Spike

Exploration

Evaluation

Challenge

Master Thesis

Responder's Inquiry - Spikes

Rikke H. Jensen

Kenneth H. Brodersen

Supervisor

Ivan Aaen

Cassiopeia - Department of Computer Science
Selma Lagerlöfs Vej 300, 9220 Aalborg Øst

Title:
PhysioSphere

Area:
Software Innovation

Field of Study:
Master of Science in Software Engineering

Project Period:
Februar 1st, 2013 - June 17th, 2013

Project Group:
sw1012f13

Participants:
Kenneth H. Brodersen
Rikke H. Jensen

Supervisor(s):
Ivan Aaen

Number of Printed copies:
4

Number of Pages:
83

Date of compilation:
June 17, 2013

Abstract

Our software industry are increasingly coming under pressure from countries with lower production cost. To remain competitive we must shift focus from just producing software to developing software that offer a high degree of value towards user and society. With other words, we must be innovative. From a developer's point of view such innovation can arise through the experimentation and exploration into the technology in search for new and unforeseen value propositions. Our 9^{th} semester project showed that such exploration can be hard to manage and structure.

This master thesis project present our effort into making innovation a part of a developer's everyday activities. We see this in the light of Essence, a development methodology supporting the innovative software team. We introduce an activity that will help structure a developer's exploration into the affordances the technology provides. We call this activity a Spike. To help frame our proposal we base our efforts on theory from the philosophic view of pragmatism, and the idea that we must experiment and explore a situated problem to understand its meaning.

We experiment with the Spike on our produced solution from our 9^{th} semester project. PhysioSphere is a system assisting in the rehabilitation of patients. It accomplish this by utilising the Kinect sensors to verify that exercises are performed correctly and on time. Through the application of the Spike we hope to identify new and interesting ideas.

Preface

This report documents the master thesis made by group SW1012f13. The master thesis is made as the 10 $^{\text{th}}$ semester project, developed under the topic: Software innovation. Our research has been conducted under the Systems Development (SD) research group, part of the Information System (IS) research unit at Department of Computer Science at Aalborg University.

The reader of this report is expected to have knowledge regarding software innovation, software engineering and processes. Some basic understanding of software design, Human-Computer-Interaction (HCI) and basic programming concepts may become useful.

We would like to thank our supervisor Ivan Aaen, Associate Professor at Aalborg University, for providing ongoing feedback throughout the semester and for showing a great interest in our project. We appreciate his commitment and contributions – with regards to the development of our project, sharing knowledge with the use of tools and techniques related to Essence, and software innovation in general.

We would also like to thank Thomas Pedersen for designing our front cover for this report.

This report consists of the following four parts:

1. *Introduction:* This part introduces the concept of software innovation, gives an introduction to the development methodology Essence and recaptures our effort and learned lessons from our 9$^{\text{th}}$ semester project.
2. *Our Contribution:* This part presents the theory of pragmatism, relates this to an innovation context and present our theoretical contribution – the Spike activity.
3. *Experimenting:* This part documents our own experimentation with the Spike activity, in form of two Spike experiments.
4. *Learning:* This part concludes the project where we present our reflection, conclusion and future works.

Chapter 1

Introduction

Software solutions are progressively being applied to solve problems and challenges we face in the society today. Once software succeeds – when it meets the needs of the people who use it– it changes the way we experience the world. Currently, much research is conducted into understanding such needs and how to produce software in response.

In the quest for efficiency, creativity and exploration into what the technology affords is spared little attention. We find this surprising considering our belief that such effort must be made to face the challenges experienced by our industry. In a globalised world we struggle to compete with countries who produce standard solutions at a much lower cost. To become competitive in this environment we must produce better solutions by not only meeting user requirements but exceeding them. To do this we must explore the technology and user needs to uncover unforeseen potential that can surprise both customer and user. Going from simply mass-producing software in response to requirements, to reflect and explore in search for value, is the essence of software innovation.

Such a shift poses new strains on us as software developers. From just considering how to develop a solution in response to requirements, we must now reflect on what this solution affords that the user might be unaware of and why this matter. This places us in a new role surrounded by uncertainties and possibilities. It is our responsibility as innovative software developers to find a balance between managing these uncertainties and exploring the possibilities.

The problem we face – as practitioners – is that not much research is being conducted into supporting such an effort. Most people seem to be for innovation but, in our view, it has turned into a whim of fashion discussed at an organisational level. As innovation, in its simplest form, is the successful exploration of new and novel ideas, we believe that most of these ideas come from the hands of engineers and practitioners.

This master thesis is concerned with the research into and development of an activity that supports these practitioners in their day-to-day activities. This activity is intended as a contribution to a larger development methodology – Essence – designed to support the innovative software team.

The design of this activity has been a research journey into a reflective reciprocal relation between conducting practice and developing theories to support and understand the problems we meet when doing this. The practical aspect involve the development of an innovative system supporting the rehabilitation of patients. Our journey into research and theory has brought us past innovation topics, creativity methods and tools, software engineering principles and the philosophical view of pragmatism.

Contents

Part I

Introduction

Chapter 2

Software Innovation

It is well understood that our western societies face some unique challenges in the upcoming decades. Two of these are the outsourcing of production to countries with lower labour costs [38] and the increasing pressure on our welfare systems [28]. The difference in labour costs has been here for decades, but trade reforms, decreasing transportation costs, the development of cheap and reliable communication tools together with an continuously increasing educational level in eastern countries, mean that outsourcing is becoming profitable even in high-tech industries. An example of such is seen here in Northern Jutland where the industry which develops and produces mobile phones has slowly been faced out the last decade [44].

To remain competitive in this globalised world our software industry must develop products and services which greatly differentiate them from what is being produced elsewhere. A way to promote such development is through a strong focus on producing solutions that offers high value towards users, customers and society. Creating such solutions is the essence of software innovation [2]

Software innovation is not a term that is well defined. To define its meaning within this project a short introduction to innovation in a more generic context must be given. The first Section provide this introduction while the second Section define the meaning of software innovation in the context of our project.

2.1 What is Innovation?

In its simplest form innovation can be defined as the *successful exploration of new ideas* [38]. Innovation is behind many of the dramatic moments in the history of industrial change. Examples of such include the invention of the spinning jenny transforming the textile industry during the 18 hundreds, while the telegraph and later the telephone enabled fast and reliable long-range communication.

It is here necessary to distinguish between inventions and innovations [24]. Invention is the creative act of producing novel ideas while innovation is the act of bringing these innovations into wider use [44]. The result of a successful innovation can be observed as change in peoples work or entertainment habits, the way business is carried out or other aspects of social change.

Invention and innovation is sometimes closely linked but often a time gap between these is present. This gap reflects the different requirements and conditions needed for working out ideas and for implementing and commercialising them [24]. Requirements for bringing them into wider use might include efficient production facilities and building materials with sufficient durability which at the time is not available.

As such innovation is rarely achieved in a single radical step overnight [44]. Instead most innovations result from an incremental process containing a number of steps. Each step might involve idea generation, exploration or smaller innovations, which slowly over time result in social change.

By introducing better services and products, improved business models and more efficient production processes, innovations not only enable companies to be more competitive. They also become a source of economic growth [38].

Developing new innovative products and services is also important in meeting the challenges of maintaining the quality of our welfare system which increasingly are coming under pressure [28]. This subject will be elaborated on throughout this report.

2.2 Innovation and Software Development

The IT industry is especially vulnerable towards outsourcing. A lack of physical products, combined with modern communication and collaborative tools, make it possible to distribute a development project among several locations. With fast and reliable internet connections these locations can be distributed across the globe. Considering this, moving parts or entire development projects to countries with only a fraction of the labour costs compared with western countries, then appear tempting. To remain competitive in such an environment our software industry must differentiate itself. It must deliver solutions that justify the extra cost. A way to achieve this could be to shift focus from simply complying with requirements to exploring and developing solutions that deliver a high degree of value towards customers and society. Determining how software can deliver such value is the essence of software innovation.

Software innovation is not a well established term and considering the challenges faced by our industry surprising little research is being performed within this field. The contributions that has been made spans a range of organisational levels and development stages [2], but not much of it is related to the activities performed by software developers on a daily basis. As software engineers we are not as interested in innovation on an organisational level. Instead our interest lies within embodying innovation into the everyday activities performed by the software development team.

For us as pragmatists, software innovation is about the exploration of new ideas – carried out by a software developer – that lead to new and previously unseen value propositions. The challenge arises because the current development methods, supporting the development team in their daily activities, provide little encouragement for such exploration [2]. As mentioned above innovation is the successful exploration of new ideas and can be observed as social changes. To facilitate such we not only need to focus on *how* to develop solutions but more importantly *why* we develop them and *what* we want to achieve. In the modern development methods strive to meet user requirements emphasis is put on how to most efficiently develop a quality solution. Why to develop it and what to achieve is not addressed much. This is a subject which will be elaborated on throughout this report.

Chapter 3

Essence

In this chapter we give a short introduction to the Essence framework. This description is mainly based upon [1], [3], [2]. The Essence framework has been key aspect in our research in both our 9 th and 10th semester projects. In our 9 th semester project we utilised most of the structures found in Essence to experiment with an innovation project. This project is described in full details in Chapter 4.

Producing quality information systems is in many cases a highly complicated task involving a process that requires complex knowledge into multiple domains. Software development methodologies are frameworks that help structure, plan and manage this process [40]. There are different approaches to overcome the problems software teams faces during the development and help manage the software project. Commonly these are categorised into two paradigms: The traditional and agile paradigm. While development methods inspired by the traditional paradigm focus on a sequential, planned and well documented process [40], agile methods adapt an iterative and evolutionary approach as a response to the ever changing requirements found in both user and technology domain [30]. However, the most important concern addressed in both these paradigms is producing quality software, the most time and cost efficient way.

Essence is a software development methodology that facilitates creative thinking and the production of high value solutions targeting the innovative software team. This is a contrast to more well known development methods found within the traditional and agile paradigm. Here focus is on customer compliance and meeting their requirement, accomplished by developing valuable solutions to the challenges posed by customers. This is not to say that meeting customer expectations and producing highly efficient and quality products are not crucial in innovation projects. However, focus in Essence is on exploring the opportunities that both can be found within the boundaries of the user and technology domain, by creating high value solutions that exceeds the customers expectations.

Inspired by the agile paradigm Essence also promotes close collaboration with the customer and flexible development that allows for constant changes. Nonetheless, the Essence framework diverts from agile development philosophy by making innovation the main ambition for the development project. As mentioned Chapter 2 innovation can be seen from different levels and perspectives and in Essence innovation is addressed on project level.The goal in Essence is to bring software innovation onto team level from when the project starts till it ends and mature potential novel ideas in an iterative and evolutionary manner. A creative and innovative process is more than often characterised by chaos, as we work in areas driven by uncertainties. These uncertainties come as natural element as software innovation deals with the realisation of new and novel ideas matured into software products that meets market needs adding value in new ways.

Essence is a framework that focuses both on controlling this chaos and to inspire and mature new and innovative ideas into a software product. As new and creative ideas often happens spontaneous – the A-ha moment, which often cannot be created on demand, it entails a highly chaotic development process. To overcome this chaos, Essence suggests that we work with structures rather than having focus on controlling the process itself. The structures in Essence is not a set of methods focusing on the creative part of generating ideas.There are many well documented methods and tools that can be brought into play for this [12]. Instead these structures should facilitate innovative exploration and experimentation. Having structures allows the team to act spontaneously as it will not have to follow some predefined method.

To help discuss Essence structures, values associated with each structure will be presented in our description of the structures below. These values are inspired by the outcome of the clash found between the traditional and agile paradigm probably best stated in the Agile Manifesto [9]. The manifesto was written in 2001, at a summit of 17 practitioners of different development methods, but all interested in iterative and agile methods. The outcome was the Agile Manifesto and the Agile Alliance which explore and apply agile principles and practices in software development [30] [4].

The Agile Manifesto is comprised of four values (Appendix A presents the four values of the Agile Manifesto) and defines how agile software development should be approached. It was presented as a critique of the traditional paradigm. Even though values of the traditional paradigm have never been explicitly stated, the fundamental ideas behind the reasoning on the paradigm's take on software engineering is well addressed in the Agile Manifesto.

In [2] Aaen moves beyond the values found in both these paradigms. The ambition is a search of values that stimulate and steer software innovation. By bridging the two paradigms Aaen has located four values that overcome the struggle between the traditional and agile paradigm and will help move Essence towards values supporting software innovation. We present these values in association with the four Roles of Essence. The substance of the values can also be detected in our description of the four View structures. For more information and discussion of the bearing of these values we refer to [2] and [3].

3.1 The Four Roles

Roles are one key structure in Essence. The roles represent different stakeholders involved in the project e.g. customers, developers, users, managers, and consultants. Different stakeholders have distinct knowledge, insights and interests in domains covering different areas of the project. A user will normally have special interest in the user domain, whereas the developer will possess distinct insight into technology.

In Essence there are four roles named Child, Responder, Challenger and Anchor. These roles are generic, influenced and compatible with many of the roles found in agile development methods. The roles are assigned permanently when the project starts and allocated to the team members that are best suited to fulfil a certain role. This assignment should be the person with the most interest, experience and expertise in a compatible domain, so the role match the personality of the team member. There is one fleeting role in Essence and this is the Child role. Why this is the case we explain below where all four roles are introduced in details.

3.1.1 The Child

Inspired by the agile paradigm, close collaboration between the customer and developer team is also nurtured in Essence. Close collaboration with the customer/user in agile development methods helps the developers to better understand the user context, as the team continuously work on the solution. This opens up for ways to evolutionary deduce requirements and create acceptance tests that help determine if requirements are met [30]. However, instead of assuring compliance with requirements, this relationship in Essence is more about reflection. In an innovation project, any challenge a team is confronted with, must be open for interpretation by allowing it to be questioned by both the customer and developers. This helps to fully explore and mature ideas on both user and technological domains, so new and interesting ideas can surface. In Essence this joint exploration is nourished with the Child role. As the only fleeting role it is shared among both customers and developers.

The role of the Child is committed and obligated to speak her mind, whenever she has something on her mind. The Child is seen as the free optimistic spirit willing to learn and open for new interpretations, but nonetheless unbounded by any rationale. The Child is interested in exploring and discovering interesting ideas for new or improved features with no concern about the feasibility of the actual implementation aspects. It means ideas that contradict decisions made earlier, are always welcome among other children.

The main stage for this role is the ***Paradigm View*** and the value that is attached to this role is: ***Reflection over requirements***.

3.1.2 The Challenger

To overcome the many uncertainties found in an innovation project the role of management becomes paramount. Management is usually representative for the customer and highly involved in planning the course of the project. Traditionally, planning in software projects is based on the resources available and delegations and assignments of known tasks. But unlike values found in both traditional and agile methods, management in innovation project involves exploration and interaction into the project, but here planning must be open-ended to allow for the entire team to experiment with visions for the project. This form of management requires insight into both the external and internal forces that drives the project, but also strong improvisational skills as uncertainties take a central role in innovation projects.

In Essence the role of management is called the Challenger. The role assembles the role of the customer and to some degree the project manager found in the traditional paradigm and the onsite customer or project owner found in various agile methods.

The Challenger is the main contributor of posing a challenge for the team to continuously work and experiment with. As the Challenger is the main character who possesses a profound understand of the application domain, she has to be fully aware of the boundaries and opportunities found within the challenge and relate this to the current context. The Challenger is also responsible for having this challenge rendered into a viable project vision, representative for the overall ambition of the team. To help mature and realise this vision, the Challenger mainly focuses on making things happen and therefore has to able to prioritise and choose among the new ideas and features that pops up during development.

The role of the Challenger is associated with the ***Project View*** and the value that is attached to this role: ***Vision over assignments***.

3.1.3 The Responder

In any software project going from ideas/requirements to an actual software product requires technical expertise. Traditionally, this expertise derives from the software developers who hold the necessary and hopefully sufficient knowledge of the technological domain to vision and model requirements into a software solution. To be a developer following the agile paradigm requires skills related to competency, collaboration, motivation, continuity, trust, and respect leading to greater sense of ownership and commitment to the project [4]. Also tools and techniques such as pair programming, test-driven development, design patterns and code refactoring are all promoted to overcome to the inescapable changes which occur in an agile development project [8].

In innovation projects all these skills are still essential. But to add to these skills, the innovation developer needs to posses a technological foresight in choosing technology, which will help to see unexplored potential in the technical aspects of the solution. It means that the developer both has to see the potential and boundaries in the chosen technology and be able to explore and experiments with these.

The role of software craftsmanship is dedicated to the Responder in Essence. It is her job to respond to the challenge posed by the Challenger by looking at the project vision from the inside. This is accomplished in close collaboration with the Challenger where the Responder is responsible for keeping an overview of relevant technologies and help the Challenger to choose among these.

Responders are main contributors to designing and developing a technical platform, by choosing technology that affords new options and possibilities. Their insight will hopefully help to shed new and alternative ideas associated with the solution space. There are normally more than one Responder in an Essence team, where each Responder adds different expertise to the team and insight into the technology. A Responder has to be a team player but also capable of promoting her own ideas and take on the technology.

The role of the Responder is associated with the ***Product View*** and the value that is attached to this role: ***Affordance over solution***.

3.1.4 The Anchor

The process and progress of a software development project can often spiral away from a predictable path. To keep a project flowing towards a solution that corresponds with the needs of the customer and disposed resources, someone has to keep an eye on how things are done and make sure that the project is running smoothly. This includes providing tools and techniques that allows the developer team to be operational and productive when obstacles are met. Traditionally, it is the responsibility of the project manager to control the process of the project. This is normally accomplished through standardised monitoring, planning, evaluation, resource allocation, and measurements. In agile teams process, facilitation normally stem from roles such as the Scrum Master in Scrum or coach and tracker in XP. Instead of standardised control measurements, agile teams normally rely on incremental adaptation of work practices to keep the team productive.

In software innovation we need to facilitate innovation and are not that focused on stabilising work processes. In innovation projects the team is often surrounded by chaos and uncertainty and thus some stabilisation to the process is needed. Therefore we need someone who is responsible for the project is always progressing towards the project vision. This is accomplished by ensuring outside noise is filtered away, providing tools and techniques that motivate creativity and idea generation and evaluate and maturing these into productive visions.

In Essence we call this role the Anchor. The Anchor in Essence is responsible for ensuring facilitation of thorough and neutral evaluations. These evaluations concern ideas, visions, solutions, and current research. The Anchor will provide and adapt tools and methods that facilitate such evaluation. The Anchor is also responsible for making sure that the team is well functional. This is accomplished by ensuring that problems within the team is handled before they become disruptive, that discussions are kept on track, and the process itself is kept creative and open for new interpretations.

The role of the Anchor is associated with the ***Process View*** and the value that is attached to this role: ***Facilitation over structuration***.

3.2 The Four Views

As idea generation and experiments are an essential part of an innovation process, keeping an overview of the result of these activities can be pretty difficult if the approach is not somewhat structured. In Essence we utilize another key structure – Views – as a place to organise and manage the innovation process. There are four Views in Essence, each covering core aspects concerning the management of the project. They are named: Paradigm, Product, Project, Process.

A View in Essence is a physical location, either represented by a digital smart board, white board or a similar material that can store information. Views allow the entire team to keep a visual overview of the entire project, but at the same time allow team members to work on different and diverse concerns on separate Views. As the team works on one View there will naturally occur temporal differences between the Views. These differences are not an unsought property as the contrast may bring reflection and new and alternatives ways of seeing the solution. The four Views are described in greater details below.

3.2.1 The Paradigm View

Understanding a problem domain can be a very complex task to undertake for a team of software developers. Working with innovation only makes this domain harder to capture and understand for both the customers and developers as the innovation entails a vast amount of uncertainties since we almost never know beforehand how the solution will turn out. As the problem domain is situated with users and their context, while the solution specific domain knowledge is located with the developers, it is desirable that knowledge is shared among these two groups. However this kind of knowledge can be difficult to share. Firstly, can domain specific knowledge appear *tacit* – information that is hard to explicit formulate [23]. Secondly, this knowledge is often *sticky* – information that is hard to gather and transfer among domains [15]. This is where the Paradigm View in Essence comes into play.

The Paradigm View sees the challenge from the perspective of the user. Most of the idea generation happens here. Within the boundaries of the challenge we explore and experiment with the use context, while taken the technical aspects into consideration. Most ideas are contrived here, in close collaboration between the customer and the developer. This relation is cultivated at the View, as it help us contrast our perception of the solution.

To help conceptualise and model ideas, use cases or use scenarios fitting the current understanding, are often developed on this View. To help externalise concrete ideas, prototyping is often applied as activity. It means the Paradigm View holds information related to the use context such as use cases, use scenarios, state diagrams, actors or stakeholder behavioural patterns, organisational charts, maps and prototypes.

3.2.2 The Product View

The Product View is used to explore the technical aspects of the project. This is where the solution is seen from the ''inside" and from the perspective of the developers. The View represents how to build the solution with focus on the overall architecture and how possible features can be implemented. As knowledge into the technology is gained, new ideas may emerge which is then reflected back to the Paradigm View.

The reciprocal relation between problem and solution domain is often quite complex to comprehend in a software innovation project. Ideas comes and goes and the ones with the most potential have to be matured and developed into final software products. As the solution space itself will through the duration of the project be explored and exploited, choices surrounding the design of software and hardware components have to be kept as flexible and simple as possible so changes and new components easily can be added over time. The technological foresight of the developers is also crucial as choices of technologies have to radiate affordance.

Information kept on the Product View is related to how we vision the solution could be build. It includes different and refined perspectives of the architecture, UML class and component diagrams, algorithms, and possible deployment diagrams.

3.2.3 The Project View

The Project View is all about project management. However in Essence focus is promoting innovation and not cost and risk driven. To nurture innovation the challenge here is not only to meet customer requirements but also to reflect on the needs of user and exceed their expectations. Such reflection happens as a joint exploration and discovery between users perceptiveness of the problem domain and developers insights into the technological. This will need to be managed.

As an innovation project is circumvented with uncertainties, the managerial tasks involve nurturing and maintaining an overall project vision. The vision is a reference point utilised to steer the project towards. This vision represents the ambition of the team – a common goal for the project. There are several structures in Essence that can be applied to capture a project vision. In the following two of these are described in greater details:

The Toulmin Structure

The Toulmin structure in Essence is based on Toulmin's argumentation model [55]. In the model the challenge and the vision is reciprocal representation of the main idea. In this model the main vision for the project is conceptualised embedded in design rationale. This rationale is expressed in different components planted in the structure. These components can be seen in Figure 3.1.

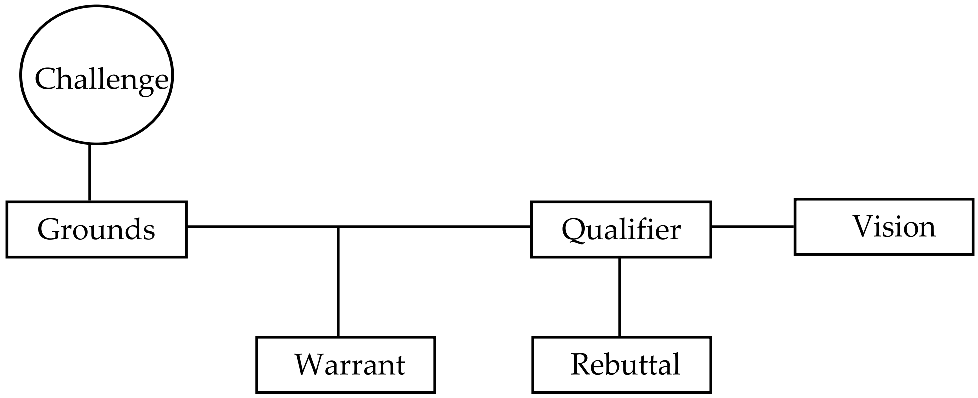

Figure 3.1: The Toulmin Structure

The Challenge represents the overall problem which needs to be solved by the team. Within this the context the vision is defined. The other elements in the structure represents the response to this challenge. It answers the whole or more commonly parts of the challenge.

Grounds is where facts are presented as problem statements. It defines why a solution is needed, by offering reasoning and evidence for this.

Warrants are utilised to argue the relevance of the Grounds. These should help to legitimising the solution. As the Vision presents a possible solution to a challenge, Warrants present argumentation for why this current solution should even be attempted.

Qualifiers indicate a leap from Grounds to Warrants. These will help to refine and scope the Vision even further, as specific issues can be raised as to how the solution may be realised.

Rebuttals serve as counter arguments to the issues pointed out in the Qualifiers. Here the team is able to see drawbacks with solution and present reasoning as to how these can be resolved or point to an approach that may help solve these.

The Vision is seen as a response to the Challenge – the main idea. It may solve only parts of the challenge. The Vision should be seen as dynamic component, which will change as knowledge and perception of the different domains develops over the project period.

Prototypes

In an innovation project where we experiment with our design, prototypes are an commonly produced artefact. A prototype is a means that can help externalise our understanding of an idea or part of an idea into a concrete manifestation that can be shared among team members. As we explore and experiment with the materials that we have at hand, the prototype also becomes a means for reflection. According to Schön it is through our exploration with prototypes we learn more about the design problem itself, by letting the situation talk back to us and reflect upon the design itself [46]. This reflection will hopefully open up for new and alternative way of perceiving the overall idea.

Following Schön's view, prototypes can also be seen as a tool for traversing the design space. [32] suggest an anatomy of the prototype that helps understand the complexities involved in defining what a designer seeks to achieve with creating a prototype. They suggest that a prototype is only one manifestation of an idea and should be seen as a tool to externalise a concrete aspect of the idea - not the whole idea. A prototype is characterised by different dimensions. These dimensions will help the designer to filter different aspects in prototype to single out features to be demonstrated. At may be that the designer only wish to demonstrate the appearance such as colour, size and shape and filter out any functionally or interactivity. As part of the evaluation process, the prototype will become a means for the designer to allocate new problems and seek alternative ways to solving these problems.

One interesting observation can be made of prototypes. As no rationale are captured in these models and they are open for interpretation, different people may interpret the prototype in different ways.

3.2.4 The Process View

Maturing and framing ideas into viable solution is in Essence one the core aspects of process facilitation. It means we need a place where we evaluate what we have achieved so far and overview on how we progress in investigating new and alternative ideas. In Essence this happens on the Process View. Activities performed here are about creative facilitation and evaluation. Here we discard ideas with no potential and decide what ideas and technologies to keep working on. We may also decide that an idea or technology have interesting potential and therefore needs further research.

To help evaluate ideas, technologies and research different tools can be applied. SWOT(Strength-Weaknesses-Opportunities-Threats) and PMI(Plus-Minus-Interesting) are some of the tools that can employed. These tools will help the team to see and evaluate any advantages or weakness of a given aspect and thereby help to weed out ideas or technologies that show no potential and mature and seek those that show an interesting potential.

Summary

In this chapter we looked at Essence - a software development methodology targeted at the innovative development team. We described in detail the structures and values which this framework is built upon. This includes the four View structures and what information and activities are related to these. We also looked at the four Essence roles and described how these are related to roles in other paradigms and how they differ and contribute in an innovation context. In our 10^{th} semester project we will concentrate mostly on the role Responder in our experiments with focus on the Project View.

Chapter 4

Previous Work

This master thesis project extends on a foundation built on our 9th semester [16]. We had two goals with the preceding semester. The first goal was to experiment with how Essence, introduced in Chapter 3, could be applied in the design and development of a real world application. This application, called PhysioSphere, assists a patient in their rehabilitation after a physical injury or operation. This is achieved by using a Microsoft Kinect sensor to verify that exercises, performed in a patient's home, are done on time and correctly. The second goal with the 9th semester project was, through this experimentation with Essence, to locate a research topic for our master thesis project.

This chapter outlines the work accomplished in our 9th semester. Firstly, an introduction to welfare technology, rehabilitation and physiotherapy is given. Secondly, the development of PhysioSphere, including the process and final product will be presented. At last, a small reflection is presented related to the outcome of the project. It is this outcome that has directed us towards the research being performed in our master thesis project.

4.1 Welfare Technology, Rehabilitation & Physiotherapy

Due to better standards of living the life span of the citizens in the western world are ever increasing. As a result the number of citizens, like elderly, who are in need of welfare services are steadily growing while the number of people in the workforce are decreasing [20].

As our research last semester showed a welfare society like Denmark, is especially vulnerable towards this trend. To maintain our current quality of welfare, new and innovative approaches are needed that allow us to provide services more efficiently and at a lower cost.

A way to achieve this is through the application of technology. In Denmark such technology is increasingly becoming a source of interest and research. In later years the Danish government has been promoting innovative usage of technology that will have an impact on how our welfare sectors are run. According to the Ministry of Social Affairs and Integration welfare technology denotes new technology and working procedures, which can reduce the time and resources spent on social welfare services [49]. Welfare technology can also be the means to enhance the quality of the provided services or to improve the working environment for the professionals who work within the welfare system.

Even though the term has gained much attention, our research last semester showed that the potential in welfare technology is mostly unexplored. Studies show that application of welfare technology can be beneficial for all levels of our society [28]. The citizens can benefit from improvements to their quality of life. The practitioners may benefit by allowing the technology to improve the work environment and work processes leading to more efficient use of resources. The government benefits from welfare technology by potentially decreasing the cost of running an adequate public welfare system. As a future benefit the marked potential for this kind of technology is largely unexplored. If the private and public sector acts quickly, companies might be placed in a market leading position where the export of such technology can be a source of economic growth [28].

To reach such a position, the public sector, private companies and research institutions must work together to explore the opportunities in the technology. Such exploration is the essence of innovation and is the area in which we believe a valuable solution could be created. PhysioSphere can be defined as such a technology. Before presenting the development of PhysioSphere a short introduction to rehabilitation and physiotherapy is given.

4.1.1 Rehabilitation & Physiotherapy

A major goal of rehabilitation is to make quantitative and qualitative improvements in daily activities in order to improve the quality of independent living for patients who have experienced some form of disability [51]. In a welfare society like Denmark such services are provided to citizens by the government. As such there is much potential in exploring how the technology can be used to better provide these services.

Rehabilitation is the process of restoring a patient back to herself prior to illness [41]. The term rehabilitation covers a wide range of disorders and rehabilitation programs are divided into different types according to a patient's needs. Examples of such types include occupational rehabilitation, cognitive rehabilitation and physical rehabilitation [42]. Physical rehabilitation is used for patients who have suffered muscle or bone injuries. It is the task of the physiotherapist to design and supervise a rehabilitation program that quickly and effectively assists the user in recovering from such an injury.

A rehabilitation program often contains a number of simple exercises that have to be performed repetitively over time to slowly build up strength and agility of the muscles or bones. Instructions in performing these exercises are given by the physiotherapist in a clinic. Here the patients routinely meet with the physiotherapist to evaluate and refine the program. The majority of the exercise sessions however has to be performed by the patient at home. Through our research, backed up by our meetings with cooperation partners, two main problems with these home-rehabilitation programs were detected. Firstly, exercises must be performed correctly to ensure optimal rehabilitation and to avoid secondary injuries. Secondly, as studies involving the rehabilitation of stroke patients show, it is very hard to maintain the motivation to perform these exercises. This is partly due to the repetitive or boring nature of the exercises and partly due to the psychological conditions, like depression, which often follows an injury [33][17].

Our initial vision with PhysioSphere became to develop a solution that overcomes these problems by utilising the Microsoft Kinect sensor to verify that exercises are performed correctly and on time.

During our research several examples have been located that illustrate how technology can assist in meeting these challenges. It is well known that games are a great way to excite, motivate and challenge people [17]. By creating games where the exercise sessions a part of the gameplay, repetitive tasks can be made agreeable or even fun, while removing a patient's attention from the possible pain involved.

Such games are known as Serious Games (SG) [33]. These games cannot only be utilised for instruction and motivation, but also to gather knowledge about a patient's performance. Our research last semester showed that much research are done into such games [21].

Another approach to motivate patients and provide feedback of their performance, is through Augmented Reality (AR). Here sensory equipment and cameras are utilised to capture a patient's body posture. By feeding the video image back to the user, together with overlayed feedback, a user may be able to continuously self-adjust incorrect movements without the supervision of the physiotherapist [53].

4.2 Development Process

As mentioned in the introduction to this chapter the first goal with the 9th semester project was to experiment with the use of Essence in the development of PhysioSphere. The process was experienced as highly iterative and chaotic. This Section provide an overview of this process and present the final PhysioSphere prototype. First our use of cooperation partners is introduced followed by a quick overview of the Microsoft Kinect sensor. The two following sections present the development process and the final PhysioSphere prototype.

4.2.1 Collaboration Partners

Working with Software Innovation require the development team to explore and combine knowledge from a number of domains. To effectively see how a technical solution can offer value much insight into the use domain must be gained.

This insight was gained through a variety of sources but most importantly through the use of collaboration partners. Working with these have allowed us to gain and verify domain knowledge and to continuously refine the vision of the project through demonstration and evaluation of prototypes.

Through the 9th semester we collaborated with two different groups of physiotherapist, which each contributed to our understanding of the domain in different ways. The first group approached was Træningsenheden Nord. Træningsenheden Nord are responsible for providing rehabilitation services to citizens in the Aalborg metropolitan area. Initially it was planned to hold regular meetings with these physiotherapists every second week. But due to their busy work schedules only two meetings were held. The main outcome of these meetings will be presented later.

To further assist us, a second collaboration partner was introduced. We approached two physiotherapist students from University College Nordjylland halfway this semester. They assisted us in building a exercise dataset used throughout the development for evaluating the exercise recognition algorithms. Additionally they provided some more technical and theoretical insights into the domain of physiotherapy which provided an alternative view on this domain.

4.2.2 The Kinect Sensor

From the beginning of the project it was clear that our project would be concerned with the utilisation of technology to assist the rehabilitation of patients. The initial motivation was to explore how such assistance could be provided through the de-

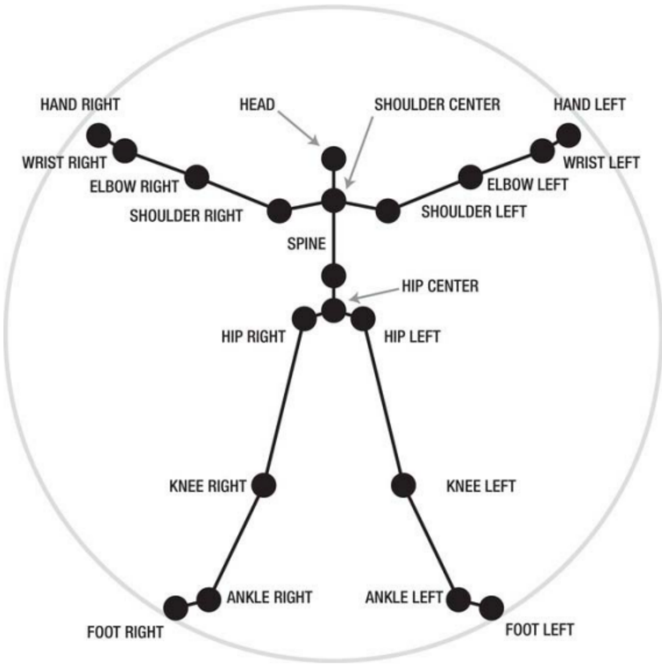

Figure 4.1: Skeleton illustrating the 20 joints tracked by the Kinect sensor [58].

tection and tracking of movements. To narrow the scope a little some research was placed into selecting a technology early that could enable such tracking. The device selected for this purposes was the Microsoft Kinect sensor. The following section present the opportunities this sensor offers, together with the motivation behind selecting it.

4.2.3 Kinect and rehabilitation

The Microsoft Kinect is a motion-sensing gaming accessory developed by Microsoft. It utilises a combination of cameras, sensors and advanced algorithms to detect and track movements performed. It was originally developed for the Xbox 360 gaming console allowing gamers to interact with the console through a natural user interface (NUI) involving gestures and voice commands. This makes the Kinect sensor useful in our quest to recognise and verify exercises.

The main attraction of the Kinect sensor is a feature known as skeleton tracking. Image data from a colour camera and depth measurements from an infrared camera are gathered and analysed to construct a virtual representation of the human body. This representation consists of twenty different points each correlating with a joint on the human body. These points can be connected with simple lines to form a human skeleton as can be seen on Figure 4.1 The position of each joint in this skeleton is continuously tracked and updated as the person in front of the sensor moves.

Compared with other available technology we believe the Kinect provides a higher degree of flexibility and hence more potential for exploration. Devices like the Wiimote, the Playsation Move controller and bluetooth accelerometers have all been utilised in earlier research projects [52] involving rehabilitation. Common for these however is the limitation that only a single joint movement can be tracked for each

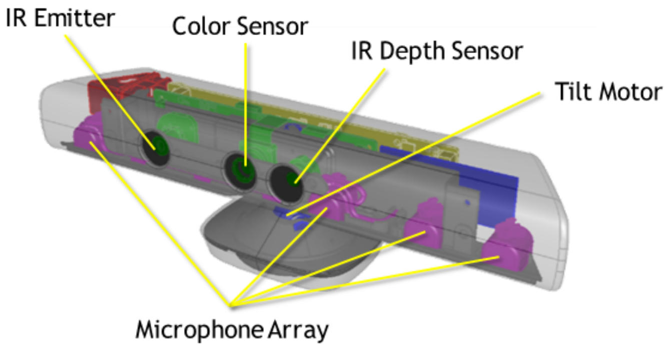

Figure 4.2: This illustration show the sensors integrated in the Kinect [36].

physical device [52] making tracking of complex movements involving multiple body parts unreliable.

The flexibility of the Kinect is also evident from the number of sensors and integrated functionality accessible through the Software Development Kit (SDK). Besides skeleton tracking the Kinect supports the capture of video, an advanced microphone array and functionality for voice command recognition. The SDK also includes a framework for recognising hand gestures. While not necessarily needed initially these options afford later exploration.

The Kinect sensor does have some limitations. It provides lower refresh rate than some of the other sensors investigated. Further, while the SDK does offer information about a joints rotation, it seems to be highly unreliable. After some experimentation we are not sure that the Kinect sensor is suitable for recognising and tracking fine motor skills due to the low resolution of the depth camera.

4.2.4 Development Process

The initial goal with PhysioSphere was to assist a patient in ensuring that a rehabilitation program is performed correctly and on time. Essence was applied to assist us in this process. As introduced in Chapter 3 a main concept in Essence is the continuous development and maturation of ideas captured as a Project Visions. This maturation happen as new insight are gained through exploration of technology, development of prototypes and interaction with domain experts. This does lead to a highly chaotic development process involving many uncertainties. In the following a short overview of the development process is given. This overview is presented as a timeline.

The timeline is split into three chunks, which we chose to call configurations. Each configuration captures the work carried out on the four Essence Views in a given period of time. A change in configuration is denoted by a substantial change in Project Vision initiated by meetings with our cooperation partners.

Our motivation for presenting this timeline is not to give a comprehensive description of every activity performed during this time period. Instead we want to show how the project has matured throughout the semester as our understanding of the technology and use domain has evolved. We illustrate this by presenting our challenge and vision for each configuration. In the following the three configurations are presented.

4.2.5 First Configuration - Rehabilitation

Our effort within this time period was split in two directions. Foremost, we aimed to obtain an initial understanding regarding the area of physiotherapy and rehabilitation together with challenges faced within these fields. Secondly, we wished to explore how technology can be utilised to respond to these challenges. The result, three low fidelity prototypes, was produced and brought to our first meeting with Træningsenheden Nord.

Challenge:

Patients who have been through an injury or operation are through the Danish welfare system obligated to receive rehabilitation. Such rehabilitation can be provided through a number of sources and depends heavily on the individual patient. A rehabilitation program consist of clinic sessions where the patient, alone or in small groups, perform exercises while being supervised by a physiotherapist. These sessions are supplemented with exercises that have to be performed in a patient's home. A problem with these home exercises is that they are often performed incorrectly or not even performed at all. This can potentially lead to increased rehabilitation time and other complications.

Project Vision:

To meet this challenge a number of digital and analogue prototypes were produced. The digital prototypes were designed to help communicate how the Kinect sensor can be utilised to evaluate the correctness of an exercise. This primarily involved the presentation of the skeleton tracking feature introduced in Section 4.2.3. Three approaches to motivating the patient were suggested and captured in analogue prototypes. The first involved the use of the Kinect to create virtual many-to-many group sessions. The second focused on a one-to-many system where a single physiotherapist is online and can switch between online patients to ensure that they perform their exercises correctly. The last prototype captured what we call a electronic tell-tale. Here the system autonomously verifies that exercises are performed correctly and on time and only signals the physiotherapist if problems are detected.

4.2.6 Second Configuration - Telemedicine

During the first meeting with Træningsenheden Nord, the prototypes were presented and evaluated. They sparked a discussion related to the challenges that the two physiotherapists experienced in their daily work. They concurred on our judgement that supporting the user in performing their exercises at home is of high importance. Today such support is often provided by travelling back and forth between clinic and the home of a patient costing much in time and resources.

Challenge:

As the physiotherapists pointed out during our meeting: Technology can assist, but never replace the human interaction between physiotherapist and patient. Today much time and effort is spent on transporting patients or physiotherapists back and forth between clinics and homes. Technology is not only needed to verify that exercises are done correctly and on time, but also to facilitate the interaction between physiotherapist and patient. By limiting the time used on transport, more time can be used dealing with the individual patients.

Vision:

As a response to the challenges above, the vision was modified to be bidirectional. Besides providing a simple system for recognising and verifying exercises, the physiotherapists expressed a desire to utilise the integrated microphone and camera of the Kinect sensor for interacting with the patient. Such interaction would allow the physiotherapist to instruct the patient in performing exercises without the need for transportation to/from the clinic.

4.2.7 Third Configuration - Rehabilitation

In the second configuration much time was spent on exploring the implementation of the video conferencing system allowing the physiotherapist to assist a patient remotely using the video/audio features of the Kinect. This did prove to be a far greater challenge than initially expected and it was decided to scrap this part of the vision and focus on the recognition and verification of exercises. In the beginning of the third time period two meetings were held. One with Træningsenheden Nord and a second meeting with the physiotherapy students. How these meetings changed our challenge and vision are expressed below.

Challenge:

As mentioned in Section 4.2.1 the physiotherapy students were involved to help us build a testable dataset of exercises. This meeting however turned into a discussion of the possibilities provided by PhysioSphere. Here they expressed a real concern with the current rehabilitation programs which, in their experience, are not sufficiently adapted to the individual abilities of the patient. According to these students many physiotherapists can have a tendency to just select and apply standard programs.

Vision:

The vision in this iteration was to utilise the Kinect sensor to verify that an exercise program is followed and that the exercises are performed correctly. Further, data related to the quality of the movements, like sign of tiredness or overeagerness, should be detected and presented for the physiotherapist allowing her to continuously adjust the program to better suit the abilities of the patient. The prototype manifesting this vision is presented in the section below.

4.3 Product

Throughout the first semester several digital prototypes were developed. The prototypes serve as concrete manifestations of the Project Vision and have as such continuously evolved as the Project Vision has matured. These prototypes have served a multitude of purposes. Firstly, they work as a great communication tool between the development team and the cooperation partners, facilitating idea evaluation and discussion. Further, innovation is more than just generating ideas. It is also about exploring these and turning them into viable products. Creating a finished product was out of scope for the semester but the prototypes developed allowed us to explore the viability of ideas. In the following our main prototype is presented.

4.3.1 PhysioSphere Prototype

The ambition with the prototype has been to produce a concrete manifestation of the project vision. As such the focus has been on designing a flexible, testable solution that recognise and verify exercises using the Kinect sensor. The main prototype consists of a single window shown in Figure 4.3.

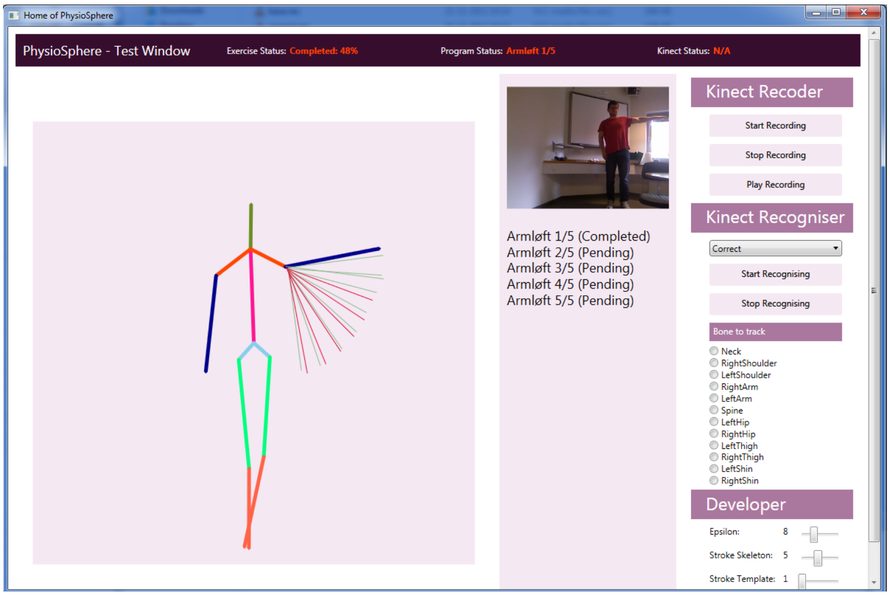

Figure 4.3: Our main prototype in the 9th semester project.

This window has not been designed with usability in mind. Instead it is a proof of concept showing the potential in recognising and verifying exercises using the Kinect sensor. The following functionality is embedded in the prototype:

- Recordings of exercises can be made using the buttons in the top right.
- These recordings can be replayed, recaptured or saved as an evaluation template. This evaluation template is used as reference point when comparing exercises performed in real time.
- When the evaluation template is saved a simple rehabilitation program is generated consisting of 5 repetitions of the captured exercise.
- When the recogniser is started, the user's movements will be tracked and repetitions will be either accepted or failed depending on the measured performance.
- During recognition simple red and green lines will be provided as simple visual feedback mechanisms to the user.

One of the unique challenges in working with innovation is the amount of uncertainties. A prime source of these uncertainties is the constantly changing vision for the project. The product must contentiously be adapted to reflect such change.

The problem of responding to changing visions is not unlike the problem of responding to changing requirements known from agile development. To deal with these changes many of the principles known from agile development have been adopted in this project. These include the creation of a simple, flexible and testable architecture and heavy use of refactoring throughout the project [30].

In the following a short overview of the implementation is provided. Our implementation of recognising gestures is inspired by [19] and the technical response is more comprehensively documented in [16].

Recording Exercises To recognise that exercises are performed correctly mechanisms are needed to record and represent these. In PhysioSphere the mechanism used for recording is called the Exercise Recorder. When started, the recorder hooks into the skeleton stream offered by the Kinect sensor and save it to intermediate storage. From this storage the recording can be played back, recaptured or saved.

Saving Exercise From the raw skeleton data an, evaluation template can be created. This template is used as reference when comparing and verifying exercises. Simplified, a template store a set of body postures. Each of these is called snapshot.

Recognise Exercise Recognising exercises is performed by comparing real-time skeleton data from the Kinect sensor with the snapshots saved in the template file. The list of snapshots can hence be seen as a list of positional checkpoints that a user has to repeat. A simple form of providing visual feedback has been chosen. The red lines denote these checkpoints and the colour of the checkpoints change to green whenever a checkpoint is successfully compared. If all checkpoints are recognised in the right order, the repetition is accepted as being correct. The epsilon value, changeable in the user interface, is used to define the error tolerance used by the recogniser when comparing real time data with snapshots.

The prototype works well as a proof of concept. The coloured lines are a simple proof of concept showing how the user can receive visual feedback during the workout itself and can easily be extended at a later time. Currently the system does not provide feedback on the quality of a performed exercises. The use of snapshots or checkpoints, potentially allows for such functionality to be implemented.

4.4 Outcome

The project has allowed us to experiment with the use of Essence in the development of a practical application. The process has been both exciting but also quite frustrating at times. Some of this frustration is sourced in the amount of uncertainties experienced by the developer and a lack of tools for handling these. To understand problem we must look at the unique responsibilities placed on a Responder in Essence.

In Essence the Responder is seen as the hero [3]. The Responder's responsibility goes beyond designing and implementing solutions in response to requirements. This is well captured in the two Essence values *affordances over solutions* and *reflection over requirements*. In Essence affordances are concerned with technological foresight and potential. This value implies that a Responder must consider what a design can afford to the user beyond what was required to solve the needs that have shaped the design in the first place. This suggests that user needs are not something that can be determined beforehand but has to be determined through exploration and reflection. This is captured in the value *reflection over requirements*.

The requirement for the Responder to answer not only how a design should be implemented but also what the solution affords to the design and why this matter encircle us with uncertainties. In our 9^{th} project this has been experienced as a chaotic process. We can illustrate this problem in Figure 4.4.

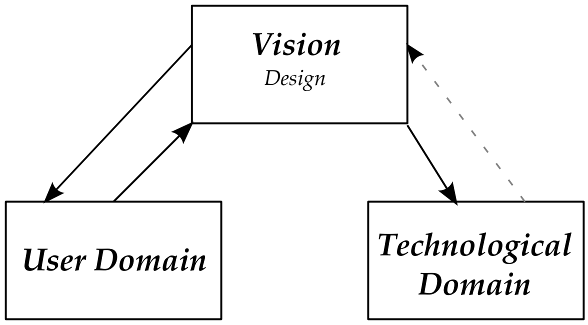

Figure 4.4: Reflection

Uncertainties always surround a development project. These normally involve understanding what the user needs and how to transform such into a technological solution. To help us understand user needs we have a range of tools including different types of sketching, prototyping and use-scenarios development [18], [10], [43]. To transform such needs into a technical design we can apply different kinds of modeling methods including OOA&D [35] and construct designs using good architectural principles [7]. All these help us manage these uncertainties and transform them into knowledge that assists us in constructing our solution.

The problem arises because the Essence guidelines and structures provide little supporting in managing the uncertainties that arise when moving in the other direction seen from the technological domain. No current activity or structure helps reflect upon what the technology affords of new opportunities. From our own experiences we have identified three main issues related to exploration of technological affordance. These are:

Prioritising and managing opportunities: In our experience the process of creativity and reflection has both been interesting and fun. But this reflection has also resulted in a huge number of ideas and value propositions that all have to be explored, researched and evaluated to figure out how they affect the vision of the project. Working on too many of these simultaneously would bring progress to a halt.

Resources: As mentioned above our continuous reflection produced a large number of ideas and opportunities that had to be explored. We often made assumptions related to the time and effort needed to do such an explorations. In many cases these assumptions proved to be incorrect as new uncertainties were introduced.

Isolation: In a few instances we had such explorative efforts run out of control. Halting these efforts and reverting the design back to its initial state, proved to be a serious problem due the side-effects introduced into the problem.

The best example from our 9th project was the exploration into enabling a video link between the patient and physiotherapist. We assumed that the public Skype API could be used to implement a prototype a day or two. Realising that this was not the case we started searching for other possible solutions. This research process ended up taking three weeks before realising that providing video communication over the internet is by no means a trivial task. By this this stage our exploratory detour had affected the project challenge and vision, our project prototype and the overall architecture among other artefacts. We believe these side-effect had two consequences. The amount of time and effort placed into this direction made it very challenging to take the decision to halt the research effort. Further it made it very challenging to revert our design to the state before conducting this research.

Part II

Our Contribution

Chapter 5

Research Process

As experienced in our 9th semester project not much research has been conducted within the field of software innovation. We believe the best way to obtain knowledge and experience with this field were through the design and development of a practical project using Essence. This experience has assisted us in locating an area wherein we believe research contributions can be made. This area is related to how we encourage innovation in the everyday activities of the developer in an innovative software team.

Our research into this area has been a process of discovery. In this chapter we outline and reflect on this process.

5.1 Research Question

In this section we present our research question for our master thesis. This question is derived from our practical experiences working with PhysioSphere. When we explore with the technology, we aim to transform uncertainties into technical solutions. As new opportunities are seen, new uncertainties are introduced. This experience showed that the process of exploring and experimenting with the technology in an innovation context often become chaotic.

It is evident that some kind of tool or activity is needed to assist in controlling this chaos. This leads us to to the following research question:

Research Question: *How can we support the Essence Responder in exploring technological affordances in search for new value propositions?*

Our research showed that one answer to this question lies in associating the philosophical paradigm of pragmatism with a Responder's innovative conduct and sandbox this effort in a structured activity. The inspiration for this activity is drawn from a practical tool found in the agile development method eXtreme Programming.

This association is the result of a discovery process into theory related to creativity, innovation and software development. Before presenting our contribution we would to outline this process.

5.2 The Process

Our research process is based upon the idea of Applied Science and problem-oriented research [25]. In Applied Science we deal with problems introduced from practice. What characterise these problems is that they have been experienced in the real world and require further attention and action. A practical problem may be an indication that the current understanding of the theory – Pure Science – is improper. As practical problems are experienced outside Pure Science, we utilise Applied Science to resolve them. In Applied Science we conduct our research among theories and methods that may give a theoretical explanation to our problem. This happens as a highly dynamic interplay between research and practice. Practice produces the practical problems, where the research provides possible answers through theory. This conduct leads to new knowledge that may help locate a theoretical and practical solution.

Our adapted research process is outlined in Figure 5.1. It has been an evolutionary and reflective process to understand problem and solution. As the figure shows the process has in no way been sequential. Our practise during the 9th semester showed us that we had problems controlling chaos and we did not have the theoretical – or practical – background to efficiently define the problem and propose a solution [16]. Through problem-oriented research we have searched for scientific theories and methods that could help abstract our problem and give a theoretical explanation to it. Such an explanation is needed to design and motivate a solution. This research has lead us into many corners of creativity theory and innovation, which has helped shape our solution.

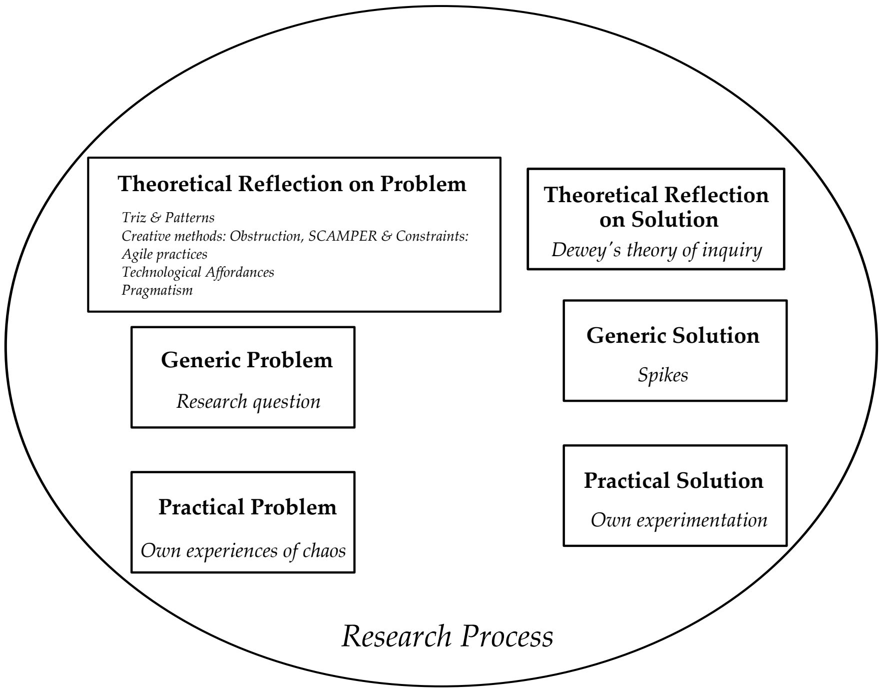

Figure 5.1: Research Process

In the following we will briefly outline the areas of theoretical research, which have helped to produce knowledge of our problem and solution.

Technological Affordance: Technological affordance is a concept proposed by Aaen [3]. The concept was our initial starting point to theoretically reflect on how a Responder in Essence can discover new and unforeseen potential in the technology. In this research we encountered various theoretical perspectives of affordance and how it relates to a context [57], [26], [37]. The concept of affordance does explain the need to understand and explore what a technological artefact affords in a context. However, the research we located does not elaborate on how we as practitioners can conduct such exploration. Nonetheless, the knowledge gained from this research has had a profound influence on how we understand our the problem.

Triz & Patterns: Triz (Theory of Inventive Problem Solving) is one of the biggest studies conducted on human creativity [34]. As many problems and solutions are repeated across industries and sciences, Triz methods suggest that we can learn from these and utilise them to solve almost any problem we might encounter in an inventive context. The Triz researchers studied thousands of patents to determine patterns that link problem and solution together. There are several algorithms and Triz matrices, that assist an inventor in abstracting her practical problem to locate a generic solution. Even though we initially liked the idea of being able to abstract creative problems to solutions, our research also showed that Triz algorithms and matrices can be very hard to comprehend. We also had difficulties relating the abstract problems and solutions to software development. We therefore put this theory to rest. But the principles of abstracting a problem and solution have helped form our understanding of how to construct our solution and that simplicity is a must if it has to be an activity utilised by practitioners.

Creative methods: Obstruction, SCAMPER & Constraints: We researched several creative theories and methods we hoped could be abstracted into an activity used by software developers [48] [12]. One of these was the theory of performing a synthetical halt through a creative obstruction [1] and constraint management [11], [13], [39]. We found this research very interesting. These methods provide new ways to facilitate creativity. Through our reflective research on both problem and solution we realised that we did not wish to propose another method for generating ideas. What we needed was a method assisting in managing the exploration and maturation of novel ideas and none of these provided much support in doing so.

Agile Principles & Practices: We have also conducted research into the topic of agile principles and practices found in eXtreme Programming. This conduct helped us gain a practical understanding of developing software solutions where constant change is embraced. These principles were a source for inspiration in shaping our generic solution. We will be elaborating on this topic throughout the report.

Pragmatism: All of the theories and methods above could to some degree be related to our theoretical problem or parts of it. What all of them lacks is the ability to provide a link between abstract problem and solution. In the research process we located the philosophy of pragmatism. We realised that this philosophy could provide a conceptual understanding to the problems experienced. More importantly it has provided the link between generic problem and solution. The philosophy of pragmatism becomes the foundation for our practices described throughout the succeeding chapters.

5.3 A Case

The theory and concepts presented throughout this project may seem abstract and complicated as it originates from disciplines like philosophy. In the following we present an imaginary case, which is based upon our own experiments conducted during our 9 th project. This case is designed to help illustrate the link between these concepts and in an innovative software development context. We will refer to this case – **the tattletale case** – throughout the rest of the report.

[1] Inspired from the Danish move ''De fem benspænd'' with the two Danish film directors: Lars von Trier and Jørgen Leth

The Context: let us assume that Lise is a physiotherapist in the Aalborg municipality. Her responsibility is to assist people in recovering from shoulder injuries. This process involves the creation of individual rehabilitation programs. The program lasts between 8-12 weeks and contains a number of exercises which must be carefully followed, to build up strength and mobility in the shoulder. Lise instructs the patient Holger in performing these exercises and Holger is invited to participate in group training sessions once a week. Nonetheless, the bulk of these exercises must be performed in the home of Holger, where he himself is responsible for carrying out these exercises and evaluate if they are executed correct.

The Challenge: The unit wherein Lise works would like a solution that allow them to monitor Holger's progress. Currently they can only monitor him when attending the weekly training sessions but they would like to keep an eye on what is going on when he is training at home. What they really need is system that acts as a tattletale.

A small software development team has been hired to meet this challenge. Let us assume we are that software team. The team is made up by a small group of software developers and we got assigned to make our tattletale system. We have never built such a system before. Nevertheless, we hold some degree of familiarity with the technology and a slight understanding of the physiotherapy world.

Chapter 6

Pragmatism

In this part of the report we wish to illustrate how employing pragmatism and especial John Deweys take on pragmatism, may help to bring an applicable perspective on the creative design process, which fits well with the role of software innovation and Essence. In this first chapter we depict some central concepts found in pragmatism. These will later be used to highlight and discuss how we construe these concepts from the perspective of Essence and how they may influence the role of Responder. Later we wish illustrate how employing these concepts in software innovation context, may bring about new and interesting ideas. We wish to show how controlled experimentation and exploration may help developers in a software innovation team to see new and unforeseen potential in technology.

In this chapter we concentrate on some central concept related to John Dewey's perspective on pragmatism. We also describe how pragmatism fits in the world of design by looking at Schön and the situated reflective conversation. To help explain these rather abstract and sometime intangible concepts, we exemplify some of these with the case described Section 5.3.

6.1 Pragmatism & John Dewey

Pragmatism is a philosophy tradition that originated from the United States in the last decades of the 19^{th} century. The philosophy emerged as response to the problems migrants were exposed to, settling the East Coast of America, also known as the historical Wild West [45]. Life in the Wild West was a constant struggle and the migrants had to make many hard choices in order for them to survive and cultivate this harsh and demanding territory in their search of the American Dream. To choose among alternatives, these choices had to somehow be judged and this judgement was based on the practical consequences each choice induced. The knowledge and understanding resulting from these consequences were then used to solve real life practical problems as they occurred in this rogue environment. A philosophy originating from these conditions, the core concept of pragmatism is well captured in the pragmatic epigram that we humans "***learn by doing***" [45].

The pragmatic philosophy is characterised by how it understands human action as a creative act that tackles life's problems head on without having to justify every action used to find a solution [45]. Opposite a rationalist philosophy, pragmatics believe that our reality does not dictate any rational solution to the many problems we face during our lifetime [14]. It means that one cannot prescribe a solution to a given problem, but that a problem is situated to the time we live in and through exploration, experience and experimentation – practice – we have to try and make sense of this world by transforming into a world we can handle and understand [27]. For Dewey this world is a generic term for what we interact with [45]

One of the main contributor to the pragmatic philosophy tradition is John Dewey (1859-1952) [27]. His work covers a lot of topics including logic, ethics, politics and technology and has influenced thinkers such a Donald A. Schön. Dewey's work is also widely discussed in the context of the development of educational theory and his mindset is likewise reflected in Problem-Based-Learning (PBL), a popular format practised at Aalborg University [29]. Through the PBL format, students will be taught about a subject through the experience of problem solving, by involving the students to participate in experiential and active learning.

The time in which Dewey lived, saw many mechanised innovations come to life; power plants, automobiles, airplanes, the telephone, radio, AI and the first robot only to mention a few. Dewey was an active part of this society and his work is shaped by it. The innovation of technological artefacts was a catalyst to his work and his philosophical thoughts are seen as an reaction to the opportunities and problems that occur in a highly technologically society [27].

One of Dewey's points of interests was how technological tools and instruments come to be, how they change the way we humans experience the world, and how they become part of shaping and building our own future. Dewey was of the opinion that these tools and instruments include objects without material forms such as numbers, ideas, theories and even logic [45]. This can also be observed in Dewey's description of the reciprocal connection between science and technology [27]. One of Dewey's core ideas is that the production of ***artefacts*** – both *physical* (technology) and *mental* (theory/ideas) are two instances of the same creative process [45].

Despite the fact that the concept of technology is central to Dewey's philosophy, he did not devoted to a single piece of work to the topic. His critique of technology is instead described throughout most of his work and therefore can be hard to pinpoint. The critique of technology becomes a synonym for his main theory of inquiry in Dewey's later works. The concept of inquiry will be described in the following.

To help our discussion on how Dewey's take on pragmatism fits within a software innovation context, we have selected three core topics which we will describe in details. These topics are however very much interconnected and intertwined as they all stem from Dewey's philosophy of ***inquiry***. Nonetheless we choose to explain these topics individually to try to simplify these rather intangible concepts and exemplify these from our case. We start our description by looking at Dewey's take on the concept *problematic situation*. Thereafter we describe inquiry, and means and ends.

6.1.1 Problematic Situation

In Dewey's perspective of pragmatism, all humans actions are ***situated***. A Deweyan situation is determined by the subjects, the environment, artefacts, social and spatial constructs that are currently present and influence the situation [14]. It means that human actions can only be understood in the context of what situation we find ourselves in. A person or ***inquirer*** who acts upon a situation can therefore be seen as situated to the current context.

A ***problematic situation*** is according to Dewey a situation that is indeterminable and doubtful [14]. A problematic situation will disrupt our current flow of activity and we will stop up and ponder how to resolve this situation. To trigger a problematic situation some kind of error will have to be detected. This error can be defined to be a mismatch between what we expect to happen and what does actually happens [5]. This occurs when the outcome of some action does not match what we anticipate. At this moment we become surprised and through new actions and reasoning we try to resolve this situation.

The detection of error may not necessarily have a negative outcome as the term may suggest. An error is not a mismatch in itself, but rather an indication that a mismatch can currently be observed in our situation. This detection may actually lead us to actions that will solve the problem better than we initially had planned for.

> **Tattletale case:** Let us assume that our little developer team has worked on our tattletale solution for some time. We, the developers, have come to the conclusion that the best technology to recognise movements in the home of Holger, is the Kinect device. We have worked on a module that can recognise a specific shoulder movement in real time. This module is also able to test Holger's movement against how Lise thinks this movement should be carried out. The team has decided to utilise a screen to output feedback, so Holger can see how he should be executing a specific exercise.
>
> We have chosen to build the solution for a touch screen, so the device can also act as the interactive device between Holger and the system. We see the touch screen as the place where Holger can execute commands to the system such as start, stop and replay a program, exercise etc. We believe such a touch device will make it easy for Holger to operate the system and it may even enhance the user experience.
>
> However, we have detected a usability flaw with the system – Holger has problems interacting with the system, while he is actively participating in the exercise program. When he is performing an exercise in his living room, he has to stand at least a meter away from both the screen and the Kinect. It means that when he wants to communicate with the system, he has to walk towards both devices to be able to touch the screen. While he is moving towards the system the Kinect device will not be able to detect Holger's movements as he is out of field range of what the Kinect sensor can observe.

In the case above we have identified a problematic situation. We have discovered an error in this setting. We expected the touch screen to be easy to operate, but this is a mismatch of what actually happens when Holger engage with the system. Here our problem is situated. We operate in situation that is constituted by Holger, us the developers, the Kinect, the touch screen, and Holger's home. The spatial context is found in Holger's living room and the fact that he needs space to perform his exercise program. The problem occurs in this context. We expected that Holger would find it easy to operate a touch screen, but this is not the case in the current context.The space dimensions become a problem when Holger needs to communicate with the system.

As we have no other way of confronting our problems head on, we need to resort to creative thinking and action. This can be seen as a process where we with action will try transform our problematic situation into a situation that is determinable. This process is described in the following.

6.1.2 Inquiry

Inquiry in the world itself means to make an investigation into matter of interest or problem, in a search of knowledge. However, an Deweyan inquiry starts with our problematic situation. A Deweyan inquiry is when we try and transform the problematic situation into something determinable and predictable, by action and mental reasoning [5].

The inquiry process is one of the cornerstone in Dewey's philosophy and can be described as a series of iterative and intertwined steps [14]. 1) Firstly we become aware of the problematic situation, as the outcome of solving a problem does not match our expectations. 2) We try and identify what aspects of the situation have induced the problematic situation. 3) We then consider and conceive how we may resolve these aspects. 4) Lastly we experiment and explore with the actions we should employ to produce artefacts (ideas/technology) that will make the situation determinable.

The process of inquiry is played as a transaction between us and the situation itself. As the problem is situated, the action we reach for to solve a problem, will have to be understood in the particular context. However, the outcome of the inquiry may bring on new surprises and doubts. This mutual relation will continue in an highly iterative manner. While we try and solve a situation with action and reasoning, this process will bring a new situation with anew problems. It means that we ourselves are an active part in constructing a problematic situation, but also play an active role in resolving this new indeterminable situation with inquiry of new actions [5].

Tattletale case: Earlier we identified a problematic situation in our development of the tattletale system. Holger finds it difficult to execute commands to the system, even though we believed that a touch screen was the best platform for interactively communicate with the system. We identified that the spatial dimensions are problematic. Therefore we are determined to investigate if there are other ways to communicate with the system apart from touch.

As other Kinect solutions are built upon the domain of Natural User Interface(NUI) design, we get the idea that we can utilise the Kinect API to recognise simple hand gestures. It means with simple push and shake gestures, we can produce a software module – an artefact – that allows Holger to execute different commands with his hand. However through our exploration we realise that despite the fact the module can help us solve the space dimension problem, we run into another problem. Holger has just had an operation to his arm, which is why he is participating in the rehabilitation program to help mobilise this joint. He finds it extremely difficult to make the command gestures precisely enough for the system to recognise these, without too much pain inflicted.

In the case above the inquiry into our problematic situation resolves in a new problematic situation. Despite the fact we solved the space dimension problem we created a new problematic situation that is situated to our current context. We as developers have been an active part in solving the first problematic situation, but also active in creating a new indeterminable situation.

When we set to intelligently inquire into the situation we aim to discover means that will produce some kind of desired result [22]. It implies that our production of artefacts somehow have to be evaluated in the process. The concept of means and ends will help describe this evaluation.

6.1.3 Means & Ends

To help evaluate or judge the production of solutions to a problematic situation, we need some kind of criteria we base our evaluation upon [45]. Dewey saw criteria as something that would arise within the process of inquiry. This something should be used to determine if the production of artefacts is successful or having failed in solving a problematic situation. Dewey coined this something in the concepts of means and ends and the inseparable relation between these. In the following we give an account of these two concepts and the relation between them.

The concept ***means*** is to be understood as when an activity becomes an adapted human action with a purpose that suits the situation we find ourselves in. One way to help understand the concept means is to look at how to distinguish between materials, tools and means according to Dewey [22]. We will illustrate this point with a little example.

If a builder is looking at building a wooden box, the nails and planks are not strictly speaking *means*. They are the *materials* which the builder utilises to produce the wooden box. Saws and hammers are on the other hand *tools* that have the potential of becoming means. However, these tools only become *means* when they are actually employed in making the wooden box. As means they become active part, within the actual the situation and context the builder finds himself in. So if the builder is not using the saw and hammer in the situation of building the wooden box they are just passive tools, but the moment he picks up the hammer with the purpose of utilising it to hammer a nail into a planks, the hammer becomes a part of the activity of producing the wooden box - a means.

As means help us adapt an activity we need something that give meaning to the activity. If there is no meaning to the action we employ, the action becomes blind and disorderly. The actions we commit ourselves to in a problematic situation have foreseen consequences that give meaning and direct the action. Having an end or aim with an activity is what Dewey calls ***ends*** or ***ends-in-view*** [22]. The concept of ends-in-view means ends that are active and alive and they through interplay with our means not only will be tested but also forged [45]. Where **means** can be seen as intermediates, a series of act with an end, **ends** help us elaborate on what act to be performed and look at the next act in perspective of the context. It will give us a clear direction of the course of action we should take to solve a problem. Hence, means and ends can be seen as inseparable [45]. They coexist and cannot be defined without taking the other into consideration - they are two names of the same reality[22].

The sources of both means and ends can according to Dewey be found in our social experiences [45]. We have to experience and interact with the world for us to understand how to pragmatically solve the problems we face, as we have no choice but to confront these problems head on when they are encountered.

Tattletale case: The concept of means and ends and their inseparable relation can also been seen in the context of producing our tattletale system. Let us start with the materials, tools and means. To produce our gesture recognition module, we need the following materials: The Kinect to detect the hand gestures, a screen to give Holger feedback of the command he has just executed and the most important material of them all – the code itself.

To help produce code a number of tools can be employed. As software developers we have a large set of tools in our toolbox to help us produce our software product. Visual studio is a tool that helps us with code highlighting, intellisense, file management etc with the purpose of managing and producing manageable software. Another tool in an agile context, is code refactoring that has the purpose of bringing more quality and value to the running code.

When these are employed to produce an actual piece of code they become means for what we are currently making - our gesture recognition module. However they are only means when they are brought into the context of what we are producing and thereby become active in our production of our tool. These means is our series of actions we take to reach our end-in-view - to produce a software module that is able to recognise gestures. We aim to build the module with flexibility in mind so it can easily be modified and fitted with the current architecture. These are adapted to the current situation. Visual studio will help us manage the production of artefact; structure our classes, the code itself and highlight potential code mistakes. Refactoring will help us produce a module that is simple, flexible and readable

This concludes our description of Dewey's perspective of pragmatism. As can be extracted from the description of the three topics above, the core features in Dewey's work is a reciprocal nature of science and technology, experience and ideas, the abstract and concrete, purpose and thought, practice and scientific inquiry, action and meaning [45]. This nature can also be found in Schön perspective on design, which we will submerge into in the following.

6.2 Pragmatism & Design

One of the core disciplines in software engineering is design. A developer is faced with this discipline multiple times during a project whether this is during conception of essential components of the system or design of the user interface. This is also the case in innovative software development where is the design activity often occurs highly iterative as the developers are presented with a number uncertainties.

If one is looking in the direction of Donald A Schön, a clear connection between pragmatism and design can be found. Schön was also an American influential thinker. He completed his thesis at Harvard University, which dealt with John Dewey's Theory of Inquiry. This will become apparent as we introduce Shcön's take on the process of designing. Schön has also worked with educational theory and much of his work is influenced by reflective situated learning. His most notable work is depicted in the book "The Reflective Practitioner" and his thoughts of the design process have also influenced the Essence framework [3].

One of the cornerstones in Schön's perspective of the design process is the *reflective conversation* that happens between the designer and the materials that are present in the actual design situation. It is Schön opinion that the design process can be captured in three main steps he calls ***seeing-moving-seeing*** [47]. The designer observes the current design and the materials at hand, then transforms the design to solve a current design problem and then observes the design again. This happens as small iterations where the designer will decide upon a move action which will allow her to let the situation talk back and see the design situation anew. Schön compares this reflective conversation with the situation with the Deweyan inquiry where designers make inquiries into a problematic design situation which they will resolve with action and mental reasoning. They react to the possibilities and requirements that are present in the situation - a situation they have created themselves [46].

When the designer *sees* a configuration of a design she will use her ***appreciative system*** to help judge the quality of the design. She may observe her design and determine that something in the setting feels inappropriate or incongruous and not matching the current context. The judgement is somewhat subjective to the individual and based on the individual's appreciative system. This system is variable and develops and change over time. The system is build upon experiences made both individually or as group of peers and it makes us able to make a judgement about a phenomena without having to disclose the criteria of which this judgement is based upon tacit or explicit knowledge [47].

The *move* action can have ***unintended consequences*** which cannot be foreseen before the actual move is progressed. This can lead to a situation where the designer may see new and alternative ways of looking at the design problem.

It means that the process of design should be seen as a highly chaotic process, that relies on exploration and experimentation to overcome the obstacles and complexity found in the design situation. The designer will draw on one domain to see and judge the design at first and instigate a move action. But through the move action the designer may discover that the move action also produces consequences that affect other domains. This way the designer deals with the complexity of a design situation. The designer does not have to assemble all possible consequences before the move action is instigated, which would be a highly rational and complex task for any designer to mount. Instead will the designer through exploration and experimentation discover consequences of the move action and thereby see new alternatives of the design itself.

Through this learning process the designer will both accumulate knowledge of the actual design domain which can be used for other projects, but also allow the designer to obtain more better insight and understanding of the actual design problem.

Summary

In this chapter we made a short introduction and presented some of the core concepts of Dewey's work on pragmatism. We linked this philosophy to world of design by looking at Schön's take on the reflective designer and presented how prototypes can be a means to explore and experiment with the design space itself. A designer always face a choice of action and this choice can have a profound influence on how we observe and see the design situation anew. In the next chapter we wish to bring this theory into the context of developing innovative software and show how it can be seen from the perspective of the role of the Responder in Essence.

Chapter 7

Responder's Inquiry

As presented in Chapter 3 current development methodologies provide little encouragement for innovation as they strive for quality and efficiency. With their focus on efficiently meeting user requirements, inquiries into seeing new possibilities in the technology are not as such encouraged. Instead inquiries into the technologies are limited to transforming uncertainties related to ***how*** such requirements should be met.

In Chapter 2 it was stated that innovation, in it simplest form, is the successful exploration of novel ideas. In software innovation it is the exploration and maturation of these ideas that lead to new and surprising solutions that offer a high degree of value towards users, customers and society. In an innovative context it is not enough for a developer to make inquiries into ***how*** a solution could be implemented but also ***what*** it can afford and why it matters.

Experimenting with the development of such a solution, presented in Chapter 4, uncovered a lack of activities in Essence for supporting a developer in performing and managing such exploration. In this chapter we argue that such an activity should encapsulate an exploratory effort into the affordances provided by the technology. To help motivate this point and describe this exploration effort, we rely on Dewey's contributions in the field of pragmatism.

In the following section we argue the importance of encapsulating an exploratory effort into the technology to determine *what* it affords. We illustrate this from a pragmatic perspective on Essence.

7.1 A Pragmatic View on Essence

Essence has been developed in response to the challenges of bringing innovation into a software development context. Such challenges involve the balance of facilitating the creative act of generating and maturing ideas with managing the uncertainties to remain productive. Essence facilitates innovation by going beyond the compliance with requirements to continuously evaluate ***what*** we are developing, ***how*** to do it and most importantly ***why*** such a solution is valuable. As such Essence concentrates on the effort of bringing innovation to the project level.

7.1.1 Activities in Essence

Our take on the innovation process in Essence can be seen in Figure 7.1. We believe the entire Essence project can be seen as a problematic situation surrounding the Challenge. When a project is initiated we are quite uncertain about how the outcome of our effort will turn out. This outcome is determined through a series of inquiries made into the uncertainties surrounding the situation.

These inquiries are performed in the different activities on the Essence Views. The outcome of each inquiry will affect our understanding of the problematic situation

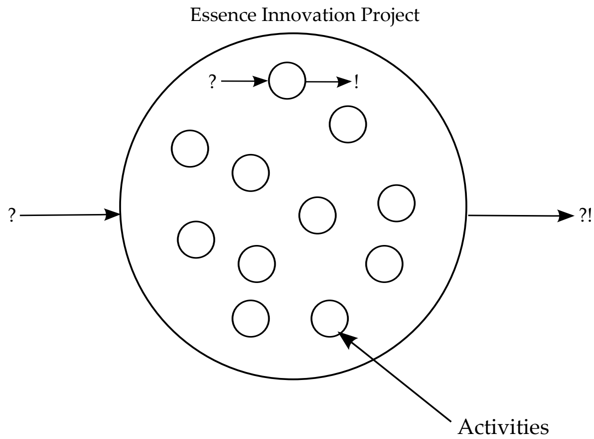

Figure 7.1: Inquiries in Essence

and might, in some cases, even affect the challenge itself. To illustrate this the following case is presented.

Tattletale case: As mentioned in Chapter 5 our development team has been challenged to build the tattletale system. As Responders we quickly propose that one of the simplest solutions to answer the current challenge would be to create a web page, or mobile application, where Holger can enter his exercise data. However as we work with use scenarios on the Paradigm View in collaboration with Lise, we realise such a solution would not work well. According to Lise's experiences, patients have a tendency to lie or simply forget to report their exercises. In Lise's view a tattletale system is something that actively sits on the sideline and monitor Holger performing his exercises.

During a brainstorm session we decide to take a look at the WiiMote. We believe this is an interesting technology, and we presume that the WiiMote is able to recognise Holger's movement as this is often seen video gaming. We agree to further engage in this technology and quickly locate an open-source WiiMote PC API and employ the activity of prototyping.

With this prototype, the location of Holger's hand can be tracked and the prototype is presented to Lise. Lise sees the possibilities but argues that the shoulder exercises involve movements of both arms. We do not see this as a problem, as we can just add a second WiiMote. Lise however argues that the positions of the shoulders and general body symmetry are just as important to monitor as the position of the arms.

With this knowledge we move back to the Product View to make further inquiry into the technical solution. During this inquiry we identify the Kinect sensor which seems like the perfect option. It can not only track the position of the hands and shoulders but also 16 other points on the body. We agree and quickly throw together a prototype which is shown to Lise. This Kinect sensor is exactly how she sees we can monitor Holger.

Now we want to show Lise how the Kinect is able to perform beyond her current needs. By standing on a single leg and moving the other up to an angle of 90 degrees we demonstrate how the Kinect can be used to monitor other parts of the body. Seeing this prototypes makes Lise wonder. *Is it possible to aggregate data over time and use it to evaluate the speed and quality of progress?*

We respond that we might even be able to tell whenever or not the exercise was too easy or too hard.

Lise now sees a new value proposition. She might use such a system to continuously evaluate and adapt the program to the needs of the individual patient. This could mean better rehabilitation programs and shorter rehabilitation time. This is where a paradigm shifts happen. From just focusing on Holger, a change is made to also facilitating Lise in creating and improving rehabilitation programs.

What has happened throughout this process is that a series of inquiries was made into the problematic situation surrounding this original challenge. Our first inquiries were initiated because we experienced a lack of knowledge with developing such a tattletale solution. This combined with little knowledge about the use context, meant that our challenge in our tattletale was encompassed with uncertainties. We might have had a pretty good idea of where to start, but we are quite unsure where the final solution will end.

So as such can our response to the challenge be seen as an inquiry into answering this challenge. To overcome the uncertainties we engaged in different activities on the Essence Views. Through these inquiries involving interaction, exploration and experimentation with both the use and technological domain, we have gained valuable information which helped resolve the uncertainties. However, in this process we have been active in creating entirely new problematic situations. These have led us to new technology and new value propositions.

7.1.2 Means & Ends in Essence

According to Dewey some kind of criteria is needed to help evaluate whenever a solution to a problematic situation is successful. Dewey sees criteria as something that can arise within the process of inquiry. He calls this something means and ends. While the means are the action and tools we apply to resolve the situation the *end* or *end-in-view* is what will give us a clear direction of the course of action we should take to solve it. Our perspective on this seen from the four views in Essence can be seen in Figure 7.2.

In Essence an end-in-view can be seen as the Project Vision. The Project Vision captures the idea, or set of ideas, that currently outline the ambition of the project. A vision is not static but is continuously matured and changed in response to the activities performed on the Essence views. Essence's reliance on Visions rather than requirements, indicates that there is a stronger bidirectional relationship between means and ends than in other current development methodologies. In these methodologies it is assumed the use context and user needs are known. In other words we expect our customers to define an end-in-view for us either upfront or through a series of iterations. Selecting the tools, materials, and actions to meet these, becomes a rational process based on reasoning. As such the design options we choose between are at least to some degree known.

In Essence a Responder's responsibilities goes beyond implementation. She must be able to reflect on user needs based on the exploration into the technology. With an unknown use domain, and user needs, the design space become non-trivial. Through the exploration of this space, the Responder gains knowledge and ideas which will affect the end-in-view.

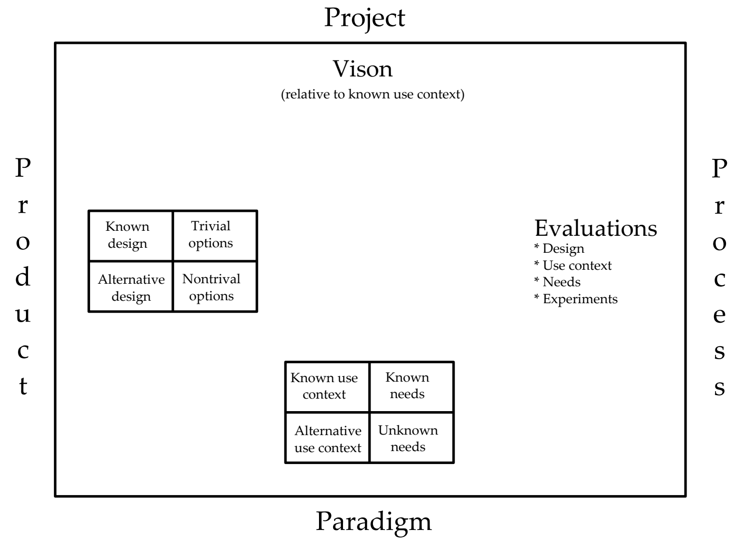

Figure 7.2: Means & Ends in Essence.

Tattletale case: In the last case the scope of our tattletale project was extended to cover two challenges. To build a tattletale system to monitor Holger in his home and to aggregate and present historical data of Holger's exercises, which allow Lise to better tailor the rehabilitation program to his abilities. When Lise leaves after the Kinect demonstration we decide on an initial Project Vision. The Vision is to use the Kinect's skeleton tracking feature to monitor Holger and collect data for Lise.

At the moment much doubt still surround the Challenge and Vision. The team's Challenger cannot provide clear answers to questions regarding the use context and user needs. These questions mainly surround the kind of data Lise is interested in. This leaves an open and large design space and we decide to make inquiries into the technology – materials and means – to identify possible answers. During a small research session we discover a commercial for a dancing game. It utilises the Kinect sensor to recognise dance movements and grand points for their correctness. We argue that combining the Kinect's skeleton tracking feature, with some kind of movement recognition algorithm, we can aggregate and rate Holger's performance. Once a day these ratings should be send to Lise for presentation and evaluation.

We agree that such an approach seem reasonable. The tools and materials identified now include the Kinect and an internet connection. Actions to implement the vision includes the implementation of a movement recognition mechanism, a presentation module for Lise and some server logic for handling the communication. Implementing such a movement recognition mechanism make us a bit nervous. This might be quite complicated. To answer this question we decide to experiment with the production of an artifact. A third party gesture recognition API is quickly located and we realise that this can be used in a simple prototype. It recognises a simple movement of the arm. A counter in the upper left corner shows the percentage of movements which is done correctly.

The prototype is evaluated by the rest of the development team but we are not impressed. Yes a counter is there, but its behaviour is strange. It is not really clear to us when and how to determine whenever we are successful or not. We argue that some kind of feedback is needed to tell Holger when to start and stop. This feedback can be provided by combining a video image of Holger with a data overlay. At this point we start to reflect. Such visual feedback could not just be used to ensure that the system operates reliably, but also provide Holger with visual instruction in how to do an exercise, and in case his score is low, what went wrong.

This is an example where the act of constructing and evaluating artifacts – bringing the tools and materials into action – can lead to reflection regarding the use context. To facilitate this reflection we move to the Paradigm View and adopt the Child role. Here we are free to brainstorm and experiment with how this new realisation could manifest itself in value for Lise and Holger. It is determined that the value lies in providing Holger with continuous feedback and instruction, which assist him in independently ensuring that the exercises are performed correctly. In this way Lise can save time which she can use on other patients.

The process presented above illustrate the strong bidirectional dependency between means and ends in Essence. In contrast to traditional and agile development methods, the responsibilities for a Essence developer goes beyond design and implementation. We must ***reflect over user needs***. This leaves a larger design space and, in our view, increases the chance that a novel solution might arise. The reflection do arise when bringing the tools and materials into action. It is this reflection that influences and changes the end or vision, of the project.

7.2 Responder's Inquiry & Essence

The role of Responder in Essence is assigned to the software developers in the team. The Responder is the heroine, who through software craftsmanship and technical expertise can realise a solution. But she is also the technologist, the engineer, the pragmatist who has the technical for- and insight to seek opportunities within the technological design space.

The sofware developer will in any software development project rely on her skills and the activities found in her professional toolbox to beget a software construction that fulfil the customer's expectations. However, in an innovation context there are certain skills that are more sought after and valuable than others. In Essence it is the responsibility of the Responder to choose an implementation that radiates flexibility and expandability, so the produced technological artefact affords new and alternative ideas. This can be related to the concept of ***technological affordance***.

The concept of technological affordance derives from Gibson 's original definition of the concept affordance [26]. Gibson defines the term as all possible actions an object offers to an actor. A hammer affords gripping and a door handle affords pushing down. Norman brings the concept into a HCI context, where he defines perceived affordance to be all possible actions an actor can perceive [37]. This could the click action a website button affords.

Technological affordance is highly associated with the Product View and the Responder role in Essence. Aaen [3] defines the concept as a choice of implementation alternatives that allow the team to add new and alternative features to the solution as the vision matures and evolves. The implemented technology ***affords*** new ideas.

To choose such an implementation strategy requires technological insight and not the least technological foresight.

7.2.1 The Pragmatic Responder

As mentioned in Chapter 3 the role of the Child is to explore novel ideas and seek alternatives. This role is shared mutually between the customer and the developers of the team. We see the developer in an Essence team as a double agent who is responsible for both exploring and discovering the use context, but also responsible for locating the technological means which will help realise a *novel solution*.

As a Responder it is our task to locate and explore these technological means. It means we as Responders have to actively engage in cultivating our foresightedness and curiosity with the potential in technology. Often this requires activities that allows the Responder to explore and experiment with a given technology. This is especially the case when the knowledge into the technology is limited.

We can see this exploration – a Responder's inquiry into the technical domain, very much like that of Deweyan inquiry into a problematic situation. A Responder's inquiry can easily be compared to the pragmatic view on how to deal with doubt. A Responder's doubt is bound to the current context and situation we find ourselves in. When we make an inquiry we search for knowledge that help resolve this doubt. This search is concentrated on the technology. We have to explore the technological means and tool that will help transform this doubt. Doubt can occur if we are unsure if an interesting feature is realisable with the technology at hand. Or we may wish to explore the technology further to see if any interesting use cases could be discovered e.g. cultivate our foresightedness and curiosity.

Our doubt is situated e.g. it is bounded by the context and situation we find ourselves in. However for us to fully understand the implication of an inquiry, we have to experience it through practice and seek the materials and tools that will help transform our doubt. As software developers we normally become pragmatics in this instance. We sit down with our hardware and software components and utilise these to construct and produce new technological artefacts, which will hopefully help transform our challenge into a realisable solution.

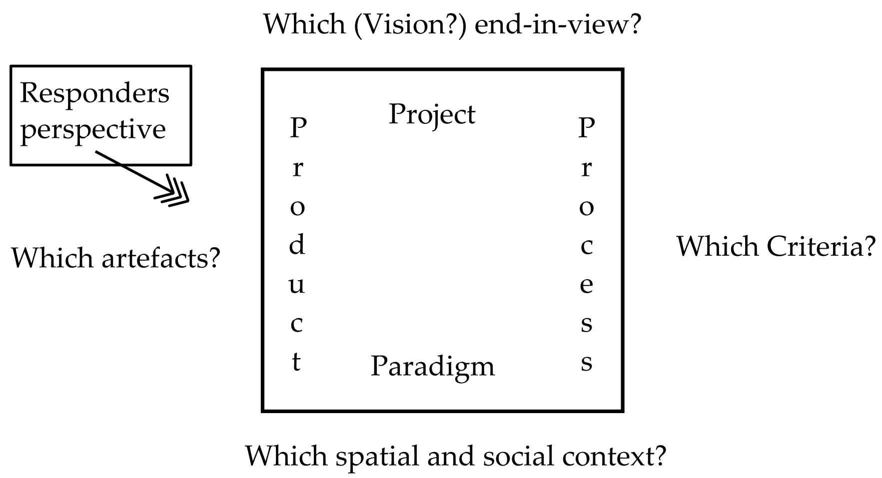

Figure 7.3: Responder's Inquiry in Essence

To help guide and inspire the Responder in an inquiry into the technological domain, she may seek information related to the four View structures in Essence. Our take on this can be observed in Figure 7.3. On the Paradigm View she may look for information describing the social and spatial context of our use domain. On the Product View she may inspect the technological artefacts we have produced so far, but also take stock of the artefacts that are being employed in the construction of the solution. Do any of these demonstrate interesting potentials that may lead to undiscovered use scenarios? On the Project View she may search the project Vision for a point of direction in her efforts. An end-in-view to guide and adapt her actions to the current situation. On the Process View she may seek inspiration for some kind of criteria to help evaluate her efforts. However, none of the structures or activities in Essence really support a Responder to make such an inquiry.

Tattletale case: In the tattletale case our development team proposed that the Kinect is able of more than just monitoring Holger and informing Lise on Holger's progress. Through movement recognition and visual feedback, Holger can be assisted in *how* to perform the exercises, and, if he is doing them incorrectly, *why* an exercise may be wrong.

As Responders we are now assigned to explore how such feedback could be implemented. This is an example where we as Responder act as double agent. Not only are we responsible for understanding the affordances and limitations in the technology, but also for understanding the use context. During this inquiry we study the Kinect's specifications. We realise that the Kinect has some very unique video and audio features making it very useful for teleconferencing and in-game voice chats. Here we have shown technological foresight by choosing technology with a broad set of affordances.

Being the double agent and believing in the two Essence values of *affordance over solution* and *reflection over requirements* we decide to explore and reflect on these affordances offered by the Kinect. By utilising the video camera and microphone on the Kinect, a video link between Holger and Lise could be established. Both video camera and microphone are designed to support interaction with games, so we assume they will work great even in challenging conditions with limited light and background noise.

However we are not sure if there is value in such a video link so we ask the team's Challenger to join us on the Paradigm View. Here we adapt the Child role and engage in a brainstorm session. We determine that such a video link could help Lise instruct Holger in performing his exercises. In this way Holger would not have to travel to the clinic thereby saving time and resources on transportation. The proposal is presented to the Challenger and we agree that it seems like a really great idea.

We decide that our vision has become complicated enough and we move to the Product View. An inquiry is made to identify the tools and materials needed to successfully reach this vision. The result of this inquiry is seen on Figure 7.4

Each of these technologies introduce their own uncertainties into the project. How do they work? Can they do what we expect them to? Do they afford anything else? To answer these questions inquiries must be made. We split up the team and each Responder is assigned an area that must be researched.

In case above it quickly becomes evident that our exploration effort becomes resource demanding. Every time an uncertainty is transformed, new ones arise. We

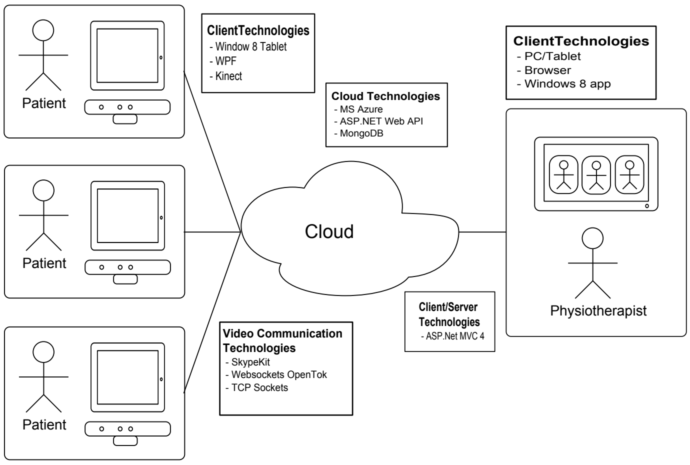

Figure 7.4: Example of Architectural Diagram for the Tattletale System

experience how this slowly moves our focus away from what is important, the Challenge and Vision, to instead focus on many of the details which might or might not, eventually become a part of the project. In other words, the exploratory effort has run out of control.

7.2.2 Exploration into an uncertain technological terrain

As described earlier in Chapter 4 our own practical exploration and experimentation with the technology on the 9th project can sometimes be frustrating and hard to control. Through our own experience as Responders we found the responsibility of actively seeking and engaging in an inquiry into the technological domain can be difficult to structure as an activity. In our own experiences it is not difficult to produce new and interesting ideas. What is hard is to explore, mature and transform them into practical solutions.

When we explore potential with the technology we face a vast amount of uncertainties in our exploration. What aspects of the technology is worth exploring and how do we structure and contain, moderate and evaluate such effort.

We wish to propose an activity that will guide and structure a Responder in a Responder's inquiry. Our ambition is make a Responder's inquiry a purposeful effort, which we wish to frame in an manageable activity. The aim is to divert and deflect uncertainties into a framed sandboxed activity, where we can explore and experiment with the technology in an isolated effort. The gained knowledge can then form a background from which we later can make decisions if we want to pursue this inquiry and let it affect and mature the overall project.

We call this activity a ***Spike***. Why we choose this name will be elaborated in the next chapter, along with the objective and a detailed description of the activity itself. Here we wish to look at this as an gestalt activity as depicted in Figure 7.5 and elaborate on what may *trigger* such an activity and what the *outcome* may be.

Triggers

We see a Spike as an activity that has to be triggered. The trigger will stir our curiosity and lead to an inquiry into the technological domain. We believe a Spike

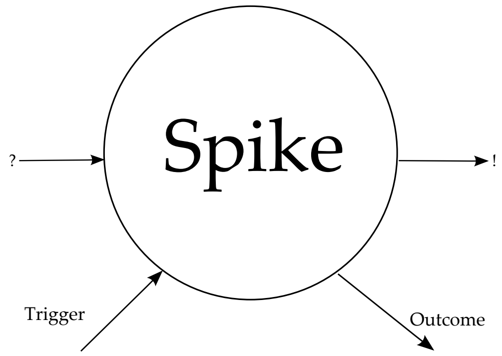

Figure 7.5: Spike as an isolated activity in Essence

can be triggered for two reasons:

- A Spike may occur as a ***natural halt*** in our current development. It implies we cease for a the moment as something in the technology has sparked our curiosity. We may have discovered some new interesting features with the technology we are working with. Or the outcome of an old Spike might have indicated that an undiscovered feature in the technology is worth exploring.
- A Spike may occur as a ***synthetic halt*** in our current development. We may observe our current solution is rather dull and we wish to try something new. This may spark an inquiry into technology to seek new and alternative ideas.

Outcome

The sandboxed nature of a Spike activity frames the inquiry onto a problematic situation in a way that both scope, time and effort is limited and disjoint from the main project itself. As we see a Spike as an activity that will resolve a doubt into the technological domain, we will at the end be able to evaluate if the effort is something we wish pursue and bring into larger perspective or if we wish to leave our effort behind. No matter what we need to reflect on what the Spike has taught us.

There may be cases where we come to the conclusion that we see no potential in technology which can help mature the overall vision of the project. More often however we expect the inquires to locate possible and interesting features that could be used on our main project. The outcome or reflection phase is about taking our newly found knowledge and apply it to the entire project.

The scope and type of learning outcome is unique for every inquiry and will as such affect the four Views in Essence in different ways. An inquiry revealing an effective algorithm for gesture recognition might only affect the Product View, while other inquiries might reveal new technologies that greatly affects the vision of a project. An initial starting point could be to reflect on how the social and spatial context presented on the Paradigm View is affected by the Spike. This is where new use scenarios or prototypes are created based on the opportunities seen in the technology. If such opportunities are found, it must be evaluated if and when they are important enough to affect the vision of the project. It might well be determined that such change in vision requires further research or knowledge structured and evaluated on the Project and Process Views.

As a result we expect to observe a ripple effect where the outcome of one Spike leads to unexpected changes calling for further inquires performed by Responders, Children or even Challengers. A chain reaction of Spikes might even emerge.

Summary

In this chapter we presented how pragmatism fits within a software innovation context and Essence. We argued that neither the traditional nor the agile paradigm addresses a Responder's inquiry into the technology. However we believe that in an innovation context facilitating such an inquiry is vital, as it may afford new and alternative technological features that affect the overall vision of the project and bring about novel solutions.

We linked the Responder's activities in Essence and related these to Deweyan inquiry. We illustrated that no structure or activity exists in Essence that aids and direct a Responder's inquiry into the technological domain. We also argued that such activity is needed to help frame such effort. We proposed how such an activity – a Spike – can be observed as an gestal activity. We depict what might trigger such a Spike and what the outcome may be and how it should be handled in an Essence project. In the next chapter we describe our objective with a Spike and illustrate our proposed structure of this activity.

Chapter 8

The Spike Activity

The preceding chapter presented how pragmatism fit within the context of software innovation and Essence. We argued that inquires into the technology are essential in nurturing new and novel ideas that can lead to the creation of highly valuable solutions. The four Essence values support the concept of a Responder's inquiry but Essence do not provide any activities for managing and structuring such inquiries. As experienced throughout our 9th semester such inquiries run the risk of becoming resource demanding, as our exploratory effort reveals new uncertainties and opportunities.

Our aim with our contribution is to make innovation a part of the developers everyday activities. To do this we must transform the theoretical concept of the Responders inquiry into something that a developer can understand and apply. This chapter presents our proposal for a generic activity that allows the developer to practically explore the technology in search of unforeseen opportunities while assisting in managing the risks involved in doing such. To help us frame our Spike activity we wish to present our overall objective. These are summarised as follows:

- We wish to assist the Responders in an Essence team in uncovering undiscovered potential in the technology they are surrounded by.
- We wish to make the Responder cease for a moment and look at the technology at hand and reflection upon this. item We wish to make the choice of technology a problem instead of part of the solution for a short period of time.
- We believe that the Responder can achieve this by exploring the potentials in the technology we have not addressed in the current solution and seek what the technological affords.

For the Spike activity to be applied in a practical innovation context, we wish to make this activity as simple and generic as possible. Our aim is to create a sandbox where Responders can experiment with the technology without utilising too many resources and without directly affecting the main project, but also a place that allows for exploration.

8.1 Inspiration

As introduced above our objectives with our Spike activity essentially serves two purposes. Firstly, to facilitate a Responder's exploration into the technology in search of new and exciting value propositions. Secondly, through some degree of control and isolation to help manage the uncertainties introduced by, and resources spent on, such an effort. In the end, a Spike must be an activity which a development team can afford to apply in their daily work. Inspiration for creating this activity can be drawn from the real world.

Some companies do apply special activities encouraging innovation through the experimentation and exploration into technologies. One of these is Google. Here the company encourages employees to use 20 % of their time on personal projects that could benefit the company. This means that if you got a good idea you are allowed to run with it [54]. To develop such an idea into a product, Google relies on the engineer to convince colleges of its qualities and gather in grouplets. A grouplet has almost no budget nor any decision making authority but they have engineers who are committed to an idea and who are willing to work to convince the rest of the organisation of its value [54]. Around 50

Google is not the first company to promote such activities. Lockhead Martins Advanced Research Department, informally known as the Skunk Works [50], has since the 1940s been developing some of the most innovative aircraft designs including the B-2 and SR-71 spy plane and F-117 stealth fighter. The Skunk Works claims to success is partly given to its founder Kelly Johnsons unconventional organisational approach. He broke the rules, challenged the current bureaucratic systems that stifled innovation and hindered progress. Later the term Skunk Works has been adopted in business and engineering to denote a project developed by a small and loosely structured group of people who research and develop a project primarily for the sake of radical innovation.

What we aim for with our activity is this kind of playground where the developers can play with the technology to understand what it can do and what it offer to us. We however want it on a much smaller scale where it can be embedded and applied in their everyday work. To do this we let us inspire from a tool in the agile development methodology eXtreme Programming (XP).

The tool in XP is used for risk-management by structuring a time and resource bound research effort in search for knowledge. This tool, called a spike, provides the foundation for our own proposed activity.

8.1.1 eXtreme Programming & spikes

With our proposed activity we aim to bring innovation into the daily activities performed by the software developers. As seen in Chapter 3 working with innovation introduces much uncertainty into a project. A main challenge experienced by us is related to the management of these. We believe what is needed is an activity that allows the developers to come up with new and alternative value propositions through the exploration and experimentation with the technology while shielding this exploration from the rest of the project.

The name for this activity – ***a Spike*** [1] – is adopted the tool found in XP. In XP a spike is a tool used to drive out risk and uncertainty in a project [31]. A spike is an exploratory effort with the goal of acquiring the knowledge that allows the team to deal with these risks and uncertainties [6]. It differs from a user story, known from agile development methods [8], by delivering knowledge rather than working code.

The literature applies spikes differently but they seem to converge on the transformation of two kinds of uncertainties. Functionality spikes are applied when there are considerable uncertainties involving how users interact with the system. Technology spikes [31], or conceptual spikes [6], are used to perform research into the technical domain. This insight might be needed to estimate the complexity of a user story – planning spike – or to compare different technical solutions. With some

[1] When capitalised we refer to our proposed activity Spike. Otherwise, the word spike refers to concept found in XP

adaptation we believe this tool can be adapted to bring the notion of Responder's inquiry into a software innovation development context.

The spike, known from XP, must be reshaped to fit this context. The XP spike is applied as a risk management tool. Exploration into the design space is, in our view, performed with the goal of determining *how* a solution should be designed, *how* the user would like to interact with the user interface or *how* many resources must be allocated to implement a user-story.

We want to apply the spike in exploring the technology in search for new and unforeseen opportunities. Through this exploration into an open and largely unknown design space we expect the developer to reflect on ***what*** it affords and just as importantly ***why*** exactly these affordances matters to us.

8.2 Activity Overview

In this section we present an overview of the Spike activity. This is how we picture a Spike activity in Essence to take form. Our objective has been to keep this activity as generic and simple as possible, so it becomes a means rather than an obstacle in our exploration effort.

This activity is inspired and modelled after Schön's perspective on the design process. As mentioned in Section 6.2 Schön captures this process in the three steps, he calls *seeing-moving-seeing* Figure 8.1. In this process the designer utilise her appreciative system to observe and judge the current configuration. Based on this judgement she frames her move, decides on a move action that will help her transform the configuration and then she observes the configuration again. The move actions may bring about unintended consequences which will bring new and alternatives ways of constructing the configuration.

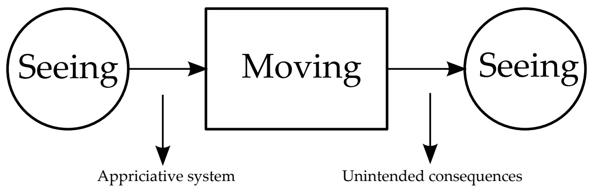

Figure 8.1: Schön's seeing-moving-seeing

We believe Schön's perspective on the design process is interesting in an innovation context. Schön acknowledges that we cannot predict all consequences of our actions beforehand. Such understanding must be gained through exploration — inquiry into — the materials at hand. In an innovative context we strive for the unintended consequences. This is where we come up with new solutions that neither we nor the customer expected. This is where innovation is unfolded.

We also believe there is a logical link between Schöns design perspective and our challenge to manage the Responders inquiry into the technology. In the first step the developer recognises a problem related to the technology. It become situated and an initial understanding of how it must be resolved – our end-in-view – is constructed. In the second phase inquiries are made into the technology to where we utilise our means – the technological tools – to transform our challenge into an artefact that will resolve our problematic situation. This transformation is all about experimentation – trial and error – which pragmatists believe is necessary to confront problems [45].

In our experimentation we need to evaluate if our construction is successful or have failed in resolving our problem.

Our proposed Spike activity draws inspiration from these two views on problem solving. The activity is constructed around Schön's seeing-moving-seeing steps. Our activity also consists of three phases that can be observed in Figure 8.2. These phases are performed as an isolated – or sandboxed – effort. It is ***triggered*** by a natural or synthetic halt and its ***outcome*** must be reflected into the larger development process. The three phases are:

Spike Challenge: by observing and framing our problem in the Spike Challenge, we aim to define and bound our explorational effort.

Spike Exploration: this where we explore our challenge and experiment with the technology at hand.

Spike Evaluation: is where we evaluate on our success.

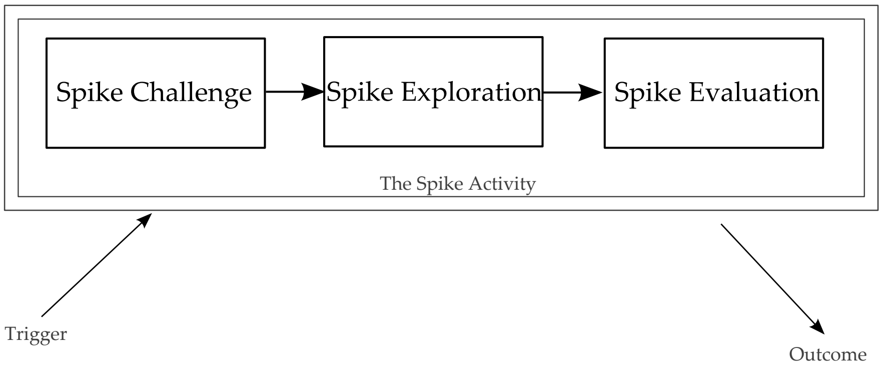

Figure 8.2: Overview of the Spike structure

In the following sections these three phases will be elaborated on.

8.3 Spike Challenge

We see the Spike Challenge phase as the place where we frame our problem. Inspired by the Toulmin Structure (see Chapter 3) we wish to frame such problem into a challenge. We can relate this to framing a Deweyan problematic situation. A challenge is where we situate our technological problem into our current context, and works as a concrete manifestation of our problem setting. We believe that the character of a challenge can help direct our experimentation. In this sense the challenge also become our end-in-view. The challenge brings meaning to the activities later performed in our Spike Exploration.

However, we also believe the challenge must be open for interpretation and not state all the possible consequences of our experiment. We see the challenge as problem statement and not as a possible solution to our problem. This is consistent with Schön's take on the reflective conversation. If we beforehand formulate our problems in terms of all the consequences our experimentation may induce, the complexity of solving the problem may be overwhelming and not very explorative. Therefore we choose to see the problem setting as a challenge driving exploration and discovery.

The tasks that we believe must be carried out in this phase are:

Set a Spike Challenge: The challenge act as an open problem statement in the sense that it help scope our exploration effort and manage the complexity involved in the Spike experiment, but also driven by discovery.

Set boundaries for time and effort: These boundaries will help scope our effort and allocate the resources we wish to invest in this effort.

To exemplify our intention with the Spike Challenge, we will go back to our tattle case.

Tattletale case: Let us assume our team has worked with the Kinect for some time and utilised the skeleton tracking for our recognition module. To even commence on a Spike, it has to be triggered somehow. We know that some of Holger's exercises involve the use of training-equipment in form of dumbbells. These have different colours that indicate the weight of the dumbbells. This triggers our curiosity. Would it be possible to track the dumbbells as well as Holger? If we can tell the weight of the dumbbells, Lise may have better premise to monitor Holger's progress.

The team decides to initialise a Spike and defines a Spike Challenge. We state this in the following problem statement: ***Is it possible to utilise the Kinect to detect the dumbbells Holger is using in his exercise program?*** Through negotiation with the Challenger of the team, we decide to assign one Responder to investigate this challenge. The Responder has two days to complete the Spike.

In this Spike Challenge we situate the problem to the current context. It involves Holger, the Kinect and Holger's exercise program. We frame this situation in a problem statement, and set the boundaries for how much time and effort they wish to allocate for this Spike.

8.4 Spike Exploration

This phase is where we make inquires into the technology and the affordances it provide. Using the challenge we direct our actions to transform this into knowledge that can assist in answering it. We believe the primary form of action to be applied in this phase will be prototyping. Here we experiment and explore with technological solution space which manifest itself in artefacts. The artefacts we construct could be a software module, a software architecture or an algorithm.

In the construction of the prototypes – artefacts – we bring into action the tools and materials we as software engineers have at hand. Such could include software modules both of external and internal origins, promising technologies or even software engineering practices.

We see this exploration into the design space as an highly iterative activity. As both Schön and Dewey point out, inquires or move experiments, are seldom a single and sequential activity. They are comprised of many small sub-inquires, or move experiments, performed iteratively. Each of these iterations involve a statement of intention, an inquiry into its resolution, and evaluation of the results. These small iterations lead to new problems as we see our artefact anew when we experiment and explore with the materials and tools.

This is where we believe our Spike differs from an XP spike. Here we are not risk driven as we forth-come this exploration and the consequences it leads to. As Scön points out, our move actions may have unintended consequences and this is where we believe the A-ha moment might arise - and we become creative and innovative in our thinking. Let us exemplify our Spike Exploration phase with our tattletale case.

> **Tattletale case:** We – the one Responder – decide to explore this challenge through a simple isolated code prototype. We know that all of Holger's dumbbells are clearly coloured and this should make them easily detectable by the camera. We create a simple algorithm able to locate an object with the same colour as the dumbbell. We transform this to a code prototype, which scans through the image data from the video camera. Surprisingly, we realise that the task was easier than anticipated and the prototype clearly detects the presence of a red dumbbell.
>
> With nearly one day left we wonder if the prototype can be extended to determine in which hand Holger holds the dumbbell. We achieve this by using the skeleton tracking feature to obtain the position of each hand. By comparing these with the location of the dumbbell we can determine in which it is held. This further make us wonder. *Can we detect two dumbbells of different types?* This is a subject for further exploration but our two days are nearly over, so we must move on to evaluate our efforts. The sandboxed nature of the Spike cease this Spike.

In our Spike Exploration phase we become pragmatics. Through our experimentation and construction of artefacts we learn new things and gain knowledge of the problem we set in our Spike Challenge. When we construct our artefacts we utilise the materials and tools we have at hand. This includes the Kinect and the built-in video camera and Holger's dumbbells. In our exploration effort some unintended consequence arises – our A-ha moments – where we see the problem anew.

8.5 Spike Evaluation

As mentioned above we expect our Spike Exploration phase to occur in many small iterations all involving the construction of intentions, experiments and evaluations. In this process many unintended consequences might arise that shape the exploratory effort into unexpected directions. In the Spike Challenge phase we drew the parameters for how much time and effort should be put into our Spike Exploration. Because this nature of our Spike activity our exploration will come to a forced halt and this is when we hit the Spike Evaluation phase.

As our Spike Exploration has come to a halt, we have to evaluate what we have learned in our effort. In the Spike Evaluation phase we have to determine if the learning outcome of our Spike Exploration has lead to new knowledge that might affect ***what*** we are doing in the project and ***why*** we are doing it. It is here that our responsibilities as Responders differ from traditional developers. As double agents we must not only explore the technology in search for knowledge regarding technical solutions to problems. We must ***reflect*** on how this knowledge could affect the use context in a broader sense. While the Spike Exploration might have brought up one or more A-ha moments, or novel ideas, resulting from intended or unintended this is where the Responder must explore how they fit in a broader context. As a result we see that the Responder must answer two questions in this

Spike Evaluation phase:

Answer the Spike Challenge is it possible to use this technology and is it feasible to our current context?

Reflection If yes – can or should our newly acquired knowledge affect the vision of the broader project?

Having to answer these questions the Responder must answer the team if this line of inquiry is worth exploring further or not. This is inspired by how an engineer in Google has to sell an idea to gather a grouplet.

All three phases are isolated in the sense that our exploratory effort, and evaluation, are performed without introducing side effects directly into the project. If the Responder can answer yes to the questions above it is time to bring this knowledge out of isolation and nurture it into the project. This is strictly not a part of the Spike itself but happens in activities succeeding it - what we call the outcome. Let us exemplify the Spike Evaluation phase with our tattletale case.

Tattletale case: Nearing our deadline of the Spike we have to evaluate our Spike effort. At the morning standup meeting the next day we have to present the result and evaluation of our effort. We determine that the exploratory effort was quite successful and we see a potential in this technology. It was easier to detect the dumbbell than expected and we believe that this technology is feasible to bring into the context of our project. We tell the rest of the team why we believe this could be interesting in a broader context:

- Holger does not have to tell the system what kind of equipment he uses.
- Maybe the system could determine how easily Holger performs an exercise. If it is easily performed, it could suggest a heavier dumbbell, while signs of tiredness could result in suggestions for lighter dumbbells. Maybe other equipment could be detected?

However our evaluation also reveals that some challenges exist:

- Dumbbells might have the same colour as other objects like shirts.
- The implementation is resource demanding. Some efficient algorithm will need to be located.

After we presented our Spike Evaluation the team decides to take our newly found knowledge into the main project. But the team also agrees that more knowledge about the use context must be investigated to understand exactly how this fits in with our main project.

To explore help this Lise is called in to participate in a session at the Paradigm View. We see the potential in what we learned in our Spike. Lise is however not that impressed that we can only detect dumbbells. She points out that Holger uses other equipment, including exercise bands and an exercise ball. She would prefer the system to work reliably with all of these. We become dubious. Such an exercise ball could potentially wreak havoc with the skeleton recognition, which we use to detect Holger's movement.

But in collaboration with Lise, we also produce some new ideas. *Is it possible to recognise different exercise equipment and can we even detect Holger's movements* *from the exercise equipment he is currently using?*

In collaboration with the Challenger we come to the conclusion that this direction could add value to the overall Project Vision. However we decide to investigate this further before changing direction. We initiate a new Spike activity that has to explore the possibilities of using skeleton tracking to detect e.g. if Holger sits on the exercise ball.

In this Spike Evaluation we have evaluated our exploration effort and identified some of the potential in using a given technology. We have reflected upon our newly gained knowledge and suggested ways of bringing this into the broader context. However as our Spike activity is an isolated effort we could just as easily have decided to leave our effort behind without having it introduce side effects into the main project. The code prototype can easily be scrapped and our visions for how this could be interesting in a broader context can be cast aside.

Summary

In this chapter we have proposed an activity - the Spike that hopeful will help structure and manage a Responder's inquire into the technical domain. The Spike is inspired by Schön's model of the design process – seeing-moving-seeing. It is structured around three phases we call: Spike Challenge, Spike Exploration and Spike Evaluation. In the Spike Challenge phase we direct and set the boundaries for our exploration. In the Spike Exploration phase we experiment with the technology at hand. In the Spike Evaluation phase we reflect and evaluation our newly gained knowledge.

Inspired by spikes in XP and Skunk Works we have constructed an activity that both facilitate exploration into the technological domain but also helps to sandbox and manage such effort. In the next part we wish to bring the Spike activity into practice through our own experiments.

Part III

Experimenting

Chapter 9

First Spike

In the preceding chapter we presented our proposal for an activity – the Spike – that can assist the Responder in exploring the technology. In this part we apply the Spike in the development of PhysioSphere to help evaluate its qualities and mature it. A total of two Spike-experiments were made. The first Spike was conducted early in our research effort meaning that knowledge gained through this has been reflected back into its construction presented earlier. The second Spike experiment was conducted just before our project deadline.

9.1 Preliminaries

As mentioned in Chapter 5 the philosophy of pragmatism does not only provide the link that conceptual bind our problem and proposed solution together. It also describes our approach to conducting research. Experimentation with the Spikes has been important in developing and shaping its structure. Such experimentation has been both informal and formal. Informal experimentation has involved the creation of the cases presented throughout this report. The formal experimentation involves the two Spike experiments. The first Spike was initiated by a synthetic halt. While reflecting over its outcome, doubt arose which we used as trigger for the second Spike. An overview of this process is provided below:

Locating a Trigger Through an exploratory effort into the artefacts, technologies and affordances used in constructing PhysioSphere we realised that not much effort had been placed into usability and user experience. We decided to trigger the Spike by a creative obstruction forcing us to investigate what the Kinect affords within this area.

First Spike The goal became to explore how the unique abilities of the Kinect can be utilised to offer better user interaction in PhysioSphere. The challenge was constrained to focus on exploring the Kinect's audio and voice command features.

Second Spike The second Spike was triggered by doubts raised throughout the first Spike. This doubt concerns how PhysioSphere provides Holger with feedback during exercises. Our assumption, through both experience and research, was that such feedback should be provided visually – through augmented reality – combining video with a data overlay. *But would it not be better to let the system talk to the user?* In this way the user would not have to continuously watch the display which potentially could affect the body posture negatively. The second Spike investigated if any of our artefacts offer any interesting opportunities.

Because the first Spike was conducted as a part of our theory understanding and developing process some inconsistencies between the activity presented earlier and our actual conduct occur. We introduce a few small reflection boxes to help provide a clear separation between conduct and our own reflective observations.

9.2 Trigger

At the time we commenced our first Spike we were studying a number of theories and methods related to creativity and constraints. It was decided to apply some of these to make a synthetic halt. Some research suggest that a positive relationship can exist between creativity and constraints [39]. By removing, introducing or black-boxing –manipulating– constraints, an environment can arise that aide creativity. Such constraints can be mental or concrete. We wanted to initiate a synthetic halt by understanding and manipulating with the constraints PhysioSphere had been developed under. The goal was to ensure that the Spike would frame an exploration into a new and unexpected direction.

To understand these constraints we identified the internal and external artefacts used in the development. Internal artefacts are the one we produced including prototypes and code-components. External artefacts denote everything we use – but have not produced ourself – including the Kinect, different kind of SDKs, databases among others. These were decomposed into their affordances and evaluated according to their importance. This process proved to be quite comprehensive. It did show that our technical effort primarily had been put into the production of a prototype that could recognise and verify exercises. Criteria like usability and user experience had been spared little thought. Due to the nature of PhysioSphere– being concerned with the rehabilitation of patients – such areas are of great importance. It was decided to modify our constraints to direct our creative effort in this direction.

Reflection Box: We like the idea that the developers are forced to stop from whatever they are doing and forced to look at the technology in a new way. The efforts put into triggering such a halt must however be minimal to be applicable in a practical context. This is in sharp contrast to what happened during this Spike. Identifying, decomposing and evaluate artefacts to figure out how to manipulate the constraints took more than two days. In hindsight this work effort defeats the whole purpose of having a small quick tool – the Spike – to explore the technology.

However we believe that the concept of the synthetic halt is exciting. So how could a synthetic halt be used to trigger a Spike? One approach could be to initiate a synthetic halt whenever substantial changes happen to external artefacts. Such a change might be a new release of the Kinect, an important SDK or .NET version. These updated might offer new affordances that could be important to the project.

9.3 Spike Challenge

The current PhysioSphere prototype presented in Chapter 4 has a serious interaction problem. The Kinect must be distanced at least 1.5 meters away from Holger to track his movements. This means that Holger's exercise area lies beyond physical reach from the interaction tools – mouse, keyboard or touch-screen – which he would normally use to interact with the system. The issue is illustrated in Figure 9.1.

So how does Holger instruct PhysioSphere to start or stop the exercise session or to initiate a new recording? Currently this is solved by introducing a second person

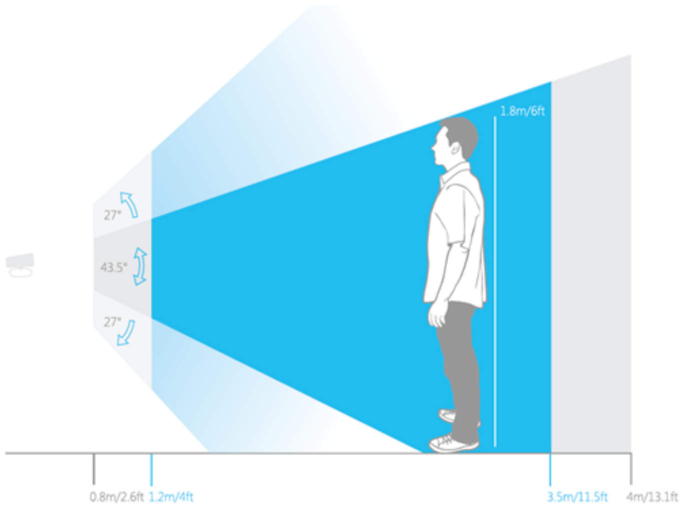

Figure 9.1: The Kinect range [36].

who manipulates the prototype controls. In a real world setting this is not a viable solution. The question that arises from this context is whether the Kinect affords something that helps resolve this interaction problem. Two possibilities were identified during the preliminary research. We could utilise the gesture API, introduced in Kinect SDK 1.7, or look at voice commands. Introducing hand gestures for people with limited hand or shoulder mobility seem like a stillborn idea. Instead our Spike Challenge was scoped to: *"Does the speech recognition of the Kinect afford any interesting interaction options?"*

Reflection Box: We did manage to successfully construct a challenge and scope it. We did not however define the amount of time and resources allocated to conduct the exploration into this challenge. The need for such will be elaborated on in the next reflection box.

9.4 Spike Exploration

The exploration within this Spike was conducted in two steps. First a small inquiry was made into the Kinect specifications to understand the basic affordances and features it provides. The second step involved exploring one or more of these further through the application of prototyping.

9.4.1 Feature Inquiry

Initially an inquiry was made into the audio specifications of the Kinect to get a feel for its affordances. This was achieved by studying the official documentation together with articles located online. While performing this inquiry we applied the technique known as Six Serving Men. It requires participants to consider the questions raised by the six interrogatory words in the English language [44]. These are *who*, *when*, *what*, *why*, *where* and *how*. It is normally applied at the point of initial specification where there are some kind of concept, but before there is a real design or implementation effort [44].

Our inquiry into the affordances revealed two interesting features. The support of voice recognition and a unique feature known as beamforming. The Six Serving Men was applied to each of these.

Command Recognition

Recognising speech was the concept identified for inquiry in the Spike Challenge. The Kinect SDK does not directly support the recognition of free speech. What it does support is the recognition of commands or sentences following a grammar. This grammar can be defined either programmatic or in XML [58].

In the following our application of the Six Serving Men is presented:

What: To recognise commands needed in the interaction with PhysioSphere. Such commands could include start recognition, stop recognition, pause exercise amongst others.

Why: Because voice commands allows Holger to interact with PhysioSphere while being outside the reach of mouse, keyboard or touch screen.

When: We identified several use-cases where voice recognition could be useful:

1. Holger might select an exercise program using traditional forms of interaction. He must however have a way to start the verification when he has moved to the area where the Kinect can track him.
2. If Holger gets distracted he needs a way to pause, stop or move to next exercise.
3. Holger might wish to capture his own exercises without having a second person to tell the Kinect when to start and stop recognition.

How: All features needed to implement command recognition is implemented in the Kinect SDK and the Microsoft Speech API. The needed hardware is integrated into the Kinect.

Where: The technology will be used in Holger's home.

Who: Holger

Beamforming

While researching the command recognition feature of the Kinect a unique feature peaked our curiosity. The Kinect includes a four-element linear microphone array. By utilising that fact that sound travels through air at a constant speed – hence arriving at the microphones at slightly different time – the angle of its source can be identified. This feature is called beamforming.

Our curiosity was peaked by an online article exhibiting how beamforming could be utilised to emulate a directional microphone. This feature seemed really exciting so we decided to apply the Six Serving Men to it.

What: Beamforming can ensure that voice commands are only accepted if they originate from Holger.

Why: There might be multiple sound sources in Holger's home. Such sources could include a stereo, television, family members or Lise.

When: While doing his exercise program in the home.

How: By comparing the angle of the sound source with the angle of the skeleton tracked by the recogniser. Only if these match will commands be accepted.

Both of these features are exciting and offer potential value to PhysioSphere. From our Six Serving Men it became clear that we see beamforming as a tool to ensure a reliable recognition of commands within the context of Holger's home. However, in this case enabling voice commands take precedence.

> **Reflection Box:** In hindsight two points are noteworthy. Firstly, our inquiry into the features of the Kinect did reveal a unique affordance that we had not been aware of beforehand. Secondly, the Six Serving Men showed how reflection can be brought into the Spike process at a very early stage. This inquiry itself can be seen as a small isolated move-experiment where the consequences of it drives further experiments.

9.4.2 Prototyping

The feature inquiry above assisted us in understanding the affordances offered by the Kinect. From this inquiry it is clear to us that the Kinect, at least theoretically, should be a very capable platform for recognising voice commands. Not only does the Kinect integrate such functionality directly into the SDK, but the hardware itself is optimised for capturing speech in a noisy environment. As experienced in the 9th semester project such rationale is no guarantee that the feature is easily implementable. So now our effort changes from understanding affordances to exploring through prototyping.

The first prototype was developed by following a code example. The goal was to gain familiarity with the SDK and the complexity involved in defining and loading grammars. This familiarity would be used to adapt the PhysioSphere prototype to support voice commands.

The code examples followed recreates a famous HCI/NUI challenge known as *Put That There* [58]. This example combines the use of skeleton tracking and voice commands. The user's right hand is tracked and used to control a cursor on a canvas. Objects of different colours can be created on the cursors position by issuing voice commands like `Create Green Circle There`. The prototype can be seen in Figure 9.2. The example is sophisticated enough to give some real hands-on experience with the creation of grammars and logic for handling the voice commands itself. Less than two days were spent on creating and evaluating this prototype and it really surprised us how little effort is needed to integrate command recognition in an application.

Throughout the implementation of this prototype a small list of experiences – for and against the use of voice commands – was developed. This list was used for our final Spike evaluation and is presented in the next section.

> **Reflection Box:** It is again interesting to note how reflection and evaluation became an integrated part of the inquiry. This can be seen as yet another isolated move-experiment. Our intention was to evaluate the complexity involved in implementing voice commands. By following the code-example – moving – we identified both attended and unattended consequences. These were used for evaluation and is presented in the section covering the Spike Evaluation.

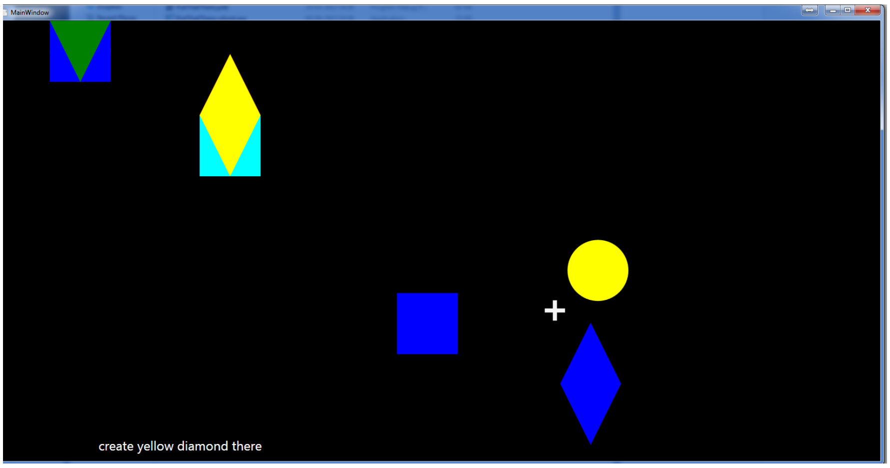

Figure 9.2: Put That There Prototype

The most surprising experience with the *Put That There* experiment was the ease of which voice commands can be integrated into a prototype. The Kinect hardware, Kinect SDK and Microsoft Speech API are all well integrated. With this realisation it was decided that the PhysioSphere prototype, developed in our 9th semester project, should be adapted to support voice commands.

The goal was not on achieving high usability but to create a proof of concept that can be used for reflection and evaluation with potential users. Voice commands must support the user in:

- Starting and stopping the exercise recorder.
- Allow a user to save or discard a recording.
- To allow a user to start and stop a recording.

With all the knowledge and experience already gained we expected the adaptation of PhysioSphere to be quick and easy.

We started refactoring the PhysioSphere architecture to support these voice commands. A command-controller was created encapsulating all command related functionality. Implementing it was as expected fairly easy but adapting PhysioSphere to work with this controller became a major issue. After more than three days we decided to put the Spike on hold.

Reflection Box: The last move-experiment described above ended up being a major learning experience. It emphasised the importance of bounding the amount of time and effort spent on a Spike which we failed to do in this case. The two first inquiries required only little effort on our behalf. The problem arose when we underestimated the complexity of adapting PhysioSphere to support voice commands. To overcome this complexity too much time was spent before realising our mistake and halting the spike.

In hindsight we believe there was another problem with this move-experiment. The first two inquiries had clear goals assigned with them related to the answering of the Spike Challenge. The last move experiment did not. Instead we went beyond the challenge to focus on implementation rather than exploration. The goal of a Spike is to explore the technology in search of opportunities and not to produce code for the PhysioSphere prototype. By starting to modify our

PhysioSphere prototype we inadvertently brought side-effects into the project hereby breaking the isolation of the Spike.

9.5 Spike Evaluation

Even though we failed to adapt PhysioSphere to support voice commands we believe the Spike succeeded in providing an answer to the Spike Challenge. Our *Put That There* prototype showed that the Kinect SDK and the Microsoft Speech API are well integrated and allows developers to quickly implement such functionality. We also realised that the Kinect offers some unique affordances – like beamforming – that could lead to new and exciting value propositions. We summarised these experiences in a small list containing points for and against the use of voice commands:

For:

- The Kinect SDK and Microsoft Speech API are really well integrated.
- It is easy to define a grammar either programmatic a XML file.
- Command recognition seems reliable and offers a great amount of flexibility.
- This provides one way to interact with the system from a distance. The microphones in the Kinect does not have problems recognising commands from several meters.

Against:

- There is no direct support for Danish language. It can distinguish between words like *start* and *stop* but have problems with sentences like *start optagelse*.
- Using voice commands introduce new interaction problems. Holger must know the commands the system can recognise and he must know when each can be given.
- The processing delay between issuing commands and the system recognising them can be long. We might run into interaction problems where Holger is not sure if his commands have been recognised.

Our final conclusion was that integrating voice commands could be worthy solution to our interaction problems and should be researched further.

Reflection Box: In hindsight we believe the evaluation phase to be satisfactory. It contain an answer to the Spike Challenge together with some strengths and weaknesses of using voice commands. The Spike construction might suggest that the Responder must place much effort into reflection and evaluation in this phase. In reality however we expect her to – like it has happened in this case – to be conducted throughout the process. The Spike Evaluation is where the Responder summarises all newly gained knowledge to determine whether or not the Spike was successful

9.6 Outcome

In the Spike Evaluation we concluded that voice control could be a great way to enable interaction between Holger and PhysioSphere. With this conclusion the isolated exploration effort comes to an end. It must now be decided if – and how – this newly acquired knowledge affects the broader project. In this Spike we made a brainstorm session where we went through each of the four Essence views to discuss how the application of voice control could affect their content. The most interesting ideas and opportunities arose on the Paradigm view and is related to the social and spatial context. We believe the most interesting outcome of this Spike is related to doubt being raised concerning the usefulness of visual feedback. In the following we present how this doubt turned out to be the trigger for our second Spike.

Integrating voice control into PhysioSphere allows interaction to happen without the use of hands. In a clinic such functionality could allow Lise to control the exercise recorder while never removing her eyes from Holger. We however also ended up discussing use-cases far outside the current challenge and vision of PhysioSphere. Both authors have experience with Crossfit and regularly train olympic lifts using barbells. Some of the movements are complicated and involve the coordination of multiple muscle groups. We would love to have a system like PhysioSphere to help verify whether or not our technique is correct. The hands must grip the barbell at all time so voice commands would be a useful tool for interacting with the system. We however quickly identified a problem with applying PhysioSphere in such a context. Many of the lifts would require a body posture which prohibit us from looking at a display. Throughout the development of PhysioSphere we assumed that visual feedback – through augmented reality – was the prefered way of providing information to Holger. These assumptions have their source in the research, presented in Section 4.1.1, which uncovered a large number of projects involving serious games and augmented reality.

Realising that using PhysioSphere for Crossfit is far outside the scope of this project we started discussing if the same problem might apply to Holger and the kind of exercises he might do. Through our experience with Crossfit we know a few mobility exercises which we believe could be a part of Holgers program. One of these is illustrated in Figure 9.3. Here Holger has to keep his legs straight while bending forwards and attempting to reach his feet. After this he stretches out achieving full lockout in knees, back and shoulders. Note that Holger has no way of keeping his eyes on the display while being in the bent position.

A potential solution was immediately proposed. Why not provide feedback through sound rather than vision? It has just been argued that there is value in talking to PhysioSphere. It was decided to trigger a Spike to determine if it is possible – and valuable – to let PhysioSphere talkback hereby providing audio feedback.

Figure 9.3: Mobility Exercises

Chapter 10

Second Spike

The first Spike-experiment concluded that voice commands could be applied to help resolve the current challenge of interacting with PhysioSphere. Doing the reflection doubt was raised concerning the use of visual feedback. A proposal was made to explore how audio feedback could be utilised instead. We decided to initiate a second Spike to conduct such exploration. This chapter present our second Spike experiment.

10.1 Spike Challenge

As outlined in Chapter 8 the Spike Challenge is designed to help guide our exploration effort. In this Spike such guideline is needed. As mentioned earlier the first Spike was triggered by a synthetic halt. In the process much time and effort was put into creating the creative obstruction. This meant that we had acquired a good amount of knowledge before even initiating the Spike. This knowledge was essential in shaping the Spike Challenge. This second Spike experiment is radically different. Here the trigger is simple curiosity – a natural halt – rather than a synthetic one. The result is a much smaller knowledge base to help direct our exploration.

This has let us to the following Spike Challenge: "Does our artefacts provide any interesting affordances in relation to providing audio feedback?"

The Spike challenge is kept rather simple and rather narrow, but it can be changed or broadened if time permits. We allocated three days to this experiment.

10.2 Spike Exploration

To explore this challenge two small move-experiments was conducted. Firstly, we identified an external artefact – the .NET framework – which affords text-to-speech functionality through one of its libraries. This can be used to implement audio-feedback. A small prototype was implemented to get some experience with the library. In the second move-experiment it was argued that we needed more knowledge regarding how such audio-feedback could be useful. To acquire this knowledge, the interaction states Holger goes through when interacting with PhysioSphere, was identified and reflected on. In the following we present these two move experiments.

10.2.1 Uncovering Affordances

Initially we had no real knowledge regarding audio feedback and how to enable it. Guided by the Spike Challenge we started experiment by going through our external artefacts to explore their affordances. The Kinect sensor provides microphones and functionality for capturing sound or reacting two voice commands. However it provides no functionality for emitting sound or converting text into speech.

Such functionality was discovered in another external artefact. .NET is a software framework developed by Microsoft primarily for the Windows platform. It includes a large library of reusable software components [56]. While browsing through its libraries we identified a Text To Speech (TTS) library. This TTS library allows applications to convert any text string into speech.

To get a feel for this library a small prototype was developed. To save time it was decided to reuse the *Put That There* prototype developed in the first Spike and extend this. This process involved using the TTS library to speak any command recognised by the Kinect.

The implementation of this prototype only took a day. Integrating TTS functionality proved to be quite easy as it only requires a few lines of code. We did encounter problems with the use of it. When playing back sounds, the Kinect microphone would pick up these and interpret them as yet another command hereby confusing the system. A quick fix was implemented by turning off the Kinect audio stream every time the TTS library was utilised. A more robust solution – allowing Holger to issue a command while the system is speaking to him – could be implemented using the beamforming feature identified in the first Sprint. It could be utilised to emulate a directional microphone always pointing towards Holger.

We found the TTS functionality to be surprisingly easy to implement but was thoroughly disappointed by the quality of the speech itself. The integrated Microsoft voice is metallic and robotic. A quick research effort showed that the .NET TTS library support third-party voices. Such voices – providing better quality of speech – is available in both English and Danish.

10.2.2 Reflecting on Affordances

We found the affordances provided by the .NET TTS library to be really fun and interesting. We wanted to explore these affordances further through the creation of more digital prototypes. But with limited time available we decided not to. Instead a small reflection was conducted to uncover how such functionality might help Holger.

This reflection effort has been conducted as a brainstorm session. The session was comprised of two phases. First we identified the interaction states Holger goes through when interaction with PhysioSphere. Secondly we brainstormed on the potential value in enabling audio feedback in each of these. The identified states were:

Initialising PhysioSphere: In this state Holger initiates PhysioSphere and select his workout .

Instruction: With the workout selected Holger can receive instruction in the exercises he is about to perform. Such instruction is currently not integrated in PhysioSphere but it would be logical step to do so. The instruction could consist of pre-recorded video, audio, text or even through a video link with Lise.

Posture line-up: In this state Holger moves to the the spot where he will be performing his exercises. As elaborated in Chapter 9 the Kinect's limited field of view, requires Holger to maintain some distance to the Kinect and the monitor. Getting positioned correctly, so the Kinect sensor can track feet, hands, and head simultaneously can be a challenge.

Exercise execution: In this state Holger is performing his exercises. In the currently prototype he receives feedback visually.

Final Feedback: In this state Holger has finished his workout. At this point he might be presented with some feedback on his general performance.

With these states defined, we initiated the second step of our brainstorm session. In this step we produced and discussed ideas related to how TTS – and audio feedback in general – could benefit Holger. The result is summarised in the following:

Initialising the Kinect: We could combine voice commands and TTS to allow Holger to select and initiate his workout without looking at the display. We are not sure if this is easier than using a touchscreen, keyboard or mouse. It might however be useful if he is visually impaired.

Instruction: Instructions could be read aloud but we are not really sure if this is valuable.

Posture lineup: In this state audio feedback could be really useful. By combining data from the video camera with data from the skeleton stream we could determine if Holger's complete body is in view. If not, simple voice statements such as: "Move back a little" or "Please move to the left" could be issued.

Exercise execution Here we believe a combination of speech output and pre-recorded sounds could be utilised to provide Holger with better feedback of progress and quality. Simple sounds could be played to mark a completed or failed exercise while voice statements could comment on the quality of his execution.

Final Feedback Again we find it a bit difficult to produce any real value propositions.

With this inquiry finished our three day deadline was nearly reached. We decided to move on to the Spike Evaluation.

10.3 Spike Evaluation

To assist in evaluating the success of this Spike we applied the Six Serving Men method used throughout the First Spike.

What: To provide Holger with feedback using audio

Why: Because Holger need feedback throughout his workout.

When:
1. Holger should be guided to a spot where the Kinect can track him
2. Some exercises require a body posture that prohibits him from watching a display.

How: By combining the TTS functionality available through the .NET framework with pre-recorded sounds.

Where: In Holger's home

Who: Holger

From this small evaluation we can conclude that our challenge was successfully met. This Spike did uncover an artefact which affords functionality for enabling audio feedback. We also uncovered a few potential use cases.

10.4 outcome

The Spike showed that we – using speech – can provide audio feedback to Holger. However this introduce a whole set of other challenges. What exactly do we tell Holger and when do we tell it to him? Answering these challenges is outside the scope of this Spike itself. This is a topic we wanted to explore further but due to time pressure it has not been elaborated on further.

Reflection Box: In our second Spike we allocated three full days to conduct this Spike. In these three days we did not have the time to fully explore the technological potential. We also had no the time to fully explore our new knowledge in a broader context. In hindsight we do not see this as a large problem. A Spike should not be too complicated, but state simple challenges that offer simple answers and exciting reflections. New Spikes can always be initiated to explore these further.

Part IV

Learning

Chapter 11

Reflection & Further Works

In this part we wish to conclude by reflecting upon the learning outcome of our master thesis project. The part will consist of two chapters. In this first chapter we reflect upon the theory which has led to our contribution – the Spike activity – the process of creating it, and suggest areas of further work. The second chapter is a conclusion outlining our contribution – the Spike activity – and its effect within software innovation.

11.1 Pragmatism

In this master thesis project we set out to understand how the Responder in Essence can be supported in her exploration into the technology in search for new opportunities. The problem was that such exploration introduce many uncertainties and risk becoming resource intensive. Locating theory that could help us abstract, and later solve, this problem proved to be a process of discovery. This discovery process let us in many directions. We investigated concepts like technological affordances and creative constraint methods, such as the synthetical halt, Triz and SCAMPER. The problem with all these was that even though they might relate to our problem or help explain parts of it, none of them provided the simple link between problem and solution.

This process also involved a study of the philosophy of pragmatism. Pragmatism provided the theory that allowed us to conceptualise and theorise our practical problem, but also to outline a possible solution strategy. Just as interestingly, it provides a perspective that align well with the way the authors of this report conduct research and practice. This is evident from our approach to our master thesis. Knowledge regarding software innovation could have been obtained through literature studies. Instead we used our 9th semester project to gain knowledge through practice involving the development of an innovative software solution.

Such conduct is also evident from our research effort in our 10th semester. We have felt that only through practice have we had a chance to gain a fuller understanding of the theory and our own practical problem. As such, our generic solution – the Spike activity – has not been the result of a single targeted research effort, but has matured and evolved through continuous reflective conversation between theory and practice. Our own experiences have been essential in shaping the Spike activity. Such experience has come from our own Spike experiments, but just as importantly, from the reasoning involved in developing the cases presented throughout this report. Putting theory – which can appear rather abstract and sometimes intangible – into practice we are forced to build up mental reasoning and practical understanding that to a great extent affects how problem, theory and proposed solution are linked together.

We had no prior knowledge of the pragmatic philosophy before our research these semesters, but it is probably obvious by now, that we have become fascinated by the philosophy. We applaud the idea that a problem is situated and inquiries are needed to understand how to transform this problem into a solution through practice. We believe many engineers and software developers share a desire for similar conducts. Considering this, it is surprising how little pragmatism is encouraged in modern development methodologies. Even on AAU – which prides itself to be based on PBL – many researchers seem to apply a very traditional approach in their courses. In such an approach the researcher, through a number of sessions, present their area of expertise through theory. In our view this encourage rote learning rather than exploration and creativity.

We have found it engaging to explore the concept of pragmatism even though it has been hard and difficult knowledge to get our heads around. We believe that the philosophy is well suited to the Responder in Essence, Essence in general and us as practitioners as software engineers. We learn when we work the technology - we are pragmatics!!

11.2 Essence & The Spike Activity

We believe the paradigm shift Essence proposes is very exciting. What we experienced in the 9th semester project is that going from simply producing software to become innovators is fun and engaging. In a larger perspective we believe the values Essence is based on, are important to develop and shape our industry to become more competitive in a globalised world. We believe software – and technology in general – can be applied to help resolve many of the problems our society are confronted with.

We also like that the Essence values the practitioner as the Hero. To us it seems that often too much effort is put into understanding and modelling the user domain to determine requirements. In this process the reality defined by the technology is rarely touched upon. In this way our technology becomes a source for solutions rather than inspiration. By requiring the developers in Essence to act as double agents – a Responder that must explore the technology and understand its affordances, while reflecting on how this affects the use context as a Child – the technology become a source for inspiration.

However just like tools, materials and activities are needed to support the exploration into the use context, we also believe there is need for these means to explore the technological domain. While many methods exist for the former we have not discovered many practical tools for the latter. The Spike is our humble proposal for such an activity. Here we have an activity that allows the Responder to explore the technology and be reflective. By isolating uncertainties while exploring and keeping the resources manageable, we hope that even a project manager could be convinced of its qualities. Due to the Spike's small footprint we hope to see it applied in daily activities as a part of Essence.

The concept behind the Spike – a managed and isolated inquiry into a problematic situation – could maybe be adapted to support other roles in Essence. We like the idea of a Challenger's inquiry or a Child's inquiry all sharing the same characteristics. A research effort that encourages exploration and reflection but is bond in time and effort, while keeping uncertainties to a minimum and isolated from the main project.

What we regret the most is that we have not had the time to thoroughly experiment ourselves with the Spike activity. We believe it must be matured by applying it in practical projects. The knowledge outcome of during such would be valuable and would probably affect its construction.

11.3 PhysioSphere

We believe that welfare technology is a really exciting topic and an area where innovation can make a difference. There is no denying that our welfare society are increasingly coming under pressure. To maintain our current quality of welfare services this means that we must come up with better services and products that allow us to be more efficient and effective. We believe technology is an important component in doing so. This however requires that someone explores the technology to understand how it can provide value in this area.

On our 9^{th} semester project gave us a taste of exploring the technology in search of such value. We focused on a subsection of the welfare sector – physiotherapy and rehabilitation – in the development of PhysioSphere. We have received much positive feedback regarding the idea of continuously monitoring and verifying the correctness of exercises using the Kinect. We also believe that PhysioSphere is a very exciting project with a real potential in developing further.

As pragmatics and engineers – we find it disheartening not to have had the resources to really develop the PhysioSphere further in this semester. Our Spike experiments with the technology and the creation of our cases, have led to new and interesting use cases: Voice commands, audio feedback and integration of training equipment. We regret not having the time to integrate and mature these ideas into a larger prototypes.

Chapter 12

Conclusion

Throughout the last year we have been exploring the field of software innovation. With initially very limited knowledge within this field we went about acquiring knowledge through experimentation in our 9^{th} semester project with the development of a welfare system known as PhysioSphere. To support us in this development we applied Essence, a new development methodology designed to support the innovative software team. Through the development of PhysioSphere we experienced a lack of support in managing the creative act of exploring the technology. Such exploration is needed to come up with new and innovative solutions, but this effort has a tendency to introduce many uncertainties and become resource demanding.

The ambition with this master thesis has been to discover a way to support the developer in managing such a technology explorational effort in an Essence project. Through a journey of discovery we searched for research and theory that could be applied to our practical problem. Within the philosophy of pragmatism we located theory that not only allowed us to conceptualise our problem but also outline a theoretical solution. We mapped the theory of pragmatism to an innovative context by looking at Responder's inquiry. Through this we came up with a generic solution strategy that has the theoretical foundation of pragmatism, which we believe can be used in a practical innovation context.

We have proposed an activity called a Spike. This activity provides a sandbox wherein a Responder in Essence can explore technological affordance. We believe the Spike activity provides an environment that allows the developer to practically explore the technology in search of unforeseen opportunities while assisting in managing the risks involved in doing so. We expect that the Spike activity will support the developer to reflect on what the technology affords and just as importantly why exactly these affordances matters to us. Such reflection is, in our view, the essence of software innovation. By engaging developers in this kind of activities, we believe a paradigm shift can happen where we go from being software producers to being software innovators.

While reflecting on this thesis we find it interesting that pragmatism not only apply well to the problem and solution but also to our way of performing research and developing knowledge. Even if our master thesis project has concentrated more on theory than productivity we have seen that it is not till we work with things - like our Spike activity - we get a profound and practical understanding of the concept.

We are also fully aware that the way the Spike activity is constructed at the moment, will change over time as other people will get experience by applying this activity. We welcome this and only look forward to the Spike activity maturing into a means that can assist the Responders in becoming innovators.

Bibliography

[1] Ivan Aaen. Essence: facilitating software innovation. *European Journal of Information Systems*, 17(5):543–553, 2008. 9

[2] Ivan Aaen. Software innovation - values for a methodology. 2013. 7, 8, 9, 10

[3] Ivan Aaen. Essence: Team-based software innovation. In *Web*. Department of Computer Science, Aalborg University, http://www.essence.dk, 2013. 9, 10, 25, 30, 38, 44

[4] Agile Alliance. Agile alliance. http://www.agilealliance.org/, 2013. 10, 12

[5] C. Argyris and D.A. Schön. *Organizational learning II: theory, method, and practice*. Addison-Wesley series on organization development. Addison-Wesley, 1996. ISBN 9780201629835. 34, 35, 36

[6] D. Astels, G. Miller, and M.M. Novak. *A practical guide to eXtreme programming*. The coad series. Prentice Hall PTR, 2002. ISBN 9780130674821. 51

[7] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. SEI series in software engineering. Addison-Wesley, 2003. ISBN 9780321154958. 26

[8] K. Beck. *Extreme Programming Explained: Embrace Change*. The XP Series. Prentice Hall, 2000. ISBN 9780201616415. 12, 51

[9] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile development. http://agilemanifesto.org/, 2001. 10, 83

[10] David Benyon. *Designing Interactive Systems*. Pearson Educational Limited, 2nd edition, 2010. ISBN 978-0-321-43533-0. 26

[11] Michael Mose Biskjaer and Kim Halskov. Self-imposed constraints as a creative resource in art and interaction design. *The Second International Symposium on Culture, Creativity and Interaction Design*, 2011. 31

[12] Michael Mose Biskjaer, Peter Dalsgaard, and Kim Halskov. Creativity methods in interaction design. In *Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, pages 12–21. Desire Network, 2010. 10, 31

[13] Michael Mose Biskjaer, Balder Onarheim, and Stefan Wiltschnig. The ambiguous role of constraints in creativity: A cross-domain exploration. 2011. 31

[14] MM Biskjaer and P Dalsgaard. Toward a constrating oriented pragmatism understanding of design creativity. In *Proceedings of the 2nd International Conference on Design Creativity (ICDC2012), Vol. 1*, pages 65–74, 2012. 33, 34, 36

[15] Marcel Bogers, Allan Afuah, and Bettina Bastian. Users as innovators: a review, critique, and future research directions. *Journal of Management*, 36(4):857–875, 2010. 13

[16] K.H. Brodersen and R.H Jensen. Physiosphere. 2012. 17, 25, 30

[17] J. W. Burke, M. D J McNeill, D.K. Charles, P.J. Morrow, J.H. Crosbie, and S.M. McDonough. Augmented reality games for upper-limb stroke rehabilitation. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on*, pages 75–78, 2010. doi: 10.1109/VS-GAMES.2010.21. 18

[18] B. Buxton and W. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Interactive Technologies. Elsevier/Morgan Kaufmann, 2007. ISBN 9780123740373. 26

[19] T. Chaves, L. Figueiredo, A.D. Gama, C. de Araujo, and V. Teichrieb. Human body motion and gestures recognition based on checkpoints. In *Virtual and Augmented Reality (SVR), 2012 14th Symposium on*, pages 271 –278, may 2012. doi: 10.1109/SVR.2012.16. 25

[20] A. Da Gama, T. Chaves, L. Figueiredo, and V. Teichrieb. Poster: Improving motor rehabilitation process through a natural interaction based system using kinect sensor. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 145–146, 2012. doi: 10.1109/3DUI.2012.6184203. 17

[21] W.O. de Morais and N. Wickstrom. A serious computer game to assist tai chi training for the elderly. In *Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on*, pages 1–8, 2011. doi: 10.1109/SeGAH.2011.6165450. 19

[22] John Dewey. *Human Nature & Conduct*. The Modern Library, 1957. 36, 37

[23] Zeynep Erden, Georg von Krogh, and Ikujiro Nonaka. The quality of group tacit knowledge. *The Journal of Strategic Information Systems*, 17(1):4–18, 2008. 13

[24] J. Fagerberg, D.C. Mowery, and R.R. Nelson. *The Oxford Handbook of Innovation*. Oxford Handbooks in Business and Management. OUP Oxford, 2006. ISBN 9780199286805. 7

[25] Stig Enemark Finn Kjersdam. *The Aalborg Experiment - Project Innovation in University Education*. Aalborg University Press, 1994. 29

[26] J.J. Gibson. *The Ecological Approach to Visual Perception*. Resources for ecological psychology. Taylor & Francis Group, 1986. ISBN 9780898599596. 30, 44

[27] Larry A Hickman. *John Dewey's pragmatic technology*. Indiana University Press, 1990. 33, 34

[28] Torben K. Andersen Jens Reiermann. Velfærdsteknologi kan blive en ny vækstmotor. Mandagmorgen, https://www.mm.dk/velf%C3%A6rdsteknologi-kan-blive-en-ny-v%C3%A6kstmotor, September 2010. 7, 8, 17, 18

[29] Anette Kolmos, Lone Krogh, and Flemming K Fink. *The Aalborg PBL model: progress, diversity and challenges*. Aalborg University Press Aalborg, 2004. 34

[30] C. Larman. *Agile and iterative development: a manager's guide*. Agile software development series. Addison-Wesley, 2004. ISBN 9780131111554. 9, 10, 11, 25

[31] Dean Leffingwell. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional, 1st edition, 2011. ISBN 0321635841, 9780321635846. 51

[32] Youn-Kyung Lim, Erik Stolterman, and Josh Tenenberg. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Trans. Comput.-Hum. Interact.*, 15(2):7:1–7:27, July 2008. ISSN 1073-0516. doi: 10.1145/1375761.1375762. URL http://doi.acm.org/10.1145/1375761.1375762. 16

[33] D. Maggiorini, L.A. Ripamonti, and E. Zanon. Supporting seniors rehabilitation through videogame technology: A distributed approach. In *Games and Software Engineering (GAS), 2012 2nd International Workshop on*, pages 16–22, 2012. doi: 10.1109/GAS.2012.6225920. 18, 19

[34] Darrell Mann. Triz: An introduction. *Report for Systematic Innovation Ltd, UK*, 2004. 31

[35] L. Mathiassen, A. Munk-Madsen, P Axel Nielsen, and J Stage. *Objekt Orienteret Analyse & Design*. Marko, 2000. ISBN 8777551530. 26

[36] Microsoft. Kinect for windows sensor components and specifications. http://msdn.microsoft.com/en-us/library/jj131033.aspx, November 2012. 21, 61

[37] D.A. Norman. Affordance, conventions, and design. *interactions*, 6(3):38–43, 1999. 30, 44

[38] Department of Industry. Competing in the global economy: The innovation challenge, 2003. 7, 8

[39] Balder Onarheim. Creativity from constraints in engineering design: lessons learned at coloplast. *Journal of Engineering Design*, 23(4):323–336, 2012. 31, 60

[40] R.S. Pressman and D. Ince. *Software Engineering: A Practitioner's Approach: European Adaptation*. McGraw-Hill series in computer science. McGraw-Hill, 5th edition edition, 2000. ISBN 9780077096779. 9

[41] Rehabilitation.org. Rehabilitation: Types, programs, centers and therapies of rehab. http://www.rehabilitation.org, . 18

[42] Rehabilitation.org. Different types of rehabilitation. http://www.rehabilitations.org/types-of-rehabilitation.html, . 18

[43] Y. Rogers, H. Sharp, and J. Preece. *Interaction Design: Beyond Human - Computer Interaction*. Wiley, 2011. ISBN 9780470665763. 26

[44] Jeremy Rose. *Software Innovation - Eight work-style heuristics for creative system development*. Aalborg University, 2010. 7, 61

[45] Roar Samuelsen. A pragmatist contribution to science, technology and innovation (sti) studies. 33, 34, 36, 37, 38, 52

[46] D.A. Schön. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems*, 5(1):3–14, 1992. 16, 38

[47] Donald A. Schon and Glenn Wiggins. Kinds of seeing and their functions in designing. *Design Studies*, 13(2):135 – 156, 1992. ISSN 0142-694X. doi: 10.1016/0142-694X(92)90268-F. URL http://www.sciencedirect.com/science/article/pii/0142694X9290268F. 38, 39

[48] Olivier Serrat. The scamper technique. 2009. 31

[49] Socialstyrelsen. Velfærdsteknologi på det sociale område. http://www.socialstyrelsen.dk/velfaerdsteknologi. 17

[50] The Origin Story. Skunk works - the origin story. http://www.lockheedmartin.com/us/aeronautics/skunkworks.html. 51

[51] Heidi Sveistrup. Motor rehabilitation using virtual reality. *Journal of NeuroEngineering and Rehabilitation*, 1(10), 2004. 18

[52] K. Tanaka, JR Parker, G. Baradoy, D. Sheehan, J.R. Holash, and L. Katz. A comparison of exergaming interfaces for use in rehabilitation programs and research. 6(9), 2012. 20, 21

[53] H. Thikey, F. van Wjick, M. Grealy, and P. Rowe. A need for meaningful visual feedback of lower extremity function after stroke. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 379–383, 2011. 19

[54] New York Times. The google way: Give engineers room. http://www.nytimes.com/2007/10/21/jobs/21pre.html?ex=1350619200&en=f4b2cd9d18f162bb&ei=5124&partner=permalink&exprod=permalink, November 2007. 51

[55] Stephen E Toulmin. *The uses of argument*. Cambridge University Press, 2003. 15

[56] A. Troelsen. *Pro C# 5.0 and the .NET 4.5 Framework*. Apressus Series. Apress, 2012. ISBN 9781430242338. 69

[57] Phil Turner. Affordance as context. *Interacting with Computers*, 17(6):787 – 800, 2005. 30

[58] J. Webb and J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apressus Series. Apress, 2012. ISBN 9781430241041. 20, 62, 63

Part V

Appendix

Appendix A

The Agile Manifesto

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the
right, we value the items on the left more.

[9]