserhandbook.eu/Exploring_Major_Trends_and_Emerging_Needs/Identification_of_Lead_Users)



The pyramiding approach is based on a structured way of looking at how expertise in a certain field is spread across the population. (Figure adapted from Churcill and E. von Hippel, 2009, "Lead User Project Handbook – A Practical Guide for Lead User Project Teams", Cambridge, MIT Press)



This approach is preferred to other, more standard, "scanning" approaches that are based on the principle of collecting information from each interviewee in order to determine the individuals who have the most advanced knowledge, for its much greater efficiency.

Once Lead Users have been identified, those are interviewed. **Interviewing Lead Users** is a key step in the phase of identifying key user needs. Through Lead User interviews, the Lead User

Research Team can gain insights into emerging needs (as Lead Users experience those), design hints and new product direction (as Lead Users usually are motivated to be at the "bleeding edge" of technology in order to be able to address their unsatisfied needs themselves) and of course ideas on how to locate further experts and lead users (as they tend to have rich personal networks with experts that can help them meet their needs).

After Lead Users have been interviewed, the key step of **Framing an Important User/Customer Trend** is executed. This step involves assessment, creative interpretation and a careful combination of the market trend information gathered in previous steps. The goal is to select and clearly define a specific user need that will be the focus of concept generation. This includes a selection of target user group, a clear need statement that captures the essence of the need, and the key attributes of the need that the new product/service will address. In the process of the need framing, it is important that first each team member presents their perspective first, then a group discussion takes place where the team members try to combine the information presented, possible needs framings are created. A sanity check is then done whether the identified trend and need meet the success criteria of the project – e.g. does a solution of those have the potential to bring market results to the project goals in the first phase of the project and more specifically in the step of setting project objectives as part of the Lead User Project Planning step.

Once the possible need framings have been assessed in the context of the project success criteria, one or two promising need framings are selected.

This phase of the flow of applying the Lead User Method is not linear. The listed activities are done in iterations and repeated multiple times.

The next and final phase of the Lead User Design Method is the **Developing New Product Concepts Phase.** The goal of this phase is to use the needs framing done in the previous phase as well as the identified possible solutions and come up with a strong product or service concept. A core activity in this phase is the Lead User Workshop – a several days long event, in which lead users, experts and the project team come together to do intensive design and problem solving together. The result of the design workshop is a comprehensive product or service proposal which can be then put for review and approval. The focus of the workshop can be to complete preliminary solution concepts identified in the previous phase, or to solve a specific design or implementation problem in the context of identified product or service concept. It is important to set a clear and specific goal for the workshop, make sure that it can be realistically achieved in the time period (usually 2-3 days) and make sure it matches the attendees skills and expertise. The participants are usually not more than 15-18 people, including the lead users, experts and research

team members. Once the workshop is conducted and a written statement of the improved product/service concept is prepared, this phase is considered complete.

The final phase of a Lead User Project is the **Completion Phase**. During this phase the developed product/service concept is presented to management, its relevance is tested (e.g. by running it by a small group of users in the target user group) and further adjustments are made (as the user need is emerging and will become really pressing only in a future moment in time, users may not be able to grasp the benefits of the proposed solution). The process of presenting and approving product/service proposals is very specific to the specific company environment, hence is outside of the focus of the Lead User Method itself.

3.6 User Centered Design Methods "At a Glance" Comparison

The above commentary described several design methods that fall into the broader category of UCD. Those methods share multiple characteristics that classify them in the common UCD category, but also have differences. Looking back at the *Design Approaches for Innovative Software Products and Services* section, there are "top ten" characteristics we chose to describe the general design approaches: involvement of users, time of involvement of users, designer role, level of complexity, process, UX coverage, level of formalism, innovation potential, data orientation, risk of "getting it wrong". The design methods we reviewed share similar "ratings" per those characteristics and exhibit the same "design profile"

- **Explicit involvement of users** – most of the described methods have heavy involvement of users into the entire design process. A common belief that all of them incorporate is that design should be done 'with' the user, not just 'for' the user. However there are different motivations for this intense involvement. In some cases (e.g. Participatory Design), the motivation is industrial democratization of the process of increased introduction of computerized systems in all areas of design and production, while in other cases (e.g. Lead User Method), the motivation is to tap into a pool of problems and solutions that some users experience that would be relevant (at a later time) to multiple users in the market place. Also some methods (e.g. Empathic Design) do not involve users in the stages of interpretation and analysis or solutions generation.
- **Time of involvement of users –** all of the described methods involve users very early, in the "fuzzy front-end" of the innovation process. Users actively participate in the solution problem definition process as well as in re-framing the problem and generation and validation of possible solutions.

- **Designer role** - in all of the reviewed methods the designer is more a "translator of input" rather than a "creator", whose efforts are mostly a "black box" to the outside world. This does not come to question the level of "creativity" in the process, but rather to emphasize on co-creation as the primary means to get to a solution, as opposed to "genius talent".
- **Level of complexity** – in all design methods described, the level of complexity that can be solved is low to medium. These methods (in their narrow sense) do not provide a comprehensive framework for solving a particularly complex workflow optimization problems, as opposed to more formalized methods like Activity Centered Design (Constantine, Human Activity Modeling: Toward A Pragmatic Integration of Activity Theory and Usage-Centered Design, 2009) and Systems Design (Saffer, 2006).
- **Process** – all of the described methods have certain front-end steps and then employ iterations to eventually work out a complete, viable and valid solution. *Table 3: UCD Process Phases in Different UCD Design Methods* below provides an attempt to map the different stages in the reviewed design methods to common, "conceptual" phases in the UCD process. The **user identification step** is very similar among Participatory Design, Contextual Design, Design Thinking and Empathic Design and can potentially involve any user who will directly benefit from the end solution. The main requirement towards users is that they are domain experts in what they do. They do not need to have the skills to elaborate their domain knowledge, in fact most of the methods assume that people are not good at elaborating (or sometimes even realizing) their needs. The process of identifying users in the Lead User Methods is more complex as the requirement for domain expertise is not sufficient. Lead Users should be ones who experience a need that is rather unique at the moment, but has the potential to be experienced by many other users in the future. They should have strong enough motivation to pursue and invest in a solution and should be open about sharing their solution. The **observation step** in most cases involves the application of ethnographic methods of user research i.e. observing users in their natural context of operation, with a different level of emphasis of observation vs inquiry. The outlier in this phase is the Lead User Method as no real observation is involved - users are interviewed instead. In Lead User Method this step is to a large extent merged with the interpretation step. The **interpretation step** is similar in concept among methods. Specific tools or artefacts may vary – e.g. in the Contextual Design, a set of models are created that represent certain aspects of the problem space (e.g. flow model, cultural model, sequence model, etc.) and a "storyboarding" is used to present the "whole picture".

Table 3: UCD Process Phases in Different UCD Design Methods

Conceptual Phases	Participatory Design	Contextual Design	Design Thinking	Empathic Design	Lead User Method
Problem Definition	Initial definition of the goal	Initial definition of the goal	Initial definition of the goal	Initial definition of the goal	Trend identification, "success criteria"
Users Identification	Identify domain experts, open to learning about new technology to support the process.	Identify domain experts	Identify domain experts	Identify domain experts	Lead User identification – "pyramiding approach" preferred over "scanning"
Observation	Visits to plants, trade shows and vendors. Discussions and mutual learning between designers and users.	Contextual Inquiry (a strictly defined form of ethnographic fieldwork)	Ethnographic fieldwork – asking the "What is?" question	Emphasis is on observation (not inquiry)	Users are interviewed (not observed)
Interpretation	No specific techniques in referenced literature	Work Models - Flow Model, Cultural Model, Sequence Model, Physical Design, Artifact Model Affinity Diagram		Analysis of the observation data	
(Re)Framing	N/A The goal is to support the existing work practice with new, computerized tools	Explicit - done during the Visioning step. Create storyboards to describe ideas. Story boards User environment design	Explicit -asking the "What if?" question, coming up with possible solutions through brainstorming and co-creation	Re-framing done mainly by designers	Explicit – need framing that is validated against the project success criteria
Solution/Idea Generation	Driven by existing tools as the "ideal" for computer-based tools			Brain-storming (design team only). Does not involve users!	Lead user workshop as a means to develop new product concepts
Prototyping	Low-fidelity/High-fidelity mockups Role plays	Low-fidelity paper prototypes	Low-fidelity/high-fidelity prototypes	High-fidelity prototypes preferred	
Validation	Based on prototypes	Based on prototypes	Based on prototypes	Based on prototypes	Based on prototypes

The **(re)framing step** is one of the most important steps in the design process. This step is different between the methods. Some methods (e.g. Participatory Design) do not involve re-framing as the basic assumption is that the ultimate achievement of the computerized tool would be to fully resemble the existing, physical tool. In Empathic Design and Contextual Design, the re-framing of the problem is done mainly by designers, after they have "stepped into users' shoes" during the observation and interpretation. The main goal is to satisfy the (un-realized or not-articulated) user needs. In Design Thinking and Lead User Method, the re-framing process involves users as well. The **solution generation step** is quite technical, after the solution vision/concept has been identified during the re-framing step and aims to transform a high-level idea into a working solution. In many of the methods, this step, as well as the following prototyping and validation steps, is executed in iterations. A portion of the solution is being modelled, prototyped and validated in each iteration. Iterations continue until a full solution is composed. The **prototyping step** also varies little between methods with main difference being the level of fidelity of the prototypes produced. The **validation step** is the final, standard step among methods, during which prototypes are "tested" with users to decide the usefulness and completeness of the solution.

- **UX coverage** – all of the discussed design cover rather fully the whole interaction of the user with the system in the problem domain.
- **Level of formalism** – is also very similar among methods. Most of them rely more on informal experimentation rather than on a very systematic and formalized approach.
- **Innovation potential** – given the re-framing step is a major step in most of the methods, they have the potential for coming up with radical innovations rather than just incremental improvements.
- **Data orientation** – all of the reviewed methods are qualitative
- **Risk of "getting it wrong"** – given the strong involvement of users in the process for all methods, the risk for the design approach to produce results which will not be successful on the marketplace is to a large extent mitigated.

4 Case Study of Applying Design Thinking at SAP

In order to relate the described theoretical framework to a practical application of some of the above discussed design methods in practice, we will discuss a case study of applying Design Thinking at SAP.

4.1 Case Study Selection Process

The selection of the specific context for the case study was driven by several considerations. One of the most important criteria was to explore a practical application of a UCD method, as this research focuses primarily on UCD, among other design approaches. Secondly, a large software development company was preferred, in order to be able to more clearly observe some aspects of the practical application of the design method – e.g. influence of the organizational structure and culture, management of the innovation process, participation of different functional roles. All of those are more difficult to observe in a smaller (e.g. start-up) environment as in such environment there is usually a rather flat structure, every employee tends to "wear different hats" i.e. combine different functional roles and processes are more informal, unstructured and non-repetitive.

The interviewee is a current employee at SAP, with the position of a Senior Product Manager and an active Design Thinking Coach, who has been involved in more than 20 projects using Design Thinking over a period of 5 years. The interview was done live around SAPPHIRE NOW, one of the biggest conferences organized by SAP where the company is hosting current and potential customers and partners to present and discuss with them solutions, visions and industry challenges. I used a pre-defined set of questions to guide the conversation. I however encouraged using free form of story-telling and reflections, in order to take the most of the experiences and take-aways made during the specific project that the case study focuses on.

4.2 Historical Background

The practice of using Design Thinking is not new for SAP. It all starts with Hasso Plattner – one of the co-founders of SAP, who has been especially active in supporting technology research initiatives. In 1998 Hasso Plattner founded HPI (Hasso Plattner Institute) for software systems engineering, based at the University of Potsdam, Germany as well as in Palo Alto, California and reportedly invested more than 200 mln EUR in financing HPI. In 2004, after reading an article about Design Thinking, Plattner was captivated with the idea of applying it in the software development process at SAP. In 2005, Hasso Plattner donated a substantial sum to founding the Hasso Plattner Institue of Design, (a.k.a. "D-School" or "d.school"), as a joint venture with the

Stanford University. He collaborated with David Kelley, founder of renowned innovation consultancy firm IDEO.

In 2008 Plattner brought 35 "design thinkers' into SAP to collaborate with the Corporate Strategy Group and make Design Thinking a strategic priority at SAP in order to drive innovation across the organization. Once the top level management of the company grasped the power of Design Thinking, the design method was infused in many different teams.

In 2012 SAP moved to scale Design Thinking as a way to help customers drive their business outcomes by re-framing their problems, while putting an emphasis on user experience.

In 2014 SAP appointed Sam Yen as a Chief Design Officer to further drive the SAP vision for design of products and services that stresses on the importance of user experience. Currently multiple groups at SAP adopt Design Thinking. There are two main purposes for which Design Thinking is used – product innovation (mostly applied within R&D) and business model innovation (mostly applied in the field when working with customers and partners to apply SAP solutions for solving business challenges).

4.3 Organizing for Applying Design Thinking

The organization of applying Design Thinking is rather de-centralized. There is a pool of Design Thinking coaches, who are trained and have different levels of experience facilitating projects, where Design Thinking is applied as a framework for fostering innovative thinking and focus on UX. The list of coaches can be accessed through an internal portal and a request can be submitted for a coach to take part in a project. Coaches themselves have to coordinate their agenda with the requirements of the project and if there is a fit, they get assigned for a period, ranging from several hours (advice giving) up to 12 weeks (working on an innovation project as part of a cross-functional team). The responsibility of a coach is mainly to facilitate the process, coordinate the work in the cross-functional team according to the Design Thinking principles, suggest tools and techniques to be used in different phases and steer the process towards a successful and timely completion. In certain cases coaches can also train project participants in Design Thinking or support the process of identifying people who are a good fit to join the project in a certain role.

The project in the focus of this case study is one where SAP, together with a partner company that provides solutions utilizing SAP infrastructure and services, had to tackle the challenge of optimizing the PoS (Point of Sale) process at retail stores.

4.4 The Design Thinking Team

The team was gathered following a selection process, during which a "T-shaped diagram of skills" method was used to assess the abilities of potential team members (Tim Brown, 2010). T-shaped diagrams are used to visualize the skills of a person. The vertical stroke of the T is the depth of skills that allows a person to contribute to the creative process. The horizontal stroke of the T is the disposition for collaboration across different disciplines. Empathy is the skill that strongly contributes to the horizontal stroke as it enables a person to imagine a problem from another perspective.

Using this technique, a team of 8 people, spanning SAP and the partner company, was formed. The team included people with different roles and areas of expertise, who could effectively collaborate with each other. The roles in the teams were: Design Thinking coach (the interviewee for this case study), software developer (one from SAP and one from the partner company), industry expert in retail processes, business development expert (acquainted with industry regulations and competitive solutions), solution manager (an expert from SAP who brought to the table rich experience integrating SAP products and services with partners solutions and devices) and UI designer (one from SAP and one from the partner company).

4.5 The Problem

In the stage of defining the problem, the problem was defined in a narrow way – "Develop a mobile app that connects to SAP cloud services to help the work around PoS in a retail store where some customers signalled about challenges following recent IT trending moves in the retail industry". On later stage the problem become more specific - the app was supposed to help the Purchasing Manager in understanding what inventory types and in what quantity get sold (i.e. their availability at the warehouse decreases) and in predicting future needs, so that the Purchasing Manager can order supplementary quantities. During the observation phase some bottlenecks were identified with regards to the job of the Purchasing Manager.

Some specific requirements had to be taken into consideration – e.g. the Purchasing Managers had to be very active and mobile, they were constantly in motion, walking around the warehouse and checking staff, hence they had to be able to handle the device with just one hand; compliance with SOX security requirements and industry regulations had to be followed.

4.6 The Process

The process flow had several distinct stages, where the Design Coach took care to help the participants move between stages, without necessarily calling out which stage the project is in.

The Problem Definition stage

As previously discussed in ***Error! Reference source not found.*** section, the project started with narrow definition of what seemed to be the goal – "Develop a mobile app that connects to SAP cloud services to help the work of the Purchasing Manager in a retail store". Some constraints and special requirements were identified – e.g. the ability to handle the device with one hand, complying with industry regulations and ensuring compliance with SOX security requirements, etc. The project was also constrained to take not more than 10 weeks, with the intent to present a prototype to decision makers (at SAP and the partner) at the end of the time period. The goal of the project was to identify a solution and to provide quantitative and qualitative motivation for customers to purchase such solution and for SAP and the partner to jointly invest in the development of such solution. During this stage main personas were sketched. Also some success criteria were drafted.

The Observation stage

In this stage the team recruited 4 different end users, employees in different retail chains of various size and operations, and conducted ethnographical studies with them. The team split into 3 groups – one group observed several users who worked for different retail chains, one group observed just one user and one group did not observe any users (intentionally were kept with clear and unbiased minds). The latter group examined the activities of the business network around the observed retail stores. This split was done to ensure different perspectives are brought into the post-observation discussion. The end users to be observed were identified based on the overlap that their functional roles had with the personas, rather than based on titles. In some cases it happened that there was no complete match with the persona – either one user had more responsibilities than the persona or several people combined covered the responsibilities described for a single persona. End users were observed in their natural environment – while doing their day-to-day jobs at the warehouse. Every detail of their operations was captured and analyzed – what time they spent where, what interruptions occurred, what trajectories they had, how much time it took them to complete a task.

After the observation sessions were completed, the team got together to discuss the observations. This was done by creating observation notes, putting them together and clustering them based on a common 'theme' or 'topic'. Some examples of 'themes' that emerged were mobility, predictability, etc.

The Ideation stage

One step that the Design Thinking team executed during the Observation stage turned out specifically useful for the purpose of re-framing the problem. Besides observing end users, the team had also talked to decision makers – managers of operations or supervisors, who provided more context on the objectives for end users, as well as success criteria and KPI (Key Performance Indicators) that were used to assess their performance at work. Those success criteria, as well as the insights about the success criteria for the critical function in focus – PoS management were used to re-frame the goal that was originally perceived to be with the vague scope "address challenges in PoS process", into the more clear goal to "Optimize the PoS process related to the activities of the Purchasing Manager". This re-framing enabled the team to come up with ideas for changing the business process itself, where the functionalities implemented in the application were just a fraction of the overall solution.

Once the problem was re-framed and the team had gathered and clustered the observation notes, the whole team engaged in a discussion of their perceptions of personas and tasks. Everyone shared their take-aways from the ethnographic study.

After discussing all viewpoints, the next step was to do a brainstorming for solutions. Those solutions were at the level of a specific 'theme' or 'topic' that came out of observation notes clustering. Only the part of the PoS process that had to be optimized was a subject of the brainstorming. Out-of-the-box and even "crazy" ideas were welcome. The Design Thinking coach specifically encouraged team members to build on top of other people's ideas, rather than discarding them in order to put their own ideas forward. Some of the 'themes' brought up more than one proposal for a solution. The Design Thinking coach recalls that solutions tended to fall into three broad categories – (i) solutions that can be built with existing products/modules/features, (ii) solutions that can be built by combining existing product/modules from SAP and the partner and (iii) brand new concepts. In some cases a certain 'theme' could not be fully addressed and then the success criteria came into play, modifying the goal to improving the existing process and tools as much as possible according to those success criteria.

After solutions were defined for each emerging 'theme', the team got to one of the most challenging tasks – building up the "puzzle" of a complete end-to-end solution out of the solutions of the smaller 'themes'. This process required selecting one solution among possibly multiple solutions and combining them together so that a complete working solution is built. The role of the Design Thinking coach was to help the team make the decisions by keeping the success criteria in mind and in a way that mitigates the risk of falling into the ownership trap. One technique that

helped in this process was rotating team members to work on other people's suggestions thus removing silos and biases.

Once a concept for a complete solution was in place, the team moved on to creating quick and dirty prototypes and mockups that would help present the concept in a tangible and impactful way. Two main alternatives of the PoS process optimization were developed with a total of 12 mockups describing different variations of some of the core functionalities of the system.

When the prototypes were in place, the solution proposals had to be presented to the users and the decision makers. This was another tricky part of the project because of the risk of getting negative feedback or challenges, such that team members could easily fall into the trap of getting in a defensive mode. So the Design Thinking coach took on the challenge of presenting and discussing the outcomes of the process with the relevant stakeholders while the team was supporting and gathering the feedback. During this phase the prototypes were unified in one that was best accepted. On later stages (testing) the prototype was further tested and improved with a couple of additional iterations of feedback gathering and rework, so that at the end the outcome was a prototype of a solution that (i) the users started to ask to use in their work, (ii) it was feasible to produce and (iii) there was good business justification to build.

4.7 In Hindsight

The final part of the interview with the Design Thinking coach was focused on their assessment of whether the application of Design Thinking has improved the process and outcome of the product development in the company as well as the biggest challenges, risks and benefits out of it. The central take-aways were the following:

The biggest benefit of applying the Design Thinking method to the process of product development was the much more intensive involvement of users, which led to deeper understanding of users' real (vs perceived) needs, increased usefulness of the end solution and broader and quicker adoption. It also decreased waste as ineffective solutions were eliminated at an earlier stage and ideas were validated in a timely manner.

One of the key factors for successfully applying the technique was the clear vision that a SAP key executive had and the top-down approach which they applied. There was a clear direction for more intensive user involvement and a strong emphasis on user experience. Another key factor was the way this application was organized, having high level of diffusion within the organization, by building a flexible mechanism for different groups to be able to tap into a pool of educated coaches and apply the method with minimal overhead. According to the interviewee,

the fact that the process was light-weight and the focus was on project outcomes rather than on the Design Thinking process and education, helped the wide adoption of the method and the seamless integration into the existing processes and structure.

The risks associated to applying the projects, per the interviewee's assessment, were mainly related to the people factor and the level of understanding of different people about the goals and benefits of the process. In their experience, applying this method was not applicable to all problems and situations. Sometimes this became clear after they were involved as a Design Coach. In some occasions expectations to the outcomes of applying the method were unrealistic and had to be managed.

Based on their broad experience, being a design coach in more than 20 projects, the overall assessment of the interviewee was very positive about the benefits that this method brought to projects it was applied to.

5 Summary and Learnings

We started this research by defining its purpose to be a review, analysis and comparison of design approaches and design methods available for innovative software products and services. This discussion is applicable mainly to the "fuzzy front-end of innovation", where the most of the miracle of product inception and design happens. The usefulness of this analysis (in the context of existing literature) comes from putting the various design approaches and design methods together, in one "big picture", and comparing them. This analysis should make it easier to understand in what context each of them is applicable. It is not a means to do a very deep dive into any individual approach or method, although it focuses more on User Centered Design rather than other approaches and gives more details, including the outcomes of a live interview, about Design Thinking, among other UCD methods.

In the initial chapters of this research, we discussed the general challenges of the creative process, by introducing the notion of "abduction" as the basic reasoning pattern of productive thinking. We talked about "closed" design problem solving, represented by the logic equation of abduction 1 and the "open", "conceptual" design problem solving, represented by the logic equation of abduction 2 (see Chapter 1.4 General Discussion of Design Challenges). We introduced the notion of a "frame", "framing" and "re-framing" as a mechanism for creation of a (novel) standpoint from which a problematic situation can be tackled. We discussed the concept of framing as a means to address the core design paradox, not by facing it "head-on", but by looking at the

broader problem situation and searching for surrounding themes that can foster the creation of new frames.

We defined three main research questions to guide our analysis.

The first questions was “Which are the ways to approach innovative software products/services development and what are the specific methods that can be applied?” We tackled this questions in the first five sections of Chapter 2: *Design Approaches for Innovative Software Products and Services,* where we presented the five main design approaches: User Centered Design, Activity Centered Design, Data Driven Design, Systems Design and Genius Design. We provided a brief discussion of those as well as a conceptual process for applying them.

The second questions was “How should one decide which approach and methods to use? What are the criteria that one should take into consideration when choosing an approach and a specific method?” We tackled this question throughout the most part of the rest of the analysis. In the last section of Chapter 2: *Comparison of Design Approaches* we suggested a framework for analyzing and comparing the different design approaches visually. We proposed a set of criteria (“top ten” characteristics) of design approaches, as well a graphical representation (“profile”, “visual representation”) of those design approaches. The set of criteria were: involvement of users, time of involvement of users, designer role, level of complexity, process, UX coverage, level of formalism, innovation potential, data orientation, risk of "getting it wrong". Each design method was assigned a “rating” against each criteria, which was a numerical value between 1 and 10, representing the place the design approach stands at in the continuum of values that the criteria could have. The “profiles” of design approaches allowed for a nice visual comparison of design approaches and illustrated in an easy-to-understand way how some approaches were more limited than others or how two approaches could be combined to fully cover the full spectrum of the defined criteria. This framework for comparing design approaches was the first building stone towards helping the reader make a decision which direction to go when tackling a particular design problem within a given context.

The main learnings out of this framework could be summarized as follows:

- UCD as an approach is very user-focused i.e. involves users explicitly, throughout the full development cycle. The role of the designer is to translate the realized and un-realized user needs into tangible requirements. The process is qualitative, iterative and the approach covers the end to-end UX. The approach uses informal, experimentation-based techniques; can handle a high level of complexity and exhibits a high degree of innovation potential.

- ACD is an extension to UCD, mainly in the sense of bringing more attention and formalization around describing and modelling user activities. It is mostly a superset of UCD characteristics, however it does not focus that much on the user per se, but on the activities that the user is interested in.
- DDD is very different from UCD and ACD mainly in terms of being completely quantitative and having almost no direct involvement of users. It also has much lower level of complexity and innovation potential and much more limited coverage of the UX. Compared to UCD, DDD only covers a fraction of the UCD "profile".
- Systems Design is again a "beast" of a very different kind. Similarly to DDD, it does not involve users directly. It puts the designer into a "creator" role. Its main purpose is to handle very complex scenarios and thus is very formal, abstract and systematic in nature. It does have a big innovation potential.
- Genuis Design is entirely focused on the designer as a creator, enabling them to work quickly towards a very clear vision and to come up with radical solutions, with high innovation potential. However, due to the very limited involvement of users in the process, it also bears a very high risks of "getting it wrong". Genius Design and UCD are almost "opposite" as approaches, each bringing their own pros and cons.

Based on those learning, we then continued to focus our research in one of the possible five directions in Chapter 3: *Drill-down into User Centered Design Methods* – namely the User Centered Design approach, which exposed interesting characteristics for driving innovation into the product/service design and development lifecycle. This was the next building block for answering the question "How should one decide which approach and methods to use?" We introduced several specific design methods that are User Centered in terms of approach – Participatory Design, Contextual Design, Design Thinking, Empathic Design and Lead User Method. For each of those methods, we presented the basic assumptions as well as the process for applying them in practice. In the last section of Chapter 3: *User Centered Design Methods "At a Glance" Comparison*, we put them in all common conceptual process description context in order to highlight specifics and differences.

Then, we moved to answering the third research question: "Should those methods be applied in their purist form or are combinations and/or modifications of methods appropriate? What is the general practice in the industry?" In an attempt to put the theoretical framework into practical application, and to combine the multiple case studies published online with personally experienced take-aways, we looked at the application of Design Thinking at SAP in Chapter 4: *Case*

Study of Applying Design Thinking at SAP. This part of the research was based on a set of live interviews with a Design Coach for Design Thinking at SAP, who has extensive experience directly participating in multiple real-life projects. Though the case study was based on a particular project, the overall take aways about the application of Design Thinking were based on the full experience out of those many other cases. The choice of company was based on the size of the company (with all challenges implied related to organization structure, propagation and dispersion of the design methods) and the specific design method, per our focus on UCD methods. The main learnings, related to applying the specific method, coming out of the case study, were in several distinct directions:

- General outcomes: The process greatly increased the level of user involvement in the design of products, services and specialized solutions. This led to deeper understanding of user needs, early validation of possible solutions, increased user engagement, earlier and wider adoption, de-creased waste.
- Process-wise: The process generally follows the prescribed steps. However at each step there are specific techniques that are introduced to facilitate achieving the desired outcomes of each step (e.g. "T-shaped diagrams")
- Organizational: A key prerequisite for the successful application was the top-down approach, where a key executive was sponsoring and leading the adoption of the method throughout the organization. Another key factor was the fact that the process was de-centralized and light-weight, which made it easy to adopt. The Design Thinking coaches available throughout the organization, who facilitated projects and helped educate others, had a substantial impact.
- Culture/mindset-related: The people factor was both in favor of and preventing successful application in different cases. Whenever the participants had realistic expectations about the outcomes as well good general skills for working within a Design Thinking team (e.g. open-mindedness, empathy, cross-functional collaboration skills), the outcomes met the goals. However, there were cases of unrealistic expectations or personal traits that prevented some people from successfully using the method to drive innovation and useful solutions.

With that, we gave a practical example of design methods application that further hardened some of the outcomes and take-aways from previous sections.

This research is open-ended and can be used as a starting point to a further analysis of design approaches and methods or as a reference in analyzing other aspects of innovation in software.

6 Bibliography

Ambrose, G., & Harris, P. (2009). *Basics Design: Design Thinking: 8.* Ava Publishing.

Bohn, D., & Hamburger, E. (2013, 01 24). Redesigning Google: how Larry Page engineered a beautiful revolution. *The Verge*.

Constantine, L. L. (2009). *Human Activity Modeling: Toward A Pragmatic Integration of Activity Theory and Usage-Centered Design.* Springer London.

Constantine, L. L., & Lockwood, L. A. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design.* Addison-Wesley Professional.

Divine, J. (2009, 03 20). *Jamie Divine's person blog.* Retrieved from http://stopdesign.com/archive/2009/03/20/goodbye-google.html

Dorst, K. (2011, November). The core of 'design thinking' and its application. *ScienceDirect, 32*(6), 521-532. Retrieved from http://www.sciencedirect.com/science/article/pii/S0142694X11000603

Dubberly, H. (2006, 07 26). *What is Systems Design?* Retrieved from Dubberly Design Office: http://www.dubberly.com/articles/what-is-systems-design.html

Hippel, E. (1994). *The Sources of Innovation* . Oxford University Press.

Holtzblatt, K., & Beyer, H. (2014). *Contextual Design.* Morgan & Claypool Publishers.

ISO9241-11. (1998, 03 19). Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. ISO 9241-11.

ISO9242-210. (2010). Ergonomics of human-system interaction - Part 21): Human-centered design for interactive systems.

Kohavi, R. a. (2004). *Front Line Internet Analytics.* Retrieved from http://ai.stanford.edu/~ronnyk/emetricsAmazon.pdf

Kohavi, R. C. (2009). *Online Experimentation at Microsoft.* Retrieved from http://www.exp-platform.com/documents/exp_dmcasestudies.pdf

Kohavi, R., & Longbotham, R. (2015). Online Controlled Experiments and A/B Testing.

Krenchel, M., & Madsbjerg, C. (2014, 11 04). Your Big Data is Worthless if You Don't Bring It Into the Real World. *Wired Magazine*. Retrieved from http://www.wired.com/2014/04/your-big-data-is-worthless-if-you-dont-bring-it-into-the-real-world/

Leonard, D., & Rayport, J. F. (1997). Spark Innovation Through Empathic Design. *Harvard Business Review*, 102-113.

Liedtka, J., King, A., & Bennett, K. B. (2013). *Solving problems with Design Thinking: 10 stories of what works.* Columbia University Press.

McClain, A. (2015, 02). A Human-Centered Approach to Data-Driven Design. (R. Magoulas, Interviewer)

Mcdonald, S. (2006). Modified contextual design as a field evaluation method. *Proceedings of the Fourth Nordic Conference on Human-Computer Interaction*, (pp. 437-440).

Muller, M. J. (2002). Participatory Design: The third space in HCI. In M. J. Muller, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications* (pp. 1051-68).

Norman, D. A. (2005). *HCD harmful? A Clarification.* Retrieved from Don Norman: Designing For People: http://www.jnd.org/dn.mss/hcd_harmful_a_clari.html

Norman, D. A. (2005). *Human-Centered Design Considered Harmful.* Retrieved from Don Norman: Designing For People: http://jnd.org/dn.mss/human-centered_design_considered_harmful.html

Notess, M. (2005). Using Contextual Design for Digital Library Field Studies. *JCDL05 Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries.*

Pavliscak, P. (2015, February 16). Six Myths about Data-Driven Design. *UX Magazine*. Retrieved from http://uxmag.com/articles/six-myths-about-data-driven-design

Pink, D. H. (2006). *A Whole New Mind: Why Right-Brainers Will Rule the Future.* Riverhead Books.

Rowe, P. (1987). *Design thinking.* Cambridge MA: MIT Press.

Rubin, J., & Chisnell, D. (2008). Handbook of usability testing. Wiley Publishing, Inc.

Saffer, D. (2006). *Desiging for Interaction: Creating Smart Applications and Clever Devices.* New Riders.

Steen, M., Kuijt-Evers, L., & Klok, J. (2007). Early user involvement in research and design projects - A review of methods and practices. *Dancing with users: how to organize innovation owith consumers and users.* Vienna: 23rd EGOS (European Group fro Organizational Studies) Colloquium.

Tang, D., Agarwal, A., Meyer, M., & Deirdre, O. (2010). *Overlapping Experiment Infrastructure: More, Better, Faster.* Retrieved from http://static.googleusercontent.com/media/research.google.com/en//archive/papers/Overlapping_Experiment_Infrastructure_More_Be.pdf

Treverton, G. F. (2007, 07). Risks and Riddles. *Smithsonian Magazine*.

Winograd, T. (1996). *Bringing Design to Software.* ACM Press.