



UNIVERSITY OF GOTHENBURG

# Usable Interventions for a Product Owner

**A Literature Review Identifying Effective Product Ownership in Lean-Agile Development Organisations**

**MATTIJAS LARSSON**

**Bachelor of Science Thesis**

**Report No. 2013:066**

**ISSN: 1651-4769**

## **ABSTRACT**

Agile and Lean software development methodologies are commonly used today. Agile software development was originally created to uncover better ways of developing software. Traditionally the focus has been on the development teams, and most Agile methods have been aimed at project management and at software design, implementation and testing. As a result most of today's implementations of Lean and Agile have an inward focus on the development team. Delivering real value to customers is not given priority with the risk of development efforts being ineffective. Typical for Lean-Agile environments is that the software requirements and the priorities are channelled through a product owner, making this role responsible for making sure that the development team is really creating customer value. This paper is a literature review of Lean-Agile subject matter expert's books and blogs, which explores available practices and techniques which a product owner can use as interventions in situations where real value is not being generated from the development efforts. The literature review revealed that there are plenty of both old and new practices and techniques available for product owners to apply. It also highlights that a tool in itself is not Lean or Agile; it is the application of the tool which has to be done in compliance with, and with understanding of, Lean-Agile values and principles. For further research in this area it is recommended that a future literature search is extended to also include publications in adjacent fields like product management, marketing, and user experience design, but in a narrowed search. Case studies of the various interventions would also be recommended for future research.

**Keywords:** agile, system development, lean software development, product management, product owner, scrum

# Contents

1. Introduction .....	3
1.1. Problem Area .....	4
1.2. Question .....	5
1.3. Purpose .....	6
1.4. Scope .....	6
2. Methodology .....	6
2.1. Scoping Review .....	6
2.2. Approach .....	7
2.2.1. Define the question .....	7
2.2.2. Inclusion criteria .....	7
2.2.3. Resource search .....	7
2.2.4. Result Screening .....	8
2.2.5. Critical appraisal .....	8
2.2.6. Synthesise the studies .....	8
3. Theoretical Framework.....	8
3.1. Lean-Agile Development .....	9
3.2. Agile Values.....	9
3.3. Agile Principles.....	10
3.4. Pillars of Lean.....	11
3.4.1. Respect for People .....	11
3.4.2. Continuous Improvement.....	11
3.5. Lean Principles .....	12
3.6. Lean Software Development .....	12
3.7. Lean-Agile Methodologies and Frameworks .....	13
3.7.1. Scrum.....	13
3.7.2. XP .....	13
3.7.3. Kanban.....	13
3.7.4. Other.....	14
3.8. Product Owner Responsibilities.....	14
4. Results.....	15
4.1.1. Product Vision .....	16
4.1.2. Return on Investment .....	17
4.1.3. Product Backlog Prioritisation .....	17
4.1.4. Requirements Elicitation and Communication .....	18
5. Discussion .....	18
6. Conclusion .....	20
7. References .....	20
Appendix A – Books by Agile Manifesto authors.....	22
Appendix B – Blogs and Websites of Agile Manifesto Authors.....	23

# 1. Introduction

It is now almost twenty years ago since the Standish group (1995) published their often quoted CHAOS report. This report is based on a research project in where 365 IT executive managers provided details for the outcome of over 8000 IT projects. The report showed that only 16% of software projects were being delivered on time and budget. The biggest reasons why projects fail were said to be the lack of user input and incomplete requirements. The report also indicated that there was a light at the end of the tunnel. It was suggested that smaller time-frames with the delivery of system components early and often could increase the software project success rates.

Different software development methodologies were shaped during this period of time, with different approaches for how to involve the users, shorten cycle times and deliver features early and often. Extreme programming (XP) created by Kent Beck (1999) advocated concepts such as frequent and direct collaboration between customer and developers, and frequent releases. Another methodology called Scrum based itself on a new way of thinking about product development which emerged in the Japanese car manufacturing industry and was first described by Ikujiro Nonaka (Takeuchi and Nonaka, 1986). Scrum prescribed short iterations, a customer representative in the team and servant leadership to replace the role of traditional project management, as described by Ken Schwaber and Mike Beedle (2002) in *Agile Software Development with Scrum*.

In February 2001 at the Snowbird Ski Resort in Utah seventeen representatives of XP, Scrum, and these other so-called lightweight software development methodologies finally came together and unofficially founded what came to be called Agile Software Development. The Agile way of creating software was captured in the Agile Manifesto, which consists of four value statements and twelve principles. These can be found on the [agilemanifesto.org](http://agilemanifesto.org) (2001) website.

The background to this thesis would not be complete without looking at another development methodology which also emerged from the Japanese car industry – the Toyota Way. The product development and manufacturing

principles of the Toyota Corporation is what came to be known as Lean Thinking. It originated in Toyota and has been attributed as the corporate mind-set and principles which has made Toyota one of the world's most successful car makers. The history of Lean thinking is well described by Craig Larman and Bas Vodde (2009) in their eBook *Lean Primer*. An adoption of Lean thinking was introduced into the Agile development community in the book *Lean Software Development – An Agile Toolkit* by Mary and Tom Poppendieck (2003).

\* \* \*

I was first introduced to the Agile software development methodology in 2005 while working as a technical analyst for a software system development company in the online gaming industry. The company was going through a turbulent time and the development organisation underwent significant changes which included the introduction of Agile development using Scrum. I had the benefit of being part of this transformation. I was given *Agile Software Development with Scrum* (Schwaber and Beedle, 2002) to read and instructed to now liaise with a person in the development team called the product owner. This project was later described in more details by the consultant who led the development organisation during this period of change called Scrum and XP from the Trenches (Kniberg, 2007). This was my first glimpse of Agile software development and I captured my interest for years to come.

I have since performed the product owner role myself in various development projects and I am a certified product owner through Scrum Alliance. I have acquired skills and knowledge required through books, blogs and endless hours of backlog grooming, sprint planning and retrospective meetings. I have seen several successful projects where the development adopted Lean-Agile practices and delivered quality software in short time frames. I feel that there is an abundance of tools and techniques available and promoted in the Lean and Agile methodologies for how to do successful software development. What has been missing in the Lean-Agile literature however are practically oriented advice on how a product owner

interacts with the stakeholders and customers outside the development organisation. A product owner is expected to effectively represent the entire business, the market and the customers to the team, still agile literature is surprisingly silent on how that is best achieved.

## 1.1. Problem Area

The adoption of Agile and Lean principles has massively influenced the software development industry of today. Ben Linders (2012) summarised the results of different studies of Agile development in an article published on InfoQ.com. His conclusion is that the shift towards Agile and Lean thinking in software development has led to measurable improvements in both productivity and quality. Mike Cohn (2012) comments on the results of the *Standish group CHAOS report 2012* in his blog at *mountaingoatsoftware.com*. He highlights that the report indicates that the software projects where Agile development has been applied has three time higher success rate. Discussing the accuracy of these reports or the methods used in the research falls outside the scope of this thesis. I will only state that there is available evidence of the improvement Agile can bring to development projects, which is backed up by the large number of companies who have adopted Agile and Lean in recent years.

Agile software development was created to uncover better ways of developing software. Traditionally the focus has been on the development teams, and most Agile methods have been aimed at project management and at software design, implementation and testing. To guarantee that the customer is satisfied Agile prescribes customer collaboration, early delivery of working software and welcoming change when it occurs late in a project.

From experiencing large scale projects first hand I have come to observe a major problem with how Agile is rolled out in many product development organisations. The end user or customer is actually rarely available to the development team and does not get to use the working software which the team is incrementally delivering. Instead customers are represented by business stakeholders, typically product managers, account managers, sales representatives, or marketers, depending on the type of company. These stakeholders in turn

communicate to the development team through a single person - the product owner. I believe that this structure is partly a consequence of using Scrum, the most commonly used process framework in Agile development. Scrum defines product owner as the role responsible for communicating and prioritising requirements.

One of the first descriptions of the role product owner can be found in *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle (2001). Here the product owner is defined as the role which solely controls the requirements backlog. It is one person, and it is the only person who can change scope and prioritise the backlog items. In the more recent *Agile Product Management with Scrum* Roman Pichler (2010) describes the product owner as a multi-faceted and context based role which reshapes stakeholder authority, product management and project management in an organisation where it is introduced.

With the product owner in an Agile environment focus shifts towards the software development and delivery. Requirements are gathered from the stakeholders, analysed and communicated to the team which designs and implements the solution. I have observed how teams that are following the Agile principles of focusing on working software and delivering frequently become very successful at building to specification and deliver on time.

The problem here is that this does not guarantee that what has been built and delivered is something that actually solved the problems of the customer and end user. In the situation I have described the product owner is disconnected from the customers and cannot assure that the project delivers quality and creates value for the end user, and as a consequence no long term value for the organisation.

Dan North (2006) summarises the problem in a blog post *Outcomes over features – the fifth Agile value* where is arguing that Agile teams are too focused on delivering requested features, when they should really be focusing on delivering outcomes.

The problem was recently described by Nick Coster (2013) in an interview where he calls it *The Agile Business Gap*. This gap exists where there is no well understood and repeatable way of converting user needs and business objectives into the level of

detail that is required to work effectively with an Agile development team. Coster raises a concern that the focus today is on the core software development and anything that is beyond the boundaries of the requirements backlog and a release is largely ignored by any of the Agile methodologies.

Mary Poppendieck, co-author of *Lean Software Development*, published a blog post (Poppendieck, 2010) in her collection of Lean Essays where she describes the problem of having a product owner who is too focused on requirements analysis and interaction with the team. She draws the attention towards Lean thinking and reminds us that “Discovery of the right thing to build is the most important step in creating a good product. Get that wrong and you have achieved 100% waste.”

On a parallel track lean ideas have gained popularity in the field of Product Management. In the best-seller book *Lean Startup* Eric Ries (2011) expands on Steven G. Blank's concept of Customer Development, first introduced in *The Four Steps to Epiphany* (Blank, 2005). Customer Development is based around the key concept of questioning all assumptions made when the product is being developed, including assumptions around who your customer is. Creating a product which solves a problem for customers, which said customers are prepared to pay for and the revenue from selling the product is greater than the cost of developing it, is called achieving product-market fit. Steven Gary Blank and Eric Ries argue that in order to achieve it, product managers in the modern fast-moving landscape must iteratively learn from how real customers in the market react to the product and adapt.

Earlier this year Henrik Kniberg published a post (Kniberg, 2013) on his blog at [crisp.se](http://crisp.se) titled *How to Build the Right Thing*. He is claiming that the software industry again is going through a major shift in mind-set. He argues that Agile software development has now solved the problem of how to develop and release software. The biggest waste in software development now seems to be that the wrong products with the wrong features are being built.

He is also mentioning that there is a number of techniques which have been developed to help the team and the product owner in making sure that what is being built is not just delighting internal stakeholders, but is actually satisfying the

customers.

So how can we build on the existing success of Agile software development and expand Agile thinking to include the complete product development process from where a customer need is identified to when a complete and valuable solution is delivered?

## 1.2. Question

The research question can be formulated as:

- *Which Lean-Agile practices and techniques are available to effectively enable a product owner to deliver real customer value through the efforts of a product development team?*

In this section the boundaries of the question are defined. First the population where the problem exists is examined in more detail. The types of practices which are believed relevant to solve the problem are further reviewed, the types of outcomes we are focusing on described, and an outline of the overall context in which this question is relevant is provided.

The population of interest for this paper includes any organisation in the digital product development space. All recommendations which can be found for product owners in these organisations are considered to be of interest when answering the research question.

In this paper practices, methods and techniques are referred to as interventions. An intervention is defined as a practice or technique which a product owner can apply to bring about change in an environment where development is not creating value. In this study only possible interventions which fall within the remit of a product owner in a Lean-Agile environment are considered.

Usage of the interventions is compared to the scenario where the product owner role is present in a Lean-Agile development environment, but is only responsible for looking inwards to the team, and has little meaningful interaction with customers. This is the situation in many organisations today as described in the problem area section, and this is the situation we are seeking effective interventions for.

To specify the question further this paper focuses on outcomes with a direct impact on the delivered customer value. It is possible that there are

interventions more suitable when focusing on increased short-term shareholder value and other business goals. These outcomes can be both relevant and important to the organisation, but are not within the scope of this paper.

To find effective interventions the context in which a research question is asked has to be limited. The reason is that an intervention which works well in one context cannot be expected to produce the same results in a different one. Typical examples of contexts which would impact the efficacy would be the differences between organisations, like industry, business model, organisation size, the resources available for investment in product development, market and competition, and many more. In this paper the broadest possible context has been considered and the research question is answered for all Lean-Agile development organisations. This is explained further in the *methodology* section below.

### **1.3. Purpose**

The primary purpose of this study is to provide recommendations for further research into the challenges of product ownership in Lean-Agile

## **2. Methodology**

To answer the research question outlined in the previous chapter an initial literature review was undertaken. The method used was based on the approach given in *Systematic Reviews in the Social Sciences* by Mark Petticrew and Helen Roberts (2006). A systematic literature review is an established method for searching, sorting, reviewing, appraising and report on large volumes of existing writings related to a research question. According to the authors the method can be used to make sense of large bodies of information, and a means of contributing to formulate answers. Petticrew and Roberts are describing the approach in the context of social interventions with the purpose of answering questions around what works and what doesn't.

This is a suitable approach as we are researching the effectiveness of product owner interventions for creating value in software development projects. Rather than evaluating a practice or

organisations when seeking to resolve ineffectiveness caused by gaps between the product delivery and the customer value stream.

A secondary purpose is to provide actionable recommendations for product owners in the field by compiling and publishing an initial list of interventions which can be used for improving product quality.

### **1.4. Scope**

The effectiveness and usefulness of Lean and Agile methodology or which types of projects it is suitable for will not be debated in this paper. Whether Agile software development is the right approach or not for an organisation or how it can lead to improved results compared to other methodologies also falls outside of the scope.

This study focuses primarily on the product owner's responsibilities outside the development team. A significant amount of any product owner's effort will be spent with the team in activities related to communicating, discussing and supporting the development effort. While it is acknowledged that this is a key area for the product owner, it is the outward facing part of the role that is studied in this paper.

technique based on the outcome of single case studies I believe it is more relevant to describe them based on multiple accounts of various different evaluations.

When attempting to establish efficacy of social interventions researchers in that field uses the systematic reviews to compare multiple studies to get an understanding of under which circumstances an intervention can be recommended. This resonates with what has previously been written about the use of methods and techniques in Lean-Agile development. We are not attempting to define best practice for Lean and Agile, but rather understand when to use the different techniques at hand.

### **2.1. Scoping Review**

A full systematic literature review in this domain would require the collection and assessment of a

very large number of books, articles, papers and other resources. A complete review was therefore not an option for the study conducted in a limited time period. Narrowing the research question would have shortened the literature search time and reduced the amount of resources to review. However, this option is not recommended without having previous studies available to provide guidance for how the question is effectively narrowed down. The other option is to perform a scoping review (Petticrew and Roberts, 2006) which is an initial limited review with the purpose of providing an early indicative answer to the research question and to give recommendations for a full systematic literature review. The recommendations would include indications of where to search to find a complete set of relevant resources and also suggestions for how the research question could be reformulated. Reformulating the question involves lumping it together with more questions if it is found to be too narrow, or split it up and focus on part of it in case it is indicated that the original question is too wide.

## **2.2. Approach**

The approach to searching and appraising literature, and compiling the results for this paper is based on the seven stages of carrying out a systematic review described by Petticrew and Roberts (2006).

### **2.2.1. Define the question**

Petticrew and Roberts (2006) recommend that reviews aimed at answering questions about effectiveness is framed using the *PICOC* model. *PICOC* is an acronym for population, intervention, comparison, outcomes, and context. The *question* section in the previous chapter has been structured to follow this model.

### **2.2.2. Inclusion criteria**

Petticrew and Roberts (2006) recommend that inclusion criteria are defined before the search for literature begins. Literature with qualitative and quantitative studies of interventions, as well as theoretical descriptions of interventions by Lean-Agile subject matter experts was included in this search. In order to appraise a large quantity of

written resources in the limited time of the scoping review the criteria for inclusion was to have available meta-data which indicates that the publication describes one or many interventions which can be used to answer the research question.

### **2.2.3. Resource search**

The systematic literature review method described by Petticrew and Roberts (2006) is aimed at searching academic databases for published articles and papers. This review in the field of Lean-Agile development focuses on the writings published by subject matter experts. The majority of this work is not available in academic papers or articles. However, as Lean and Agile development methods have been created and refined by practitioners in the software development industry the books published by these experts are seen as the original source of information. Information which over time has been complemented with blogs, articles, research papers, conference talks, etc. Today there is no shortage of accessible information in the field of Lean-Agile development. That said it is not easy to construct a systematic, reproducible method to review this literature.

One of the objectives of the scoping review is to establish an initial strategy for the search, which is evaluated together with the results. To accomplish this, focus was placed on the Lean-Agile subject matter experts rather than on their publications. A starting point was established with the authors of the Agile manifesto and the books and blogs they published on the subject of Lean and Agile methodologies.

The initial resource search for relevant publications by the seventeen Agile Manifesto authors resulted in 22 publications, listed in Appendix A, and 13 blogs, listed in Appendix B. These were all initially identified as possibly being relevant to identify interventions and to find further subject matter experts.

The primary resources for this paper are published books and blogs. Attention has also been paid to research papers, conference notes, articles, online discussion forums, news groups and even twitter feeds for the purpose of identifying further resources to review.



### 2.2.4. Result Screening

From the initial set of resources further subject matter experts were identified as their publications are either referenced or recommended in the initial resources. In this stage of the search bibliography references and recommendations in books and blogs were followed to identify further subject matter experts and their writings.

The purpose of the screening is to determine if a book or blog is meeting the inclusion criteria of the review. For books this was done by reviewing the title, sub-title and book description for each resource and matching it against the research question. For blogs the title, tag-cloud, blog description and categories, and a search for posts containing “product owner”, was initially used depending on which elements of the meta-data that was available. Authors referenced or recommended were added to the subject matter expert list and their publications were screened using the described inclusion criteria.

By following this systematic approach, sometimes called snowballing (Petticrew and Roberts, 2006) a tree structure of Lean-Agile experts and their publications was created with a large collection of books and blogs being identified for the scoping review.

This search generated a list of 113 published books in total and 62 blogs of various Lean-Agile subject matter experts and specialist interest groups based on the inclusion criteria. The search was time-boxed to approximately four weeks. A few books and blogs were found later during the detailed reviews and appraisal that followed and were included as well.

After the search was halted; the books and blogs were screened in more detail and evaluated against the research question. Finally ten books and six blogs were selected for a detailed review as interventions believed to be of interest were found. These books and blogs are references together with the interventions they describe in

Table 1 in the *results* section.

The practice of using bibliographical references to expand the search in a literature review is a well-established practice (Petticrew and Roberts, 2006), however linking this together with blogging (self-published online articles and discussions) is an experimental approach taken in this search. It is evaluated in the *result* section of this paper.

The intention was not to produce a complete list of relevant resources, as for the purpose of a scoping review this limited search can be deemed sufficient. In theory the results from the method are reproducible, but given the large amount of resources of different types, many in the digital space where content is volatile, it would be practically impossible to recreate the tree structure with an identical result. This can be considered a problem in a systematic approach and it is addressed further in the *discussion* section. Finally, the search was limited to resources published in the English language.

### 2.2.5. Critical appraisal

The interventions found in the various reviewed resources were assessed against the values and principles of Lean-Agile as described in the theory section of this paper. Synergies between the interventions described and the Lean-Agile principles are emphasised. The appraisal of the selected resources can be found in the *results* section of this paper.

### 2.2.6. Synthesise the studies

The insights gained from the search and resource appraisal are further described in the *discussion* section. For the purpose of this scoping review the recommendations for future systematic literature reviews for any similar research questions has an important part as well. Recommendations for future research based on the findings are also included in the *discussion* section.

## 3. Theoretical Framework

To meet the goals of the study, the identification of practices and techniques which are compatible with Lean-Agile development is required. In order to assess the literature in the review we therefore

need clear definition of Lean and Agile. This chapter provides definitions which were used to assess the literature.

### **3.1. Lean-Agile Development**

Lean-Agile development practices are compliant with Agile and Lean thinking. Agile practices honour four values and adhere to twelve principles described in the Agile manifesto. Practices compatible with Lean must not compromise either of the two pillars of Lean thinking. The Lean practices must also adhere to all the 14 principles of Lean thinking and comply with the seven principles of Lean software development. All the values and principles are described further in this chapter. The interventions identified in this study need to meet the criteria outlined above.

There is not one single definition for the relationship between Lean and Agile. They both originate in Japanese corporations and are closely linked with Japanese management culture. However there are different opinions on whether Agile development is another principle under the umbrella of Lean thinking or if it is maybe Lean practices that should be considered a methodology within Agile development, or both! This discussion is out of the scope for this paper. As stated in the definition above I seek practices which are compliant with Agile and Lean and these results are relevant independently from how Lean and Agile are considered to be related.

### **3.2. Agile Values**

Agile software development in this paper refers to the values and principles outlined in the *Agile Manifesto* as it is published on [agilemanifesto.org](http://agilemanifesto.org) (2001). At the core of Agile thinking are the four values:

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

Martin Fowler and Jim Highsmith, two of the authors of the Agile manifesto, published an early article titled *The Agile Manifesto* (Fowler and Highsmith, 2001) where they provide their interpretations and reasoning behind the value statements in the manifesto. The first important thing to observe is that each value statement has a

left and a right segment. Cultures which have adopted Agile values are prioritising the left hand segments over the corresponding ones on the right hand side.

First of all the authors point out that Agile is not anti-methodology, and as we will see when looking closer at the frameworks and methodologies promoted under the Agile umbrella, such as Scrum, there is a large collection of processes and tools to be found. The first value however dictates that the individuals and the interactions between them must take priority. Any process or tool that is used is there to support the individuals. In his book *Agile Estimating and Planning* Mike Cohn (2005) states that an Agile organisation acknowledges the unique strengths of its individuals and capitalise on these, rather than treating them as resources and replaceable cogs of a machinery.

Comprehensive documentation is sometimes required also in an Agile organisation, but the second value dictates that working software is always the priority. Every team must determine when creating documentation is essential (Fowler and Highsmith, 2001) and keep their focus on stable, incrementally enhanced versions of the product. This will allow the team to get early feedback and make sure that the team is working on the highest valued features (Cohn, 2005).

Fowler and Highsmith (2001) acknowledge that negotiating internal or external contracts is not a bad practice, but it is not sufficient to deliver a successful project. A contract can be used to establish boundaries and conditions for the project. However, the collaboration between the customers and the development teams is the key to delivering value to the client. In their book *Practices for Scaling Lean and Agile Development* Craig Larman and Bas Vodde (2010) warn about how contract negotiation turns into a competitive game resulting in that the involved parties optimises locally to maximise their value from the contract. By focusing on the collaboration with customers first it is possible to abolish the contract game and focus on optimising the whole.

Commenting on the fourth and last Agile value statement Fowler and Highsmith (2001) states that following a plan is extremely dangerous in software development if you become blind to change. After examining plenty of successful software projects they have concluded that these projects have deviated from the original plan as the teams were

able to respond to changes along the way. In *Agile Estimating and Planning* Mike Cohn (2005) emphasises the importance of estimating and planning in order to reduce risks and uncertainties, support decision making, establish trust, and convey information. However, the resulting plans are to be seen as snapshots of how the world appeared at that point in time when the team embarked on a road into an uncertain future. Agile planning shifts the emphasis from the plan to the collaborative planning activity.

Our first criteria for evaluating interventions will be to validate that they are compliant with these four values as they are expected to be found at the heart of any Agile organisation culture.

### **3.3. Agile Principles**

In addition to the values, the authors of the Agile manifesto also agreed on twelve principles to guide an Agile organisation. Here are the principles as they appear in the appendix to the manifesto, *agilemanifesto.org* (2001):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter time-scale.
4. Business people and developers work together daily throughout the project.
5. Build projects around motivated individuals, give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information with and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers

and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity — the art of maximising the amount of work not done — is essential.
11. The best architectures, requirements and designs emerge from self-organising teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

The multiple frameworks and methods available for Agile teams to follow are all based on these principles, and in some cases the principles were derived from insights gained from using early methods such as Extreme Programming (Beck, 1999). These principles serve as the second criteria for evaluating product owner interventions. I consider three of the principles more important as they are directly related to customer and stakeholder interaction. Principles one, two and four were given special attention in the screening of material.

The very first Agile principle dictates that the team must prioritise continuous delivery of valuable software above all. Fowler and Highsmith (2001) call this the customer value principle and reiterate the importance of not following plans, but rather continuously seek to deliver actual customer value. Uncovering practices and techniques to do so is the essence of this thesis.

The second principle is closely related to the first. Agile processes are aimed at creating competitive advantage for the customer and will always accommodate change in requirements if it is needed. Fowler and Highsmith (2001) argue that changes in the new economy are caused by the disruptive changes in both business and technology. This should not be seen as a threat to guard against, but rather as an opportunity to embrace.

The fourth Agile principle dictates that business people and developers work together daily. Fowler and Highsmith (2001) point out that daily was added to the principle to emphasise that customers and stakeholders involved in the process of selling the product are expected to be committed to the development project, take joint

responsibility, and play an active part.

### 3.4. Pillars of Lean

In the ebook *Lean Primer* Craig Larman and Bas Vodde (2009) summarises the philosophy behind Lean thinking as it has been described by Toyota where it originated. They describe the goal of Lean as: “Sustainably deliver value fast”. In a lean organisation reaching this goal is supported by the two pillars of lean:

- Respect for People
- Continuous Improvements

When evaluating Lean-Agile practices I expect them to honour both the respect for people and the commitment to continuous improvement that Lean thinking promotes.

#### 3.4.1. Respect for People

The first pillar is *Respect for People*. Larman and Vodde (2009) explain how this can be divided into six different areas.

The first is called “Don't trouble your customer”. A customer can be internal or external. It is anyone you are offering your services to or anyone who consumes your work or your decisions. Not troubling customers includes not giving them faulty outputs, not making them wait and not overloading them.

The second area is called “Develop People and Then Build Products”. Managers in a Lean organisation must act as teachers, not as directors. People in the organisation should be mentored and taught how to analyse root causes and how to make problems visible in order to improve. The third area provides further guidance to Lean management and is called “Manager 'Walk the Talk’”. All managers are expected to set an example and to understand and act according to Lean pillars and principles.

The fourth and fifth areas are related to teamwork. “Teams and Individuals Evolve Their Own Practices and Improvements” stipulates that management can provoke thoughts and challenge the people in the workplace, but it is the teams and individuals who must decide how they can improve. The area “Develop Teams” places emphasis on the importance of creating strong and gelled teams. The culture must be based on teamwork, not

group-work.

The sixth area has been named “Build Partners”. A Lean organisation aims to establish long term relationships with their partners. This includes proactively supporting partners, helping them improve and stay profitable, for mutual benefits.

#### 3.4.2. Continuous Improvement

Larman and Vodde (2009) explain the thinking behind continuous improvement, the second pillar of Lean. The main concept is that individuals, teams and the whole organisation continuously inspect themselves and find ways to improve. This is an on-going process which is part of everyone's job and it is based on the following ideas.

The first one is called “Go See”. Toyota is claiming that this is both critical and fundamental to Lean thinking. In the internal book *The Toyota Way* it is highlighted as the first factor of success in continuous improvement. The idea is that managers must not spend all their time in separate offices. They must find the real front-line place of work where the organisation is creating the value for its customer and spend time with the teams there. Problems must be understood in the right context and resolved at the source.

The second idea is known by its Japanese name; *Kaizen*. It is often translated simply as continuous improvement, but at Toyota it has a deeper meaning. It captures the philosophy and mind-set of the individual in the organisation which can be described as “My work is to do my work and to improve my work”.

*Kaizen* also introduces the Lean concepts of value and waste. By breaking down processes and understanding which activities are creating real value and which ones are wasteful Lean teams can determine where they need to focus their improvement efforts. A definition of value in this context would be: *The moments of action or thought creating the product that the customer is willing to pay for*. The customer here is always the end-customer, not internal business stakeholders. All other moments can be considered wasteful and should be removed. This typically includes bottlenecks, waiting, handoffs, task-switching overhead, defect fixing, missing knowledge, etc. ad infinitum.

Other continuous improvement ideas include the

*Perfection Challenge*, always aiming for perfection, and the *No Final Process* attitude. There will never be a final set of best practices or central process at Toyota or in any other Lean organisation as the idea is to repeat the cycles of improvement experiments indefinitely.

### **3.5. Lean Principles**

From building the organisation around the two pillars of Lean Toyota have been able to identify 14 principles which further define how Lean can be implemented and provide guidance. Larman and Vodde (2009) have summarised the 14 principles as:

1. Base management decisions on a long-term philosophy, even at the expense of short-term financial goals.
2. Move toward flow; move to ever-smaller batch sizes and cycle times to deliver value fast & expose weakness.
3. Use pull systems; decide as late as possible.
4. Level the work — reduce variability and overburden to remove unevenness.
5. Build a culture of stopping and fixing problems; teach everyone to methodically study problems.
6. Master norms (practices) to enable *kaizen* and employee empowerment.
7. Use simple visual management to reveal problems and coordinate.
8. Use only well-tested technology that serves your people and process.
9. Grow leaders from within who thoroughly understand the work, live the philosophy, and teach it to others.
10. Develop exceptional people and teams who follow your company's philosophy.
11. Respect your extended network of partners by challenging them to grow and helping them improve.
12. Go see for yourself at the real place of work to really understand the situation and help.

13. Make decisions slowly by consensus, thoroughly considering options; implement rapidly.
14. Become and sustain a learning organisation through relentless reflection and *kaizen*.

As with the Agile principles I will not go into detail of all of them in this paper, but focus on the ones which can be considered more relevant for a product owner's responsibilities. In this paper the first, third, seventh and thirteenth principle has been given extra attention.

According to the first principle of lean thinking management decisions must be made with the long-term philosophy of the organisation and product in mind. Any practices used by product owners must be compliant with long term thinking over short term profit optimisation.

The third and thirteenth principles provide important guidelines for decision making. The third principle states that work gets pulled when the team can take it on, and that decisions are deferred to when they are needed. Product owner techniques which rely on up-front detailed decision making cannot be considered compliant with Lean thinking. The thirteenth principle says that decisions should be made by consensus and by considering options. The combined principle of discussing multiple options and deciding by consensus as late as possible in the process must be adhered to by any Lean-Agile intervention.

The seventh principle adds the concept of visual management. This principle both promotes the power of visualisation, as well as the much larger philosophy of openness and transparency.

### **3.6. Lean Software Development**

The 14 traditional practices of Lean thinking as described in *Lean Primer* by Larman and Vodde (2009) were adopted specifically into software development by Tom and Mary Poppendieck (2003) and described in their book *Lean Software Development*. They have defined seven Lean principles to guide an organisation to integrate Lean thinking with Agile development:

1. Eliminate Waste
2. Amplify Learning
3. Decide as late as possible

4. Deliver as fast as possible
5. Empower the team
6. Build Integrity In
7. See the whole

When evaluating interventions for the product owner the first, second, third and seventh principles are considered to be of extra interest for the review.

The first principle is focused on *waste elimination*. It is the product owner who is responsible for guaranteeing that what gets built will generate value for the customers, and that no time or resources is spent in other activities as this would be wasteful. A product owner should not focus on the products that are being created, but on the customer problems they are solving in order to understand what is valuable and what is not.

The second principle of Lean Software Development is to *amplify learning*. The key concept here is not to rely on predictions and guesswork and create plans, but to develop a capacity to quickly learn and respond. Part of this is to find multiple options for solutions and make decisions at the last responsible moment based on the knowledge available at that point in time, which is emphasised by the third principle: *Decide as late as possible*.

*See the whole*, the seventh principle, states that a product owner must clarify the purpose of the product and the development effort. All processes must be optimised for the entire value stream from the initial concept to retirement of the product. Optimising any part in isolation will sub-optimize the whole. Optimising the whole also involves optimisation over time by applying long-term thinking.

As we see here the principles of Lean Software Development are directly derived from the original principles of Lean thinking and the Toyota way, but tailored to suit software product development.

### **3.7. Lean-Agile Methodologies and Frameworks**

As previously mentioned there is a multitude of frameworks, practices and methods available which are said to be Lean-Agile compliant. Scrum, XP and Kanban are the three most common today.

These can be used on their own or together in various combinations as they are adopted by Lean-Agile development teams. The details of the various methods fall outside of the scope of this paper. To answer the question of how product owners can be effective working with teams who have adopted one or multiple Lean-Agile practices it is however important to understand how the product owner role is defined in different methodologies.

#### **3.7.1. Scrum**

According to the *7th Annual State of Agile Development Survey*; Scrum is today by far the most popular framework for Agile development with over 70% of Lean-Agile development teams following Scrum or a hybrid version of it (VersionOne, 2013). The Scrum framework provides a set of practices and rules for the complete software development life-cycle. It was originally described in Agile Software Development with Scrum by Ken Schwaber and Mike Beedle (2001). They describe the key elements of Scrum as having small, cross-functional teams, which under the servant leadership of a Scrum Master work in short iterations called sprints. In each sprint the team commits to taking a number, based on their own estimates, of requirement items from a product backlog and have these implemented into a working system by the end of the sprint. The product backlog is maintained by a product owner which is the single source of truth for requirements and business priorities for the team.

#### **3.7.2. XP**

Extreme Programming (XP) as described by Kent Beck (1999) is often quoted as the method which “started it all”. According to the most recent *State of Agile Development survey* (VersionOne, 2013) there are few teams today developing using XP as their only method. However Scrum in combination with XP practices is a common combination as described by Henrik Kniberg (2007) in *Scrum and XP from the Trenches*. In this combination Scrum focuses on project management and organisation practices while XP practices focuses on programming activities.

#### **3.7.3. Kanban**

Kanban as a method for Lean-Agile Development

was introduced in *Agile Management for Software Engineering* by David J Anderson (2003). Originally developed by Toyota as part of their Lean production system it addresses some of the core concepts of the Lean principles. Kanban provide simple and powerful methods to visualise the work-flow and limit the work in progress. The usage of Kanban by Lean-Agile development teams is currently below 10% (VersionOne, 2013), but it has almost doubled over the past year and it is the fastest growing area of Lean-Agile methodology. Kanban does not specify a product owner role or a prioritised product backlog.

### 3.7.4. Other

More well described, but less used, methods for Lean-Agile development can be found in Feature Driven Development and Dynamic Systems Development Method. Other early Agile development methods like Adaptive Software Development and Crystal Methods are no longer quoted as being used at all (VersionOne, 2013).

Although most practices used in Lean-Agile development today are found in Scrum, XP and Kanban it is possible to identify interventions in other methodologies and so they remain in scope for the literature search.

## 3.8. Product Owner Responsibilities

The product owner role was introduced in the *introduction* section. The product owner is defined as the role which solely controls the product backlog. In Scrum the product owner is one person, never a committee, and it is the only person who can change scope and prioritise the backlog items. The authors point out the product owner can be a product manager, department manager, or other customer representative as long as the persons in the role are accountable for the product owner responsibilities (Schwaber and Beedle, 2001). The following responsibilities are compiled based on the outward focus of a modern product owner as described earlier and in *Agile Product Management with Scrum* Roman Pichler (2010). As stated in the *question* section of the paper it is important to understand the remit of a product owner in order to identify valid interventions:

- **Product Vision**  
The product owner defines and communicates the long-term vision of the product to both stakeholders and development teams, and monitors the development against goals and vision.
- **Return on Investment**  
Maximising the return on investment of the product development initiatives is officially a product owner responsibility. The product owner represents and manages stakeholder and customer interests and is accountable for deliveries, ensuring the value of the work the development team performs.
- **Product Backlog Prioritisation**  
The product owner maintains the product backlog as an ordered and prioritised list of requirement items to be implemented by the team to achieve goals and missions. Ensuring that the product backlog is visible, transparent, and clear to all is also a responsibility which falls under here.
- **Requirements Elicitation and Communication**  
The product owner defines features, clearly expresses product backlog items, and answers any product clarification questions. The product owner ensures that the development team understands items in the backlog to the level needed and gives immediate feedback on their work. The product owner is also responsible for assessing the outcomes, both deliverables and learning, of each iteration and update the product backlog and priorities accordingly.

The title product owner was originally defined for Scrum use, and is used throughout this paper when referring to the role responsible for the strategic vision, business priorities and product requirements as described above. This could for example be a product manager when working with a team using Kanban. This paper focuses on Lean-Agile development using any framework or methodology, so wherever references are made to the product owner it is the role responsible for the areas of product management above that is being considered.

## 4. Results

This section lists the interventions identified in reviewed literature. This is not an exhaustive list of all possible interventions, but a sample of practices and techniques described in the books and blogs identified. The selected interventions are listed below in Table 1 together with a brief description.

Include here are various simple and complex practices and techniques grouped by area of product owner responsibility. Some of the listed interventions are useful in more than one area. The categorisation made here is only for the purpose of providing a clear initial overview.

Area	Intervention	Description	References
Product Vision	Elevator Pitch	A technique for distilling the product value proposition into a discrete, easy-to-remember, compelling, and repeatable phrase. Writing the pitch can help a group develop and communicate a vision.	<i>Ratcliffe &amp; McNeill, 2011</i> <i>Gray et al, 2009</i> <i>Pichler, 2010</i>
	Product Box, Design the Box, Vision Box	Similar techniques used to capture the customer's vision of the new product. Creating the box first helps to identify the most exciting product features.	<i>Hohmann, 2006</i> <i>Gray et al, 2009</i> <i>Pichler, 2010</i> <i>Highsmith, 2009</i>
	Product Vision Board	A product vision board acts as the overarching goal guiding everyone involved in the development effort.	<i>Pichler, 2011</i>
	Remember the future, Trade Journal Review	Techniques to create a charter which can provide team members with a clear, elevating goal and spark an igniting purpose in a memorable way by first imagining the future.	<i>Hohmann, 2006</i> <i>Pichler, 2010</i> <i>Cohn, 2009</i>
Return on Investment	Impact Mapping, Effect Mapping	A strategic planning technique that prevents organisations from getting lost while building products and delivering projects, by clearly communicating assumptions, helping teams align their activities with overall business objectives and make better roadmap decisions.	<i>Adzic et al, 2012</i>
	Real Options	A set of principles with techniques which allow people to make optimal decisions within their current context. For each decision, identify the options available and identify the last point at which a decision can be made.	<i>Matts, 2010</i> <i>Matts &amp; Maassen, 2007</i>



Area	Intervention	Description	References
Product Backlog Prioritisation	20/20 Vision Game	A technique used to understand customer priorities and getting group clarity around which projects or initiatives are more important.	<i>Gray et al, 2010</i> <i>Hohmann, 2006</i>
	Story Mapping, User Story Mapping	A technique for how to build a better backlog that will help to more effectively explain a system, prioritize backlog items, and plan releases.	<i>Patton, 2008</i>
	Buy-a-feature, Bang for the Buck prioritisation, Divide the Dollar, \$100 Test Game, Prediction Market	A collection of prioritisation methods where participants invest imaginary money in backlog items. Can be done as a quick exercise or extended to an internal prediction market. It is used to prioritise features together with customers and stakeholders.	<i>Gardener, 2009</i> <i>Gray et al, 2010</i>
Requirements Elicitation and Communication	Customer show and tell, Show and Tell Game	A technique used to understand stakeholders' and customers' perspectives and to identify the most important artefacts created by a product.	<i>Hohmann, 2006</i> <i>Gray et al, 2010</i>
	Feature Injection	A practice which combines traditional analysis techniques and the agile technique Behaviour Driven Development to identify the business value delivered by a project without risk of introducing analysis paralysis.	<i>Adzic &amp; Matts, 2011</i> <i>McDonald &amp; Matts, 2009</i>
	Kano Analysis	A technique which is useful to group features based on their potential for impacting customer satisfaction when planning to develop a desirable new product. It supports identifying which features you are priorities and eases decisions to drop unwanted features, projects, and systems.	<i>Pichler, 2010</i> <i>Cohn, 2006</i>
	Me and My Shadow Game, The Apprentice Game	Examples of ethnographic research methods for identifying customers hidden needs and creating empathy for the customer experience	<i>Hohmann, 2006</i>
	Story writing workshop, User Story Writing	A practice for leveraging the creativity and knowledge of the team and the stakeholders to discover great stories. Properly conducted story-writing workshop can be a rapid way to write great requirements. A good story-writing workshop combines the best elements of brainstorming with low-fidelity prototyping.	<i>Cohn, 2004</i> <i>Pichler, 2010</i>

*Table 1 summarises the interventions identified in the literature review grouped by product owner responsibility area described in the theory section.*

Various different types of practices and techniques could be identified as possible interventions in order to provide an answer to the research question. These interventions are now further explained and compared to the Agile and Lean values and principles highlighted in the theory section of this paper.

#### **4.1.1. Product Vision**

The first area of product owner responsibilities is the establishment of a product vision. A first simple, but very popular, technique is the writing of an elevator pitch to distil and understand the core value proposition (Ratcliffe and McNeill, 2011). Writing the pitch as a group exercise will

focus everyone's attention on this value proposition and help drive consensus for the vision (Gray and Brown et al, 2010). Another technique which can help the product owner establish a vision is the creation of a product box (Highsmith, 2004). Inviting customer to create the box they would expect to find the product packaged and sold in will help the team to understand which features are most important and how the customer feels about the product. The *Product Box* exercise can be turned into a game to further bring out the creativeness as described in *Innovation Games* (Hohmann, 2007) and in *Gamestorming* (Gray and Brown et al, 2010).

A product vision board is a canvas where the product owner describes the product. A product vision board can be created quickly and then circulated for early feedback to iterate the product vision creation together with both stakeholders and customers (Pichler, 2010). The final set of interventions identified for establishing a product vision are called *Remember the Future* and *Trade Journal Review*. These techniques make customers and stakeholders imagine a future scenario where the product exists and is being successful in the market. It is a game-like activity based on cognitive psychology studies on how we think of the future. By imagining the future and thinking back, the future is no longer open-ended and new insights about the product vision are gained (Hohmann, 2007). The *Trade Journal Review* is based on the same principle. Here a development team works with stakeholders to write a trade journal review of the finished product while they are still only in the inception phase of a project (Cohn, 2009).

Using these techniques as interventions aimed at establishing a product vision is compliant with Lean-Agile principles. Overall there is a high degree of consensus decision making, customer focus, and visual management.

#### **4.1.2. Return on Investment**

The second area lists two interventions which are also found to be aligned with Lean-Agile principles. These are available for a product owner in order to maximise return on investment in product development. Impact mapping provides a method for visualising assumptions and connect why, who, how and what for a project (Adzic, 2012). *Real Options* is based on the mathematics behind

financial options and teaches us that options have value, options expire and we must never commit early unless it is known why. This provides a powerful Lean set of techniques for a product owner to increase return on investment (Matts and Maassen, 2010).

#### **4.1.3. Product Backlog Prioritisation**

Product backlog prioritisation is the third area where a product owner has an opportunity to increase customer value by introducing interventions to help making sure that the priorities are correct and customer value is delivered first. The *20/20 Vision Game* is again an activity which is straightforward to execute. It is a collaborative activity where customers and stakeholders are asked to rank features in order of importance where no two features can have the same priority. By running the exercise with different groups and assigning weighting the most valuable items will emerge to be prioritised for development (Hohmann, 2007). The *Story Mapping* exercise is a different way of describing the backlog. Here the requirements (the stories) are laid out in two dimensions where the x-axis is a time line for the order we need things in, and the y-axis is the perceived necessity of implementing the specific requirement. The result allows stakeholders and customers to prioritise only what should be prioritised and also to view and control the release plan (Patton, 2008).

*Buy a Feature* is a set of techniques which figures in various books. The basic concept is that customers and stakeholders are given a fixed amount of play money which can be used to buy features at different prices from the development team. This can be played in a short session as a game (Hohmann, 2007) also called the *\$100 Test* (Gray and Brown et al, 2010). The concept can also be developed into a permanent internal prediction type marketplace where customer and stakeholders invest in projects and features (Gardener, 2009).

These are all interventions which can be considered to fit well in with Lean-Agile practices. They all promote the key value and principles of customer collaboration.

#### 4.1.4. Requirements Elicitation and Communication

The final area of product owner responsibilities is requirements elicitation and communication. The *Customer Show and Tell* is a technique used to understand customers' perspectives and to identify what are currently the most important features of the product (Hohmann, 2007). Two other examples of ethnographic research methods for identifying customers hidden needs and creating empathy for the customer experience are the *Me and My Shadow Game* and *The Apprentice Game* (Hohmann, 2007). These introduce practices of shadowing customers in natural life situations where they are using the product or in the second case, using the product as an apprentice of an end user.

*Feature Injection* is an advanced practice which combines traditional analysis techniques and the Agile method *Behaviour Driven Development* to identify the business value delivered by a project without risk of causing analysis paralysis (Matts and Adzic, 2011). This practice makes use of traditional requirement gathering techniques such as *Object modelling* and *Entity-Relationship modelling* and makes best use of them in Lean-Agile development (McDonald and Matts, 2009). Another older and well established technique for understanding customer requirements is the *Kano*

*Analysis*. This technique is useful to group features based on their potential for impacting customer satisfaction when planning to develop a desirable new product. It supports identifying which features you need to prioritise and eases any decision to drop unwanted features or projects (Cohn, 2006).

The final intervention related to requirements elicitation and communication is a common practice recommended in Scrum environments called *Story writing workshops* (Pichler, 2010). It allows for leveraging the creativity and knowledge of the team and the stakeholders by describing the desired behaviour of the product together and writing it down as small stories. Properly conducted story-writing workshop can be a very rapid way to write great requirements. After this session the development team have a backlog to start work, but more importantly a solid understanding of what customers need. A good story-writing workshop combines the best elements of brainstorming with low-fidelity prototyping (Cohn, 2004).

These interventions are all suitable to apply within the Lean-Agile organisation as they focus on customer collaboration, visual management principles. They do not prescribe that requirements are created up front, and they promote, or complies with, the principle of software being delivered early to validate requirements.

## 5. Discussion

As seen in the *results* section the systematic search and literature review resulted in an initial list of practices and techniques recommended for any product owner seeking to effectively deliver real customer value. This provides an answer to the research question, which was the secondary purpose of this thesis. In this section further discussions around the findings are included. More importantly however is addressing the primary purpose, which is to account for insights gained from applying the research method and the recommendations for adjustments to future research methods. A number of recommendations for further study based on the outcomes of this scoping review are also included in this section.

First of all I found that much of the Lean-Agile literature is traditionally focused on software development and programming activities. As the

search expanded there were indications that practices and techniques meeting the requirement of suitable product owner interventions are more likely to be found outside the traditional Lean-Agile writings. Marketing, Strategic planning, User experience design, Innovation, Business Analysis, and Product management are all examples of important areas where it is likely that more relevant interventions can be found, even if this literature does not always appear to have any direct connection with Lean and Agile. A complete future systematic review needs to be multidisciplinary to be successful. I would recommend narrowing a similar search to focus on these specific areas as they overlap with the product owner responsibilities described in the *theory* section.

There is a strong indication that there are plenty of

both well established, as well as new emerging practices and techniques which a product owner can deploy as interventions in challenging development projects. A focus on identifying new tools for problems which has existed since the early days of system development is unlikely to be effective when answering the research question of this thesis. Existing tools already used in requirement elicitation and business analysis should be studied further and assessed against Lean and Agile values and principles. In older literature many of them are described in the context of sequential development methods and projects with up-front planning phases, but they can be adapted to work well as an intervention in Lean-Agile. Example is the *Kano Analysis* included in this review and *Feature Injection*, where usage of traditional requirements analysis techniques is recommended.

Another key learning is that categorising a practice or technique as Lean-Agile or not is flawed. This approach will need to be redesigned. For a future study I would recommend seeking to understand how practices and techniques can be used as interventions by applying knowledge of Lean-Agile values and principles. The reason behind this is that the practices and techniques are rarely conflicting with Lean-Agile principles. It is more often the process in which the practices are used that causes conflicts. You might not need a new toolkit; you just need to learn how to use your existing tools in new ways.

During the review I have noticed that it is sometimes in the interest of various influential external and internal groups to define fixed practices and processes for Lean-Agile teams. This is likely to conflict with the values of continuous improvement which is a key concept of both Lean and Agile.

- Consultancies need to define package solutions which can be sold as consulting services.
- Planning and tracking software vendors need to define artefacts and workflows to provide tools which are easily used and sold. This leads to the tool dictating how development is done and the team is restricted by the fixed workflows available.
- Within organisations there are often central management functions which controls

processes used throughout the organisation. These functions can find it difficult to adapt to an environment where multiple teams are experimenting with different practices and workflows.

This review has not provided any insights into which specific context the listed interventions are suitable in. Narrowing the question would be important for a future systematic literature review in order to focus on fewer possible interventions and gain better understanding of what works well, and what does not work well, under certain given circumstances.

Another recommendation for a future systematic literature review in the same area is to select one or a few related interventions. By doing a wide search, but looking at including multiple examples of how that particular intervention is used would produce a better recommendation of where and how to use many of the interventions identified. Adding more case studies to the literature search would improve the results even further.

The strategy followed to search and identify resources, which was creating a tree structure, can be recommended also for a larger systematic literature review. A large number of books and blogs would not have been identified in a regular database search as they don't necessarily contain the right keyword. Instead they were found because a subject matter expert has deemed them to be of value and added recommendations or references. It is important to note that the approach is time-consuming compared to a database search.

Adding blogs to the literature review was invaluable in growing the tree structure and finding new relevant resources. Creating a tree structure based on books only would require that books published over a long time period are included for a useful structure of recommendations and references to emerge. Blog posts on the other hand are posted on monthly and weekly basis and the hypertext web media encourages frequent cross-referencing. I would recommend the inclusion of blogs in any similar research. I would also suggest that Twitter (micro blogging) is evaluated further. On twitter the tree structure of subject matter experts exist as follower relationships and the publishing, referencing and commenting happens in hours and minutes.

As a last note, the tree structure that was grown with Lean-Agile subject matter experts and their publications very quickly assembled a network graph structure. The experts in the field frequently comment, reference and recommend each other's publications. The fact that a network graph

## 6. Conclusion

Finally we need to remember the first statement of the Agile manifesto: *Individuals and Interactions over Processes and Tools*. It dictates that people and how they interact is valued higher than the processes they follow and the tools they use (agilemanifesto.org, 2001).

In his blog post *The Decline and Fall of the Agilists* Jurgen Appelo (2009) explains how Agile with its values and principles outlines an evolutionary stable strategy. As such an Agile environment must never mandate the use of specific practices or techniques. That would cause the organisation to lose its ability to continuously meet its targets in the complex environment it exists in. This thesis has introduced a new way of thinking about product owner practices and techniques as interventions which can be used to create real value.

## 7. References

All online sources were last visited and available for web access in May 2013.

Adzic, G. (2012) *Impact Mapping*. Provoking Thoughts.

Agilemanifesto.org (2001) *Manifesto for Agile Software Development*. [online] Available at: <http://agilemanifesto.org>.

Anderson, D. (2003) *Agile management for software engineering*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference.

Appelo, J. (2009) *The Decline and Fall of Agilists*. *Agile Management - Noop.nl*, [blog] 5th Feb, Available at: <http://www.noop.nl/2009/02/the-decline-and-fall-of-agilists.html>.

Beck, K. (1999) *Extreme programming explained*. Reading, MA: Addison-Wesley.

Blank, S. (2006) *The four steps to the epiphany*. [Foster City, Calif.]: Cafepress.com.

emerges mitigates the problem of reproducibility mentioned in the *method* section. It is reasonable to believe that even though the structure will be recreated in different ways, the end result will look more or less identical.

As we put these interventions into practice we must acknowledge that:

- The interventions used must not compromise the core belief in the importance of people as human beings and individuals.
- In any Lean-Agile methodology the specific interventions are never to be considered mandatory as an evolutionary stable system adapts to context and changes over time.
- The interventions are only useful in the hands of a knowledgeable professional who understands how the tools can be applied to solve a defined problem in a Lean-Agile manner.

Cohn, M. (2012) *Agile Succeeds Three Times More Often Than Waterfall*. *Succeeding with Agile*, [blog] 13th Feb, Available at: <http://www.mountangoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>.

Cohn, M. (2005) *Agile estimating and planning*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference.

Cohn, M. (2010) *Succeeding with Agile*. Upper Saddle River, NJ: Addison-Wesley.

Cohn, M. (2006) *I Didn't Know I Needed That!*. *Succeeding with Agile*, [blog] 1 Feb, Available at: <http://www.mountangoatsoftware.com/articles/i-didnt-know-i-needed-that>.

Coster, N. (2013) *The Agile Business Gap*. *Brainrants blog*, [blog] 19 Apr, Available at: <http://www.brainmates.com.au/events/the-agile-business-gap-interview>.

- Fowler, M. and Highsmith, J. (2001) *The Agile Manifesto*. [online] Available at: <http://www.drdoobbs.com/open-source/the-agile-manifesto/184414755>.
- Gardner, J. (2009) *Innovation and the Future Proof Bank*. Chichester: Wiley.
- Gray, D. and Brown, S., et al. (2010) *Gamestorming*. Sebastopol, Calif.: O'Reilly.
- Highsmith, J. (2004) *Agile project management*. Boston: Addison-Wesley.
- Hohmann, L. (2007) *Innovation games*. Upper Saddle River, NJ: Addison-Wesley.
- Kniberg, H. (2007) *Scrum and XP from the Trenches*. C4Media.
- Kniberg, H. (2013) How to build the right thing. *Crisp's Blog*, [blog] 12th Feb, Available at: <http://blog.crisp.se/2013/02/12/henrikkniberg/how-to-build-the-right-thing>.
- Larman, C. and Vodde, B. (2009) *Lean Primer*. [e-book] Available through: Lean Thinking - Lean Primer <http://www.leanprimer.com/>.
- Larman, C. and Vodde, B. (2010) *Practices for Scaling Lean & Agile Development*. Upper Saddle River, NJ: Addison-Wesley.
- Linders, B. (2012) *Evidence of Success of Agile Projects*. [online] Available at: <http://www.infoq.com/news/2012/11/success-agile-projects>.
- Matts, C. (2010) Lean and Real Options. *Design Coach*, [blog] 7 Jan, Available at: <http://decision-coach.com/lean-and-real-options>.
- Matts, C. and Adzic, G. (2011) *Feature Injection: three steps to success*. [online] Available at: <http://www.infoq.com/articles/feature-injection-success>.
- Matts, C. and Maassen, O. (2007) *"Real Options" Underlie Agile Practices*. [online] Available at: <http://www.infoq.com/articles/real-options-enhance-agility>.
- Mcdonald, K. and Matts, C. (2009) *Feature Injection A Gentle Introduction*. [online] Available at: <http://agile2009.agilealliance.org/node/185/>.
- North, D. (2006) Outcomes Over Features - The Fifth Agile Value?. *Faster organizations, Faster software*, [blog] 28th Oct, Available at: <http://dannorth.net/2006/10/28/outcomes-over-features-the-fifth-agile-value/>.
- Patton, J. (2008) The New User Story Backlog is a Map. *AgileProductDesign.com*, [blog] 10 Aug, Available at: [www.agileproductdesign.com/blog/the\\_new\\_backlog.html](http://www.agileproductdesign.com/blog/the_new_backlog.html).
- Petticrew, M. and Roberts, H. (2006) *Systematic reviews in the social sciences*. Malden, MA: Blackwell Pub..
- Pichler, R. (2010) *Agile product management with Scrum*. Upper Saddle River, NJ: Addison-Wesley.
- Pichler, R. (2011) The Product Vision Board. *All Things Product Owner*, [blog] 10 May, Available at: <http://www.romanpichler.com/blog/product-vision/the-product-vision-board/>.
- Poppendieck, M. (2010) The Product Owner Problem. *Lean Essays*, [blog] 23rd Dec, Available at: <http://www.leanessays.com/2010/12/product-owner-problem.html>.
- Poppendieck, M. and Poppendieck, T. (2003) *Lean software development*. Boston, Mass.: Addison-Wesley.
- Poppendieck.com (2010) *Lean Software Development*. [online] Available at: <http://poppendieck.com/>.
- Ratcliffe, L. and Mcneill, M. (2011) *Agile Experience Design*. Berkeley, Calif.: New Riders.
- Ries, E. (2011) *The lean startup*. London: Portfolio Penguin.
- Schwaber, K. and Beedle, M. (2002) *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.
- Takeuchi, H. and Nonaka, I. (1986) The New New Product Development Game. *Harvard Business Review*, (January/February), p.285-305.
- The Standish Group (1995) *CHAOS. The Standish Group Report*. [report] The Standish Group.
- VersionOne (2013) *7th Annual State of Agile Development Survey*. State of Agile Development Survey. [report] VersionOne.

## Appendix A – Books by Agile Manifesto authors

- Beck, K. 2000. *Extreme programming eXplained*. Reading, MA: Addison-Wesley.
- Beck, K. and Fowler, M. 2001. *Planning extreme programming*. Boston: Addison-Wesley.
- Boehm, B. and Turner, R. 2004. *Balancing agility and discipline*. Boston: Addison-Wesley.
- Cockburn, A. 2001. *Writing effective use cases*. Boston: Addison-Wesley.
- Cockburn, A. 2002. *Agile software development*. Boston: Addison-Wesley.
- Cockburn, A. 2005. *Crystal clear*. Boston: Addison-Wesley.
- Fowler, M. 2004. *UML distilled*. Boston, MA [u.a.]: Addison-Wesley.
- Highsmith, J. 2002. *Agile software development ecosystems*. Boston: Addison-Wesley.
- Highsmith, J. 2004. *Agile project management*. Boston: Addison-Wesley.
- Hunt, A. 2008. *Pragmatic thinking and learning*. Raleigh: Pragmatic.
- Jeffries, R. 2004. *Extreme programming adventures in C#*. Redmond, Wash.: Microsoft Press.
- Jeffries, R., Anderson, A. and Hendrickson, C. 2001. *Extreme programming installed*. Boston: Addison-Wesley.
- Marick, B. 1995. *The craft of software testing*. Englewood Cliffs, N.J.: PTR Prentice Hall.
- Martin, R. 2009. *Clean code*. Upper Saddle River, NJ: Prentice Hall.
- Martin, R. 2003. *Agile software development*. Upper Saddle River, N.J.: Prentice Hall.
- Martin, R. and Martin, M. 2007. *Agile principles, patterns, and practices in C#*. Upper Saddle River, NJ: Prentice Hall.
- Schwaber, K. 2004. *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press.
- Schwaber, K. 2007. *The enterprise and Scrum*. Redmond, Wash.: Microsoft Press.
- Schwaber, K. and Beedle, M. 2002. *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.
- Schwaber, K. and Sutherland, J. 2012. *Software in 30 days*. Hoboken, N.J.: Wiley.
- Subramaniam, V. and Hunt, A. 2006. *Practices of an agile developer*. Raleigh, N.C.: Pragmatic Bookshelf.
- Sutherland, J., Solingen, R. and Rustenburg, E. 2011. *The power of Scrum*. North Charleson, SC: CreateSpace.

## Appendix B – Blogs and Websites of Agile Manifesto Authors

Author	Website / Blog Title	Available at:
Beck, Kent	Three Rivers Institute Thoughts on programming	<a href="http://www.threeriversinstitute.org/blog/">http://www.threeriversinstitute.org/blog/</a>
Beedle, Mike	Enterprise Scrum Blog	<a href="http://www.enterprisescrum.com/blog/">http://www.enterprisescrum.com/blog/</a>
C. Martin, Robert	8th Light Blog	<a href="http://blog.8thlight.com/uncle-bob/archive.html">http://blog.8thlight.com/uncle-bob/archive.html</a>
Cockburn, Alistair	Alistair Cockburn Blog	<a href="http://alistair.cockburn.us/Blog">http://alistair.cockburn.us/Blog</a>
Cunningham, Ward	WardCunningham's blog	<a href="http://dorkbotpdx.org/blog/wardcunningham">http://dorkbotpdx.org/blog/wardcunningham</a>
Fowler, Martin	Martin Fowler Agile Software Development	<a href="http://martinfowler.com/agile.html">http://martinfowler.com/agile.html</a>
Highsmith, Jim	Adaptive Imagineering	<a href="http://jimhighsmith.com/">http://jimhighsmith.com/</a>
Hunt, Andy	Pragmatic Programmer	<a href="http://blog.toolshed.com/">http://blog.toolshed.com/</a>
Jeffries, Ron	Xprogramming.com	<a href="http://xprogramming.com/">http://xprogramming.com/</a>
Kern, Jon	Technical Debt	<a href="http://technicaldebt.com/">http://technicaldebt.com/</a>
Marick, Brian	Exploration through Example	<a href="http://www.exampler.com/blog/">http://www.exampler.com/blog/</a>
Schwaber, Ken	Ken Schwaber's Blog: Telling It Like It Is	<a href="http://kenschwaber.wordpress.com/">http://kenschwaber.wordpress.com/</a>
Sutherland, Jeff	Scrum inc.	<a href="http://scrum.jeffsutherland.com/">http://scrum.jeffsutherland.com/</a>