The Product Owner role in Scrum

- A literature review and case study on Software Development

Master Thesis
SD101F19

Aalborg University
IT Design and Application Development

IT Design & Application Development
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
The Product Owner role in Scrum - A literature review and case study on Software Development

Theme:
Master Thesis

Project Period:
Spring Semester 2019

Project Group:
SD101F19

Participant(s):
Jonas Nielsen
Marie Jiříčková

Supervisor(s):
Jan Stage

Page Numbers: 44

Date of Completion:
June 7, 2019

Abstract:

This Master Thesis presents an exploratory study on the Product Owner role in Scrum on Software Development. It includes two papers - The first paper is a literature review examining separately research papers and guideline literature. First part reviews three main books on the Product Owner role and provides a list of five main aspects of looking at a Product Owner role. Second part reviews 30 articles and finds relevant information in 14. The result is a list of 40 challenges, split into eight categories and three areas. The combined result provides a richer Product Owner description comparing to the Scrum Guide in addition to identification of some gaps in the literature on the Product Owner role. Second paper is a case study with Product Owner practitioners. The results are based on nine semi-structured interviews from which a Grounded theory is build up. The theory identifies tasks and competencies of Product Owners. In conclusion we compare the results from the two papers to see if the information on the Product Owner role in the literature is comparable with the reality and conclude that it is, but there are some gaps where the literature does not cover the aspects of the Product Owner role fully.

Preface

This report presents a Master Thesis in Information Technology at Aalborg University, at the Master Programme (MSc) in IT Design and Application Development.

The report consists of four chapters and two academic papers in the CHI format, which are available in the appendix.

The project needs to be read in chronological order, as parts taken out of order could be mistaken without the context. Part of the data used in this Master Thesis, was gathered on 9th semester. The Summary is using the Harvard referencing method, meaning that after a source is used, the name and year of publication is present i.e. (name, year). If the authors name is written in the sentence, it will be followed by the year in parentheses. The research done, is subjective for the researchers, but we provide a research method section, and explanations, on how the results were gathered to make it more valid and reproducible.

Finally, we would like to thank our supervisor, Jan Stage, for the immense help and guidance throughout the project.

Aalborg University, June 7, 2019

Contents

List of Figures

Chapter 1

Introduction

Agile Software Development represents different methods all revolving around the same values found in the Agile Manifesto (Beck *et al.* , 2001). Agile Software Development methods and frameworks have evolved over time, and one of today's most popular frameworks is Scrum (Øvad & Larsen, n.d.), which is promoting lightweight processes and maximising of product value (Schwaber & Sutherland, 2017).

The agile framework of Scrum is consisting of three main roles: Scrum Master, the Development Team & the Product Owner, but most of the focus in prior research, in both academic articles and books, lies on the Scrum Master and the Development Team, and their perspective on development.

The Product Owner is described in the official Scrum Guide (Schwaber & Sutherland, 2017), as having primarily focus on the Product Backlog, the prioritisation of tasks and being in communication with the customer. They represent the customer and have some of the main decision making power and are the ones responsible for creating the most value for the customers in the product being developed (Schwaber & Sutherland, 2017). One of the main challenges that Product Owners have, is being in a role with an immense amount of responsibility and with a high requirement of a broad amount of knowledge, both technical and business, if to be followed and used as by the Scrum definition of the role (Kristinsdottir *et al.* , 2016).

The limited guidelines and lack of research on the Product Owner, rises some questions as to why, one of the most responsible roles in Scrum, is so vaguely described and why the role has not been explored more.

1.1 Key concepts

This chapter will explain the different concepts used throughout our project, including the concepts used in the papers. The concepts we will explain are Scrum and the Product Owner role in Scrum.

1.1.1 What is Scrum?

Scrum is a team-oriented agile process framework, meant for delivering products with the highest possible value. Ken Schwaber and Jeff Sutherland are the main creators of Scrum, which is based on the Agile Manifesto and its twelve principles for agile software development, which Schwaber & Sutherland also were involved in creating. Schwaber & Sutherland defines Scrum as the following:

"A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value." (Schwaber & Sutherland, 2017)

The framework is especially effective when it comes to iterative and incremental knowledge transfer between the team members (Schwaber & Sutherland, 2017). Scrum consists of a small team of people and is therefore not meant as a sole solution, for a larger and more complex team structure, which is found in many big software companies.

Scrum is based on empirical process control theory, also known as empiricism (Schwaber & Sutherland, 2017). It asserts that knowledge comes from experience and thereby making decisions based on what is currently known (Schwaber & Sutherland, 2017). There are three pillars in Scrum, which uphold every implementation of empirical process control: transparency, inspection and adaptation. These three pillars are embedded throughout the framework, in the tools and also in the role descriptions.

1.1.2 What is a Product Owner?

There is not much to the Product Owner in the official guideline for Scrum, the Scrum Guide. The Product Owner is the sole accountable for the Product Backlog, and is responsible for maximising the value of the product from the work of the Development Team (Schwaber & Sutherland, 2017). The Product Owner is the manager of the Product Backlog, which includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

Figure 1.1: Management of the Product Backlog (Schwaber & Sutherland, 2017)

The Scrum Guides description of the Product Owner is very short, and even though the focus is on the responsibility on the Product Backlog, it does not describe the immense amount of communication with clients, customers and stakeholders that it involves.

1.2 Problem Statement

Our aim is to address the limited research of the Product Owner role by exploring and investigating the following problem statement:

How does the literature further specify the Product Owner role's guidelines and are those descriptions comparable to the reality of Product Owners?

To answer the problem statement, we have formulated two research questions, each answered by one separate paper enclosed in the appendix.

Research question 1: *How does the literature describe the Product Owner role and how does it expand the description from the original source - the Scrum Guide?*

The first research question addresses the current academic and guideline literature's definition of the Product Owner role and how it might be different from the Scrum Guide (Schwaber & Sutherland, 2017). The aim is to uncover a richer description of the role, which would explain the role better comparing to the Scrum Guides description.

Research question 2: *What does the Product Owner role entail in a software development practise from the Product Owner's perspective?*

The second research question implore us to explore the practitioners in the role of Product Owners, how do they perceive their role and what are their experiences. We will uncover the reality of Product Owners by doing empirical research of the role.

Finally, the answers to the two research questions will provide us with grounds for answering our problem statement by comparing the answers.

Chapter 2

Contributions

This chapter will shortly present the two articles, which are the main parts of our Master Thesis. The two papers will be presented with a summary and the main findings.

2.1 Contribution 1

Marie Jiříčková and Jonas Nielsen. 2019. *The Product Owner role in Scrum development - A review of guideline and research literature* (2019). Computer Science, Aalborg University Denmark.

This contribution presents a literature review on the Product Owner role in Scrum. The goal of the study was to investigate the guideline and research literature, including the Scrum Guide, which is the official source of information about Scrum and the Product Owner role, and find out what is written about the role.

The paper consists of two reviews, systematic review which includes peer-reviewed academic articles and has a focus on the challenges of Product Owners, and narrative review which is concerned with three main books on the Product Owner role and their description of the role.

In the review of the academic articles, we identified 40 challenges. Which are split into eight categories in three different areas, these are the following:

- *Communication with different actors*
  - Insufficient communication with Development Team and upper management
  - Not shared expectations with clients

- *Organisational management*
  - Organisational distance
  - Task dependencies
  - Unclear definition of Product Owner role
  - Disrupting autonomy of the Scrum team
- *Working with requirements*
  - Missing and unsystematically changed requirements
  - Ineffective Product Backlog prioritisation

We reviewed 30 articles and found relevant information in 14 of them. In general, there was a low number of articles dealing directly with the Product Owner role. The amount of information about the Product Owner role per article was sparse, compared to other roles. This proves the limited academic attention given to the Product Owner role.

In the review of the books we identified five main aspects of a Product Owner role, which further contain 16 elements:

- *Responsibilities and tasks*
  - Prioritising of the Product Backlog
  - Product Backlog grooming
  - Stakeholder management
  - Being available for the team
  - Making a vision for the product
  - Being a leader for the team
- *Skills and characteristics*
  - Visionary & Doer
  - Leader & Team Player
  - Communicator & Negotiator
- *Maintaining good relationships*
  - Scrum Master relationship
  - Stakeholder relationship
  - Development Team relationship

- *Background for the Product Owner*
  - Business background
- *Common mistakes*
  - Not enough trust between management and Product Owner
  - Distance from the Team
  - Product Owner Committee

Our results show, that the book is in accordance with the Scrum Guide, but expanded immensely on the information on the Product Owner role.

We concluded by comparing the findings from the textbooks and the articles and identified holes in the literature where the textbooks did not provide any answers to challenges. In closing, we proposed more research on the challenges of the role and possible solutions.

2.2 Contribution 2

Marie Jiříčková and Jonas Nielsen. 2019. *Tasks & competencies of Product Owners in Scrum practise - A case study of Software Development with multiple stakeholders* (2019). Computer Science, Aalborg University Denmark.

This contribution presents a case study with nine Product Owners. The goal of this study was to investigate the Product Owner role in practise and provide a description of the role.

We conducted and nine semi-structured interviews, which we analysed with the use of Grounded Theory. That led to identification of the *Tasks and competencies* of Product Owners, as 12 concepts which are split into three different categories:

- Managing priorities
  - Managing different Stakeholders with different priorities
  - Effective Product Backlog Prioritisation
  - Balancing technical debt and progress
- Cultivating relationships
  - Building network
  - Being available for the team

  - Having good relationship and frequent communication with Scrum Master
  - Cooperating with upper-management
  - Maintaining good relationship with Stakeholders
  -

- Being visionary
  - Being autonomous
  - Having overview
  - Being adaptable and capable of fast decision making
  - Being unofficial leader

The *Tasks and competencies* of Product Owners are concerned with the different aspects of being a Product Owner, so how to use their main tool, the Product Backlog effectively, the importance of the relationships with the different actors surrounding the Product Owners and lastly, the influence of having a good vision of the product.

With the results we are enriching the description of the role and specifying the tasks that Product Owners face and the competencies they should possess, to maximise the value of the product they are creating.

Chapter 3

Method

In this chapter we introduce the methods used in our studies. We briefly present each method in chronological order as used in the articles. In each section we also discuss the methods strengths and weaknesses.

3.1 Literature review

A literature review can serve either as a background for further empirical research or it can be a piece on its own, mapping and making sense of existing literature, while providing a critical view on it (Templier & Paré, 2015). The importance of literature reviews is summarised by Webster & Watson (2002) as follows:

"A review of prior, relevant literature is an essential feature of any academic project. An effective review creates a firm foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists, and uncovers areas where research is needed." (Webster & Webster, 2002)

If a literature review is only a part of some academic work, it serves to identify gaps in research, provide theoretical foundation or position the new research and prove its contribution. It can also be a way of validating methods and approaches in a study (Kuziemsky & Lau, 2016). A literature review as a standalone piece provides a starting point for all researchers interested in that area. It can be a rich and valuable source of information. It provides an overview of a whole area as well as points to primary sources in that topic (Kuziemsky & Lau, 2016).

There exist many types of literature reviews. Templier & Paré (2015) classify four main types: narrative, developmental, cumulative and aggregative reviews. Each is built differently and suits to different objectives (Templier & Paré, 2015).

In our work we conducted a narrative review to review books and cumulative (also called systematic) review to review articles. Narrative reviews summarise present research on a topic of interest. They are often starting point for further research and can be helpful to researches with determining and refining research questions. In this type of reviews, researchers are very free in terms of methodology (Templier & Paré, 2015).

Cumulative reviews synthesise vast bodies of literature to draw overall conclusions. The difference between narrative and cumulative review lies in the methodological approaches and the diversity of primary studies included - the goal is to identify as many of relevant studies as possible (Templier & Paré, 2015).

Strengths and weaknesses

Strengths	Weaknesses
Gives overview over an area	Potential bias in literature selection
Identifies potential gaps in research	Subjective evaluations

Figure 3.1: Strengths and weaknesses of literature reviews

List strengths and weaknesses are shown in figure 3.1. Literature reviews, if conducted properly, can be a powerful information source for researchers or practitioners looking for current state of an area (Kuziemsky & Lau, 2016). They can help identify gaps in research and inspire new research (Cronin *et al.* , 2008). On the other hand literature reviews can be subjective and bias in selecting sources to include as well as in working with the data (Grant & Booth, 2009). There is no method to ensure all the relevant literature was considered (Grant & Booth, 2009).

We aimed to eliminate the presented weakness of bias or error in literature selection by presenting a transparent method we followed with sufficient detail on our search process as well as on inclusion and exclusion criteria. Further we applied more search techniques and conducted search individually. The subjectivity weakness we alleviated by conducting reviewing separately and then merging our results.

3.2 Case Study

In a case study, a selected case is examined qualitatively to provide detailed and in-depth knowledge in a concrete context (Lazar *et al.* , 2017). Therefore, in a case study, the focus is on the context-related knowledge in which an analysis object is

selected, for example a group of people, a local community or a company that is being investigated, and from the case it will be possible to generalise to a broader context (Flyvbjerg, 2014).

Bent Flyvbjerg (2014) mentions in a study, a case study in which the goal was to find out whether the use of organic solvents in companies could lead to brain damage of the employees. Based on several strategic methods, a company was selected, which would function as a case study. By focusing on one company, it was possible to gather information to a limited extent and subsequently generalise the findings from the case company. Subsequently, knowledge from the case study can be used as an argument that if the solvent led to brain damage in the case company, it will also lead to brain damage in other companies (Flyvbjerg, 2014). The knowledge collected during a case study can be generalised and used in a larger perspective to say something about similar conditions elsewhere (Flyvbjerg, 2014).

Strengths and weaknesses
The two primary strengths of case studies are their rich description and explanatory evidence of an area (Wynekoop & Conger, 1992). Case studies is a good way of gaining inside knowledge in specific cases or contexts (Flyvbjerg, 2014), as in our case, where we want information about the practise and experience from the Product Owners. One of the most common draw backs in case studies is their typical high cost of performing said case study and sometimes limited generalisability of the findings (Wynekoop & Conger, 1992).

Strengths	Weaknesses
Rich data	High cost
Deeper insights	Possibly time-consuming
Experience and explanatory based evidence	One-sided perspective

Figure 3.2: Strengths and weaknesses of case studies

The primary weaknesses of having high cost and being time-consuming conducting case studies were not present in our project, since the empirical studies are confined by the practicality of a deadline. To counter the weakness of having a one-sided perspective, we have conducted nine interviews with people from various industries, with different levels of experience, to gain a broader aspect of knowledge.

3.3 Interview method

One of the most effective ways to get information and qualitative data, is to interview stakeholders (Benyon, 2010). There is a range of different ways to conduct an interview. One of the most common used interview methods is the *semi-structured* interviews. They offer the option for a more open approach, where the interviewer has a prepared interview guide, but has the option to explore new topics, as the interview goes on (Benyon, 2010, Kvale, 2007). The interview guide becomes a helping tool, which structures the interview but gives it the format to find the latent knowledge of the interviewee (Kvale, 2007). The interviewer is studying and gaining some knowledge about the topic at hand, before the interview, but is keeping an open mind, and is going to let the interviewee control more of the interview (Kvale, 2007).

In our research we used the semi-structured interview method, as a way to gain more insight in an area we did not have the best knowledge of and to try to uncover new information in. The semi-structured interviews were guided by an interview-guide, where we had the option to ask questions, as the interviewee opened up about their experiences.

Strengths and weaknesses
One of the strong points for interviews is the ability to "go deep" and gain good insight in the chosen area, but interviewing is a skill, that the interviewer needs to train, in order to make better questions and getting the information needed (Lazar *et al.* , 2017). Compared to questionnaires, interviews are extremely flexible, and the amount of information gained can be substantial larger, and that can be a weakness, if too much unnecessary data is collected. Lastly interviews can open up for latent knowledge in the interviewee, but there can always be a difference in what the interviewee says he does, and what he actually does. ().

Strengths	Weaknesses
“Go deep” and gain insight	Interviewing is a skill
Flexible in information gain	Can gain unnecessary data
Can open up for latent knowledge	Disconnected from reality

Figure 3.3: Strengths and weaknesses of semi-structured interviews

To counter our weakness of missing the "skill" of interviewing, we have followed the methods, as close to the description, as possible. In order not to gain too

much unnecessary data we had a good structure in the interview guide. The last weakness was hard for us to counter and is considered a limitation in our research. A way to counter this, would be either to conduct observations of the Product Owners or getting information from another perspective e.g. Scrum Master role.

3.4 Grounded theory

Grounded theory gives a possibility to collect and analyse qualitative data in order to create theories 'grounded' in that data themselves (Charmaz, 2006). It was originally developed by sociologists Barney Glaser and Anselm Strauss who felt the need for a method which would allow for moving from data to theory (Willig, 2008).

The basic construct of Grounded theory is *categories*. They can be starting at a very low level of abstraction and evolve throughout the process, but they are never premade (Willig, 2008).

The process of identifying categories is called *coding* Willig (2008). There can be distinguished two main phases of coding - initial, which happens on low level and involves naming of each segment of data, and focused, selective phase which is concerned with synthesising and organising the data (Charmaz, 2006).

Other important constructs of Grounded theory present constant comparative analysis, or negative case analysis. First refers to researchers searching for similarities when creating categories as well as differences to break them into subcategories. The latter refers to searching for cases which do not fit into emerged category scheme and thus providing more depth and showing the complexity (Willig, 2008).

Strengths and weaknesses

Strengths	Weaknesses
Fitting for underexplored areas	Context specific
Provides insight into individuals experiences	Subjectivity

Figure 3.4: Strengths and weaknesses of Grounded Theory

The strengths and weaknesses of Grounded theory are summarised in the figure 3.4. One of the strengths is its fitness for areas which has not been explored much, because it does not work with hypothesis to be validated but it rather uncovers

the main concerns from the data itself without any prior assumptions (Hoda & Murugesan, 2016). Another strength is it provides good insight into perspective of individuals and their everyday life experiences (Siddique, 2016). The weakness of the method lies in the fact that it is context specific and it cannot be generalised to a larger sample (Siddique, 2016, Hoda & Murugesan, 2016). Finally, Grounded theory is subjective to the researcher applying it.

To reduce the affect the weaknesses we ensured that participants in our study have various backgrounds to include different perspectives and moreover each step of the process, when building up Grounded theory, was conducted by both researchers to increase validity and reduce bias.

Chapter 4

Conclusion

In this chapter we present the answers to our research questions and, finally, to our overall problem statement. Further, we present the limitations to our studies and propose future work.

4.1 Research questions

Research question 1:
How does the literature describe the Product Owner role and how does it expand the description from the original source - the Scrum Guide?

Our results from two literature reviews show that the description of the Product Owner role in the literature is far richer than the one given in Scrum Guide. The textbooks are defining the Product Owner role in five aspects, which contain 16 elements. These are connected to the tasks, characteristics, relationships and common mistakes. The articles provide 40 concrete challenges from the areas of communication, organisation and requirements work. The literature therefore defines the role in matter of tasks and responsibilities but also in matter of characteristics, the relation to other actors and the challenges related to the role.

Scrum Guide's description of the role is limited to the work connected to the Product Backlog, which corresponds to two elements out of 16 defined by the books. Thus, we can say that the literature expands the description from the original source greatly.

Research question 2:
What does the Product Owner role entail in a software development practise from the Product Owner's perspective?

In our findings from a case study with Product Owner practitioners we identified the twelve tasks and competencies the Product Owner role entails in software development practise. They can be split into the categories *Managing priorities*, *Cultivating relationships* and *Being visionary*. From the concrete tasks and competencies we point out *Effective Product Backlog prioritisation*, *Managing different Stakeholders with different priorities* and *Maintaining good relationship with Stakeholder* as these were the most common for participants in our study.

4.2 Problem statement

Problem statement:
How does the literature further specify the Product Owner role's guidelines and are those descriptions comparable to the reality of Product Owners?

To answer the problem statement, we conducted two studies on the Product Owner role: A literature review on two types of sources and a case study. We identified the elements and challenges of Product Owner role in the literature, from which only a small part was contained in the Scrum Guide. Further we identified the tasks and competencies of Product Owners in practice to be able to compare the information in literature with practise.

In our case study we did not identify any challenges, therefore we exclude them from our final comparison between literature and practise.

In figure 4.1 the elements found in literature are presented together with matching concepts identified in practise. We were able to find matching elements for 7 out of 12 concepts. It is also visible from the figure that four elements did not have a matching concept in practise, which might be given by their nature as with *Communicator and Negotiator* which is hard to compare with concrete concepts from tasks and competencies, or by the fact that the Product Owners in our study do not consider them relevant.

Aspects	Elements from literature	Concepts from practice
Responsibilities and tasks	Prioritising of the Product Backlog	Effective Product Backlog prioritisation
	Product Backlog grooming	Effective Product Backlog prioritisation
	Stakeholder management	Maintaining good relationship with Stakeholders
	Being available for the team	Being available for the team
	Making a vision for the product	
	Being a leader for the team	Being unofficial leader
Skills & Characteristics	Visionary & Doer	Being adaptable and capable of fast decision making
	Leader & Team Player	Being unofficial leader
	Communicator & Negotiator	
Maintaining good relationships	Scrum Master relationship	Having good relationship and frequent communication with SM
	Stakeholder relationship	Maintaining good relationship with Stakeholders
	Development Team relationship	Being available for the team
Background for the role	Business background	
Common mistakes	Not enough trust between management and Product Owner	Cooperating with upper-management
	Distance from the Team	Being available for the team
	Product Owner Committee	

Figure 4.1: Relation between elements from literature with concepts from practise

The concepts with no equivalent elements are visible in figure 4.2. These are the tasks and competencies which do not have a major equivalent in the literature we reviewed and therefore point to an absence of description of part of the task and competencies of a Product Owner in literature.

Concepts from practise missing in the literature
Managing different Stakeholders with different priorities
Balancing technical debt and progress
Building a network
Being autonomous
Having an overview

Figure 4.2: Concepts from practise missing in literature

4.3 Key limitations

We will now go through the key limitations, which we have identified in our work and described below.

Qualitative bias
All methods used in our work are qualitative, which enhances the weakness they have all in common - researcher bias and the question of objectivity of results. To overcome this limitation, we have been transparent with our research design to increase replicability and validity of our results. For the same reason both authors also conducted all research activities, first separately, and then compared and combined results.

Product Owner's perspective
All conducted interviews are with Product Owners, except for one Product Manager. If we had included other roles, we would get different perspectives and the results could have differ. What partially alleviates this limitation is the other sources we used on getting knowledge on the Product Owner role, apart from interviews, including research articles, which often had focus on other roles in Scrum.

4.4 Future work

To expand on our research, we would suggest several directions for future empirical studies. First, to investigate the gaps in research identified in our first paper where we pointed out challenges in areas of Product Owner role, which has not been described, suggesting there is more to the role than what is in the guideline literature. Further, we identified tasks and competencies of Product Owners in practise, some of which has not been described in the guideline literature which also points to unexplored areas of the role.
Further a study on possible solutions to identified challenges would bring valuable insight for practitioners. Lastly, additional empirical studies should be conducted on the Product Owner role from the perspective of the other Scrum roles, to ensure more rich data. Longitude empirical studies in multiple companies, with observations and interviews, will provide valuable insights on how the Product Owners perceive their work, in comparison with the observation results.

Bibliography

Beck, Kent, Beedle, Mike, van Bennekum, Arie, Cockburn, Alistair, Cunningham, Ward, Fowler, Martin, Grenning, James, Highsmith, Jim, Hunt, Andrew, Jeffries, Ron, Kern, Jon, Marick, Brian, C. Robert, Martin, Mellor, Steve, Schwaber, Ken, Sutherland, Jeff, & Thomas, Dave. 2001. *Manifesto for Agile Software Development*. https://agilemanifesto.org/.

Benyon, David. 2010. *Designing interactive systems: a comprehensive guide to HCI and interaction design*. 3rd edn. Pearson Education Limited.

Charmaz, Kathy. 2006. *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. Sage Publications.

Cronin, Patricia, Ryan, Frances, & Coughlan, Michael. 2008. Undertaking a literature review: a step-by-step approach. **17**, 38–43.

Flyvbjerg, B. 2014. *Fem Misforståelser om Casestudiet, in Brinkmann, S. og Tanggaard, L. (red.) Kvalitative metoder: en grundbog. 1. i-bog.* Hans Reitzels Forlag.

Grant, Maria J., & Booth, Andrew. 2009. A typology of reviews: an analysis of 14 review types and associated methodologies. **26**, 91–108.

Hoda, Rashina, & Murugesan, Latha K. 2016. Multi-level agile project management challenges: A self-organizing team perspective. **117**, 245–257.

Kristinsdottir, Sigurhanna, Larusdottir, Marta, & Cajander, Åsa. 2016. Responsibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study. *Pages 3–16 of: Lecture Notes in Computer Science*. Human-Centered and Error-Resilient Systems Development, vol. LNCS-9856. Springer International Publishing.

Kuziemsky, Craig, & Lau, Francis. 2016. *Handbook of EHealth Evaluation : An Evidence-based Approach*.

Kvale, Steinar. 2007. *Doing interviews*. Sage Publications Ltd.

Lazar, Jonathan, Feng, Jinjuan Heidi, & Hochheiser, Harry. 2017. *Research Methods in Human-Computer Interaction - Second Edition*. Morgan Kaufmann.

Schwaber, Ken, & Sutherland, Jeff. 2017 (November). *The Scrum Guide*. http://www.scrumguides.org/scrum-guide.html.

Siddique, Lubna. 2016. Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers? Perspective. **6**, 120–135.

Templier, Mathieu, & Paré, Guy. 2015. A Framework for Guiding and Evaluating Literature Reviews. **37**.

Webster, Jan, & Webster. 2002. Analyzing the past to prepare for the future: Writing a literature review. **26**, R13.

Willig, Carla. 2008. *Introducing Qualitative Research in Psychology*. Maidenhead: Open University Press.

Wynekoop, Judy L., & Conger, Sue A. 1992. A review of computer aided software engineering research methods.

Øvad, Tina, & Larsen, Lars Bo. The prevalence of ux design in agile development processes in industry.

Appendix A

Appendix

A.1 The Product Owner role in Scrum development - A review of guideline and research literature

Marie Jiříčková and Jonas Nielsen. 2019. *The Product Owner role in Scrum development - A review of guideline and research literature* (2019). Computer Science, Aalborg University Denmark.

A.2 Tasks & competencies of Product Owners in Scrum practise - A case study of Software Development with multiple stakeholders

Marie Jiříčková and Jonas Nielsen. 2019. *Tasks & competencies of Product Owners in Scrum practise - A case study of Software Development with multiple stakeholders* (2019). Computer Science, Aalborg University Denmark.

The Product Owner role in Scrum development - A review of guideline and research literature

Marie Jiříčková
Jonas Nielsen
mjiiko17@student.aau.dk
jnie14@student.aau.dk
Aalborg University
Aalborg, Denmark

ABSTRACT

The use of agile methodologies, and Scrum framework in particular, is growing rapidly in the Software Development industry. This of course correlates with the growing number of publications on these topics. This article concerns itself with a vital part of Scrum – the Product Owner role and the literature published on it. A systematic literature review of peer-reviewed articles is presented, which results into an overview of challenges Product Owners face. Furthermore, the main textbooks are reviewed and the result is an enriched description of the Product Owner role comparing to the Scrum Guide. Together the two reviews create a fuller description of the Product Owner role, building on information from up to date available literature as well as point to areas deserve exploration.

KEYWORDS

Product Owner, Scrum Role, Challenges, Literature review, Software Development

1 INTRODUCTION

The term *Scrum* was first introduced in a article from 1986 by Takeuchi and Nonaka which was describing the innovative practices of Japanese companies [13]. Later it served as an inspiration to Jeff Sutherland and Ken Schwaber who refined the Scrum framework as we know it now [13]. Scrum is currently the most used of agile frameworks [19]. Scrum builds on simple practices and management work products, customer participation, openness and self-organising teams [13]. Though lightweight and simple to understand, Scrum is difficult to master, as it is disclaimed in the beginning of the Scrum Guide [18], as Scrum is a framework lacking methodological processes to follow.

Scrum is based on three pillars: Transparency, Inspection & Adaptation [18], and it is these empirical pillars that define Scrum. The *Transparency* pillar is about the common understanding and language used in the project, so the Scrum Team is on the same page and share the same definition of "Done" [15]. The pillar of *Inspection* is about the iterative nature of Scrum and that the artefact and processes must be inspected frequently, but not disturbing, so undesirable variances can be detected under a Sprint [18]. The *Adaption* pillar is the awareness of changes and reacting to them, for maximising the value of the product being created [18].

Scrum consists of three roles: The Development Team, the Scrum Master and the Product Owner. These are equally responsible for the product but have various responsibilities in between [18]. The Scrum Master is a facilitator of the process, both for the Development Team and the Product Owner. He makes sure, that the process is followed and adapts it if needed [18]. Development team is responsible for delivering releasable increments of the product and the Product Owner is managing the Product Backlog. His ultimate responsibility is to maximise the value of the developed product [18].

The Scrum Master and Development Team has received attention in academic papers from various perspectives. The same cannot be said about the role of Product Owner which is of vast importance. Therefore, will this article be focusing on the role of Product Owner and will examine the information available on the role in the literature. The problem statement is as follows:

How does the literature describe the Product Owner role and how does it expand the description from the original source - the Scrum Guide?

To answer this question we, first, review books written on the Product Owner role and, second, review the academic papers connected to our research. The structure of the paper is the following: First we summarise the work related to our research. Then we explain the methods used when conducting our literature reviews. Next, we present the findings from book reviewing and then separately findings from academic papers review. In the discussion we relate our two types of results to each other and finally we present a conclusion.

2 RELATED WORK

In this section, we present an overview of the related work. To find the reviews, we used Aalborg University's set of databases and we searched with a combination of the keywords: Scrum, Product Owner, Agile, literature review and systematic review. We limited our results to the field of Computer Science and English language. This resulted into identifying four articles presented below.

Two of the literature reviews we found are concerned with the adoption of Scrum in Software Development. The first article [14] is investigating the problems in adoption of Agile-Scrum methodologies and had two research questions: "*What are the main problems for adopting agile methodologies?*" and "*What are the main problems for adopting Scrum?*". Out of the 269 papers initially found, the number is reduced to 27 full papers relevant for their research. In

these 27 papers they found four areas of problems in the adoption as an answer combined for both questions, namely: Organisational aspects, People, Project and Process. They conclude that these problem areas must be addressed in order to improve the adoption level in the organisations.

The second article [21] dealing with adoption is researching the following: *Which factors models related to the adoption of ASD are used in the software industry? Models found will be used to identify relevant factors* and *What factors are common to all the identified models? Factors will be classified based on their presence in all the identified models.* The review studies 353 papers and finds 10 relevant for their research question. Findings confirm the same four categories as the previous article: Organisational aspects, People, Project and Process. They conclude that different factors, in the four different categories must be supported, to avoid the issues in adopting agile methods.

Article [10] is making a systematic literature review on agile requirements engineering practices and challenges. Their goal was to map the evidence available, of the requirement engineering practises and their adoption and the challenges faced by the development teams. They conducted a systematic literature review on 543 studies and identified 21 papers relevant. They identified 17 practises of agile requirements engineering and 13 challenges. They concluded that agile requirements engineering as a research context needs more attention.

The last article found is a systematic literature review on quality criteria for agile requirements specification [7]. Their goal was to investigate what quality criteria for assessing the correctness of written agile requirements exist. They found 630 research papers and 16 was relevant. They categorised and analysed the quality criteria and compared them with those from traditional requirements engineering. They conclude with some recommendations for practitioners for quality assessment of agile requirements specifications.

3 METHOD

A literature review can be approached in many ways, we followed the six general steps Templier & Paré [24] suggest for any literature review: formulating the problem, searching the literature, screening for inclusion, assessing quality, extracting data and analysing and synthesising data. Further in this section we present in detail how we applied those steps for both types o literature: research articles and textbooks, which where each approached separately due to their different nature.

3.1 Method: Textbooks

In this part we present a narrative literature review. Comparing to the other types, the narrative literature review falls more under qualitative interpretations of data and does not aim to provide any generalisations or accumulations of reviewed sources, but rather to summarise and synthesise what has been written [12].

Problem formulation:
How does the literature define the Product Owner role and how does it expand on the description from the Scrum Guide?

Searching for literature
In order to identify the relevant books, we took these actions:

- General search through basic search engines (e.g. Google)
- Search through Aalborg University's library database which merges all databases available to Aalborg University researchers
- Backward search on sources cited by articles focusing on the topic of Product Owner role
- Contacted a researcher who published in this area to help us identify any literature we might have overlooked as well as give us feedback on the literature we found

The search was conducted by each researcher separately, in order achieve a higher validity by reducing bias and chances of human error when searching.
Screening for inclusion and assessing quality
The inclusion criteria for books were set as follows:

- The book was cited by a peer-reviewed article or has a connection to any of the official sources of knowledge on Scrum - Scrum.org and Scrum Alliance
- Product Ownership has to be a main focus
- The book is in English

The screening for inclusion was done together by both researchers. The books identified through our process three to review were following:

- *Agile Product Management with Scrum - Creating Products that Customers Love* by Roman Pichler [17]
- *SCRUM Product Ownership - Balancing Value from the Inside Out* by Robert Galen [5]
- *The Professional Product Owner* by Don McGreal and Ralph Jocham [15]

Extracting data
In order to extract data from the books, we set up a list of different categories of information, that we were searching for. These were organised into a table which helped us in comparing the information between books. Each book was read by both authors and then the result were compared. If there was a difference, a compromise was found based on a discussion.

Analysing and synthesising data
After extracting the data from each book we combined it and organised it into the *different aspects of Product Owner role*, which became our main category. Each aspect consists of several elements, as visible in the figure 2. This figure also shows which elements were presented by which of the reviewed books. For comparison the figure also includes the Scrum Guide.

3.2 Method: Research articles

In regards to the research articles we conducted a systematic literature review. We chose this approach because of the limited number of papers directly dealing with Product Owner role. In response to that we wanted to find all information available also in papers related to our topic.

Problem formulation:
What does the Product Owner role entail in a software development projects?

Searching for literature
In order to identify the articles we combined data sources and search approaches as advised [24]. We did not limit ourselves to only "top" sources of literature in the field inspired by Webster & Waston [25], who argue against this kind of 'elitism' in research. The search string we used had two parts. One was specifying the process - *Scrum* or *agile*, other one was concerning the role - *Product Owner*. When searching we also applied a pre-made filters of respective databases to limit our results to the field of Computer Science and to results in English published after 1990.

We conducted three types of searches:

- Search through Aalborg University's library database, which merges all databases available to Aalborg University researchers
- Specific search via EBSCO Host, JSTOR, ProQuest and Taylor & Francis databases to eliminate the chance of overlooking any potential articles
- Backward search on our already found articles (conducted on articles which were identified as relevant in the next step)

The search was done separately by each researcher to achieve higher level of validity by reducing bias and chances of human error when searching.

Screening for inclusion and assessing quality
In order to identify articles, which we would review we established inclusion criteria and quality assurances as follows:

- Article has to have a connection to our topic by concentrating on some problem connected in Scrum or agile close to the Product Owner role.
- Article has to be in English
- Article has to be published from 1990 and after
- Article has to be peer-reviewed

The evaluation if articles will be included was done first by each researcher separately on the articles he or she found, based on the title, key words and abstract of articles. The combined result was 36 articles, which we screened for inclusion for a second time. This time it was done together to inspire discussion and ensure consensus. After this nine articles were excluded. Furthermore we conducted backward search on articles found fitting to review which resulted into one more article added to our list.

A month later we conducted the same procedure of searching for literature and screening for inclusion to make sure we did not overlook any relevant article. We also added a variation to our search string by alternating *Product Owner* for *Product Manager* or *Project Manager*. The result was two more articles included. In total we therefore reviewed 30 articles. From these 30 reviewed articles, we found relevant data in 14 of them, these you can see in the figure 6 in the appendix.

Extracting data
Each article was read and coded by two researchers to avoid bias, judgement errors and to increase validity [24]. For the data extraction we used QSR's software, NVivo 12. When coding the articles, we worked with the concept of 'in vivo' categories, which help researchers to avoid prior assumptions and judgement by emerging directly from the data, because they use the terminology directly from the sentences being coded [1]. The data to be coded was identified based on our problem formulation. Each article with its codes from both researchers was then examined by both authors to unify coding via discussion.

Area	Total number of papers	Total number of references
Communication with different actors	8	13
Organisational management	12	28
Working with requirements	9	15

Figure 1: Areas where Product Owner's challenges occur

Analysing and synthesising data
When analysing the data, we went through our codes and as a main category, which related to most of our codes, identified *Challenges of Product Owner*. We revisited our codes and refined them towards our main category. We ended up with a list of 40 concrete challenges that Product Owners face. These challenges were sorted into categories and further into three main areas, where they occur. These areas can be seen in figure 1 together with the number of articles containing references pointing to a particular area and the total number of references in those articles connected to the area. The full structure of categories and challenges can be seen in the figure 3, in the Results section.

4 RESULTS: TEXTBOOKS

The results in this section are illustrated in the figure 2, where the different elements of Product Owner role are aggregated from the books and split into aspects. From the figure it is also visible what each author includes in his work on the Product Owner role. This section will describe the elements of Product Owner role.

4.1 Responsibilities and tasks

This aspect contains the *responsibilities and tasks* of the Product Owner and following are described the elements it contains.

Prioritising of the Product Backlog
What all the books and Scrum Guide have in common is the Product Backlog, even though the amount of detail on how to estimate and prioritise the Product Backlog varied through the sources. Nevertheless all the three books [5, 15, 17] have a separate chapter on

Aspects	Elements	Pichler	Galen	McGreal & Jocham	Scrum Guide
Responsibilities and tasks	Prioritising of the Product Backlog	x	x	x	x
	Product Backlog grooming	x	x	x	x
	Stakeholder management	x	x	x	
	Being available for the team	x	x	x	
	Making a vision for the product	x	x	x	
	Being a leader for the team	x	x	x	
Skills & Characteristics	Visionary & Doer	x		x	
	Leader & Team Player	x	x	x	
	Communicator & Negotiator	x		x	
Maintaining good relationships	Scrum Master relationship	x	x	x	
	Stakeholder relationship		x		
	Development Team relationship	x	x	x	
Background for the Product Owner	Business background	x			
Common mistakes	Not enough trust between management and Product Owner	x		x	
	Distance from the Team	x	x	x	
	Product Owner Committee	x	x		

Figure 2: Aspects and elements of the Product Owner role and their occurrence in the reviewed books

the Product Backlog and how to work with the artefact.

The Scrum Guide has the Product Backlog as the main focus of responsibilities, when describing the Product Owner, but it is still short and not entirely concrete on what it involves for the Product Owner [18], other than he is the sole responsible for it.

It is visible from the figure 2 that these are the only two sub-topics (*Product Backlog grooming* and *Prioritising of the Product Backlog*) which are common for all four sources considered, which confirms the vast importance of them.

Product Backlog grooming
All the books are describing the grooming process, how the new items are discovered and described, and how existing ones are changed or removed. Pichler writes:

"*Like a garden growing wild when left unattended for too long, the product backlog becomes unwieldy when it's neglected. The backlog needs regular attention and care; it needs to be carefully managed, or groomed.*" [17]

He goes into detail, how to discover and describe new items, how to re-prioritise the Product Backlog, so the new important items now are on top, and high-priority items are prepared for the upcoming Sprints [17].

The grooming process is something that the Scrum Guide also writes about, though they call it Product Backlog refinement.

Stakeholder management

Galen and McGreal & Jocham describe the responsibility the Product Owner has, when it comes to the communication with the stakeholders, mainly meaning the clients and customers, as he needs to include the stakeholders in as many meetings as possible, while getting feedback from them [5, 15].

Pichler specifically writes about stakeholder management and involving the customers and working with the users:

In addition to customers and users, product owners should involve other stakeholders, such as representatives from marketing, sales, and service, early and regularly by asking them to attend the sprint review meetings. The meetings allow the representatives to see the product grow, to interact with the Scrum team, and to share questions, concerns, and ideas. [17]

McGreal & Jocham also provide a tool, which helps to involve the stakeholders into the project [15].

Being available for the team
The better the Product Owner is at being available for his team, the more value he can bring [15]. McGreal & Jocham write about being available for the Team, so if questions arise, as an example, after a Daily Scrum meeting they can get answered immediately by the Product Owner and thereby be more effective [15].

Pichler also writes, that it is important for the Product Owner to see the team as much as possible, and goes further and suggest a one-hour-rule, where the Product Owners are in the same room as the team [17].

Making a vision for the product
How to make a vision for the product and how to adapt the vision in the project, is something all the books are talking about. McGreal & Jocham in their chapter about vision provide several tools on to how to produce and maintain a vision, such as using a Business Model Canvas, using vision statements, or having technical strategy [15]. Pichler writes that the minimal viable product can help enhance the vision and give it some direction [17].

Being a leader for the team
The Scrum Guide does not emphasise or give any attention to the fact, that the Product Owner acts as a leader for the Team, on the contrary it points to the team being self-organised [18], even though it is one of the important parts of being a Product Owner [5]. Galen goes into detail, how being a good leader, can help in understanding your team and in "championing your team" [5]. Which he further explains that the Product Owner is the voice of the team and in taking leadership the Product Owner can challenge the team to perform better [5]

Pichler writes about being a *Leader and Team Player* for the team. This is in connection with the vision of the product, as the Product Owner will have an easier time with guiding the team, if they trust him as a leader and team member [17].

4.2 Skills & Characteristics

The aspect *Skills & Characteristics* is dealing with the skills and traits a Product Owner should have. Both Pichler and McGreal & Jocham pay attention to the *Skills & Characteristics* in their books. This aspect shares some similarities with *Responsibilities and tasks*, as they are logically connected.

Visionary & Doer
The Product Owner should be cable to carry out the vision he sets upon the product, as well as encourage innovation [17]. The Product Owner needs to be an initiator and committed to his course of creating maximum value for the product [15].

McGreal & Jocham outline that a Product Owners personality affects the outcome and here point to the entrepreneurial characteristic, to be beneficial, as they have some control over the budget and often a more clear vision of the product [15].

Leader & Team Player
The Product Owner should have a dual nature of being the leader for the Team, but also being a Team Player:

"*By no means should the product owner dictate decisions, yet at the same time neither should the product owner be indecisive or employ a laissez-faire management style.*" [17]

The Product Owner should strive for giving the Team the necessary guidance and for seeking consensus in the decision making. [17].

Communicator & Negotiator
The Product Owner needs to be effective at communicating, both his vision, but also when talking to stakeholders and the Team [5, 15, 17]

The Product Owner is the voice of the customer, and is communicating their needs and requirements, acting as a bridge to the Development Team. This can sometimes mean saying no to the customer and negotiate [17].

4.3 Maintaining good relationships

This aspect deals with the relationships that the Product Owner is part of and how to maintain them.

Scrum Master relationship
The relation to the Scrum Master is something all the books have covered. Pichler writes, that the Product Owner is primarily responsible for the "what", as in creating the visioned product, whereas the Scrum Master is responsible for the "how", as in using Scrum effectively and to accomplish this, the two need a good relationship [17]. Galen suggests that the great Product Owners need to establish a partnership with the Scrum Masters, as the two roles are the central leadership within the Development Team, and one is externally and other internally focused [5]. McGreal & Jocham support this, as they express that the Scrum Master is there to help the Product Owner maximise the value of the product and this is done with the Scrum Master coaching the Development Team and then relaying the information back to the Product Owner [15].

The Scrum Guide describes the relationship of the two roles but only from the Scrum Master perspectives, in relation to the Scrum Master and his tasks. It agrees with the point that the Scrum Master helps to maximise the value of the product and is a facilitator of the process to help the Product Owner [18].

Stakeholder relationship
Galen writes that the Product Owner should engage with the stakeholders as often as possible, to make sure, that their requirements are up to date. Furthermore, that the stronger the relationship the Product Owner and stakeholders have, the better information and value for the Product Owner [5].

McGreal & Jocham or Pichler are not focusing individually on the relationship with the stakeholders, as stakeholder management is incorporated in some of the tasks. The Scrum Guide is in general very brief on stakeholder management and does not directly connect it with the Product Owner role or his responsibilities.

Development Team relationship
The relationship to the Development Team is a subject all the books describe in detail. Pichler writes [17] that "*all members must form a close and trusting relationship and work as peers*", as it would strengthen their work and bring more value to the product. The Product Owner should preferably be located with their teams, so they can have as many face-to-face interactions as possible [5, 17].

Galen explains [5] that the Product Owner should "*Champion the*

Team", and thereby know his team and support them where needed. McGreal & Jocham outlines [15] that the Product Owner should include the Development Team in much of the tactical work, like Product Backlog, refinement and acceptance criteria.

4.4 Background for the Product Owner

This section contains only the element of *Business background* for the Product Owner.

Business background
Pichler and McGreal & Jocham are concerned with the background of the Product Owner. Pichler writes [17], that a Product Owner benefits from a business oriented background, like Product or Project Manager. McGreal & Jocham describes [15] the optimal Product Owner, as an entrepreneur who owns the product and has control over the budget [15].

4.5 Common mistakes

The aspect of *Common mistakes* deals with the mistakes often seen in connection with the Product Owner role.

Not enough trust between management and Product Owner
The Product Owner needs trust from the management to make the best performance [17]. If the Product Owner has to consult their superior in every major decision, it can cause setbacks and slowing in work for the Development Team, if they can not get answers quick enough [15, 17].

Distance from the Team
The more time the Product Owner spends with the team and the more time he has, to answer face-to-face questions, the better for the relationship and value which can be brought to the product [15]. The Team needs to be located close to their Product Owner, to make sure the information flow remains efficient [17].

Product Owner Committee
One of the common mistakes is to have a Product Owner group or committee, but not having a single person responsible for the product. This can lead to endless meetings with conflicting interests and politics [17]. Pichler calls this *death by committee* [17]. The committee is possible, as long as there is a single person with the responsibility and authority to make final decisions [17].

5 RESULTS: RESEARCH ARTICLES

After analysing and synthesising extracted data we identified three areas in which Product Owners face challenges: *Communication with different actors*, *Organisational management*, *Working with requirements*. In figure 3 the full structure of our categories and under-lying challenges can be seen (column "articles" refers to the particular articles containing the challenge).

5.1 Communication with different actors

This area points to challenges in categories connected to lacks in communication within different relationships and situations.

Insufficient communication with Development Team and upper management
This category is about the challenges in communication Product Owner has with the Development Team and Scrum Master on one side and upper management on the other. It is important the Product Owner has a good connection with the team and the communication is open and frequent, in order to create as much value as possible, as they should be in contact daily [4, 11, 16]. The clients often struggle to provide clear acceptance criteria [9] together with the Product Owner not communicating these clearly [2] it can lead to the Team having a hard time estimating User Stories and implementing them [9].

Also if the Product Owners relationship and communication with the upper management is not set correctly, the team can be affected for instance when their requests are not heard or they get overloaded with work, because the Product Owner could not shield them from the pressure on results developed by the superiors [23].

Not shared expectations with clients
The Product Owner needs to agree with the stakeholders on the requirements, in order to be able to pass them on and prioritise them effectively [3]. Assuming the client's wishes can be dangerous [3].

The Product Owner can also face a situation with stakeholders having various needs [9]. The Product Owner has to manage the various expectations of the different stakeholders' effectively and it is necessary that his decisions are respected [22].

5.2 Organisational management

Area *Organisational management* groups together challenges in categories which are caused by flaws in setting in the Product Owner's organisation, for instance responsibilities not being clearly divided or overworking Product Owners with too many tasks.

Organisational distance
This category points to the issue of distance. If the Product Owner and his team do not spend most of the time physically at one place, the core value of Scrum [18] - face to face communication is endangered. Small issues that are not resolved daily can grow rapidly and cause bigger problems in the long run [2, 16].

Task dependencies
Task dependencies are something companies are trying to avoid as much as possible [1] but the more complex the project is, the more inter dependencies it will naturally have, and being able to see them and understand them is of a key importance [3]. The Product Owner needs to watch out for dependencies when prioritising and designing requirements [1] in order to avoid substantial rework [3] or Sprint cancellation [9].

Unclear definition of Product Owner role
These challenges connects to the issue of a bare description of the role in the Scrum Guide [18] because the Product Owner role is often much broader [8, 9, 11]. Often when organisations are going

	Area	Category	Challenge	Articles
Challenges of Product Owner	Communication with different actors	Insufficient communication with Development Team & upper management	Product Owner not documenting request properly for the team	4
			Unclear communication between Product Owner and team	10, 4
			Lack of communication between Product Owner and Scrum Master	10
			Difficult relationship with team	11
			Unclear acceptance criteria	7
			Bad communication within the team	8
			Product Owner not having a strong relationship with upper management	14
		Not shared expectations with clients	Not shared understanding of requirements	14
			Stakeholders have different expectations	10, 13
			Second-guessing the client	3
	Organisational management	Organisational distance	Coordinating meetings	2
			Delays and unresolved issues	2
			Physical distance of desks	2, 10
			Product Owner unavailable	12, 5
		Task dependencies	Requirement dependencies	1,3,7
			Inter-dependencies on other systems	3
		Unclear definition of Product Owner role	Unclear responsibility for management activities	7
			Unrestricted authority to make decisions in projects	13
			Product Owner not the actual decision maker	12
			Not clearly defined responsibilities	6
			Challenge of being an indirect leader	8
		Disrupting autonomy of the Scrum team	Product Owner micromanaging	10
			Disruption of the team by asking for statuses	14
			Documentation required by upper management	7
			Overcommitment caused by pressure from the product management	6
			Insufficient understanding about the development team autonomy	6
			Urgent fix or request from Product Owner	11
			Ad-hoc requirements from the management	14
	Working with requirements	Missing and unsystematically changed requirements	Requirements uncertainties	11
			Lacking requirements	9
			Delayed and changing requirements	7
			Newly arriving requirements	3
			New unnecessary requirements	13
			Too late changes in requirements	5, 14
		Ineffective Product Backlog prioritisation	Not regular refinement of Backlog	14
			Missing a clear prioritization of product backlog	10
			Problems with prioritizing feedback from various clients	10
			Not the right tools to measure requirement value	8
			Difficulty including long term quality tasks	10
			Product Backlog missing bug fixing request	14

Figure 3: Full structure of challenges of Product Owner (article numbers refer to to the article list in figure 6)

through transformation from plan driven to agile way of working, it can get confusing who takes over what tasks [8]. And situations where it is unclear who has the responsibility over what can arise [9] It is important that the Product Owner maintains the unrestricted authority in areas where he should have, e.g. in contact with the customer or stakeholder [22] and is still the decision maker towards his team [20]. Further challenge in this group connects to the Product Owner being indirect leader of the team, steering their motivation and providing vision [11], while keeping the Development Team self-organised as given in the Scrum Guide [18].

Disrupting autonomy of the Scrum team

The disruption of the Scrum team autonomy can come from more directions. An article by Tanner and Mackinnon [23] introduces a survey which shows that 44% of Scrum Team members experienced interruptions from management during Sprints for instance by adding ad-hoc requirements to the Sprint Backlog. The Product Owner should also protect the Team from the need to produce status reports [23] which might be required by the upper management used to heavy documentation [9]. The pressure from the client to receive results as soon as possible might be significant too, but the Product Owner should restrain himself from passing requests to

the Team during an ongoing sprint [16]. As well as he should listen to the Team, when it comes to system improvement and reducing the technical debt, which maximises long-term value of the system [8]. Lastly, the Product Owner needs to beware himself of strong opinions on how the Development Team should develop their tasks [16].

5.3 Working with requirements

Area *Working with requirements* contains the categories connected to the Product Owner's work with requirements, which is crucial for maximising the value of the Development Team's outcome.

Missing and unsystematically changed requirements
This category relates to the fact, that clients often change or make their mind about requirements in the middle or the end of a Sprint [6, 9, 23] which can lead to *missing or unsystematically changed requirements*. The Product Owner is the one ultimately responsible for the Product Backlog with its requirements [18] and he should be managing the requirement changes in a way that, does not hurt the team's work [6]. Missing requirements, which cause the work to standstill, also present a risk [9]. Requirements done half way and missing crucial information cause the Sprint Planning and estimation to be ineffective [9, 16]. Product Owner should further be able to reject requirements from stakeholders which are unnecessary [22].

Ineffective Product Backlog prioritisation
The last category refers to the challenges connected to Product Backlog prioritisation. One aspect of it is the general tendency to underestimate the time needed for the development of stories [23] and the consequent pressure to generate immediate value on the expenses of quality improving tasks, like refactoring [16].

The Product Owner should also aim for refining his own understanding of the user stories by engaging regularly with the client [23]. That will increase his ability to clearly prioritise the backlog, so the team can follow it [16]. The Product Owner needs to as well aim for measuring correctly the value of individual requirements in order to maximise the value of the development [11].

6 DISCUSSION

In this section we will discuss some of the findings and relate the two literature reviews and their findings.

Relation between the findings from the literature reviews
In figure 4 you can see how our findings from both reviews relate. In the first two columns, we present the aspects and elements of the Product Owner role resulting from the literature review on the textbooks. In the last column to the right, are the concrete challenges identified in the second literature review on research articles. The challenges are matched towards the corresponding elements in which they can occur.

We can see some of the elements have various challenges connected to them while others less. The most challenged element is the task and responsibility of *Product Backlog grooming*. Considering the Product Backlog and connected activities has been identified as the main focus of the Product Owner, it is not a surprise, that there this was where the most challenges were found. The second most challenged element is the *stakeholder management*, which is pointing to its importance as well, though the Scrum Guide does not mention it.

From the *Skills and Characteristics* aspect, the element with most challenges is *Communicator & Negotiator*. That confirms the need for good communication, which is the base of Scrum [18]. Further we can see in the figure 4 that two of the three common mistakes were confirmed by the challenges.

There are also elements which do not have any related challenges identified. That might be given by the nature of most of the reviewed articles, which did not focus on the Product Owner perspective and therefore did not cover the full picture. It could also serve as an indication of elements which do not have importance in the Product Owner role despite the books.

In the figure 5 the challenges which did not relate to any element are listed. All of them were from the area of *Organisational management*. These challenges are related to the Product Owner role, but not represented in any of the elements, which points to areas that the books did not take into account.

Insufficient technical knowledge of the Product Owner
One topic often mentioned in the research articles was the Product Owners not having enough technical knowledge which could lead to issues. Many of the Product Owners are having more business knowledge than technical knowledge or having a bad balance between the two [4]. The books are not entirely clear on this regard, but the Product Owner role is acting as a binding between the often-technical Development Team and the customers/stakeholders. So, having both technical knowledge and business knowledge would be an advantage, as it would be better for their "bridging"-role. Missing technical knowledge can be damaging to the development of the product and the team's work in general [8, 22].

Connection to related work
In the related work section, we presented four articles. Our literature review is quite unique, in that sense that it is about the Product Owner role as a whole, and not aspects of Scrum or agile-methods. Our results though relate to some of the challenges that article [10] found, mainly to their *Communication gaps*, as it describes how communication can fail due to gaps in roles over time and unclear vision or goals, which fits to our challenges in the area *Communication with different actors* and in the category *Organisational management*. The *Requirements validation* and *Requirements documentation* challenges are also showing similarities to our challenges in *Working with requirements* area.

Aspects	Elements	Related challenges
Responsibilities and tasks	Prioritising of the Product Backlog	Missing a clear prioritization of product backlog
		Not the right tools to measure requirement value
		Stakeholders have different expectations
		Problems with prioritizing feedback from various clients
	Product Backlog grooming	Product Owner not documenting request properly for the team
		Requirements uncertainties
		Lacking requirements
		Delayed and changing requirements
		Too late changes in requirements
		Not regular refinement of Backlog
		Difficulty including long term quality tasks
		Product Backlog missing bug fixing request
		Not shared understanding of requirements
	Stakeholder management	Unclear acceptance criteria
		Not shared understanding of requirements
		Stakeholders have different expectations
		New unnecessary requirements
		Difficulty including long term quality tasks
		Second-guessing the client
	Being available for the team	Lack of communication between Product Owner and Scrum Master
		Coordinating meetings
		Delays and unresolved issues
		Physical distance of desks
		Product Owner unavailable
	Making a vision for the product	
	Being a leader for the team	Challenge of being an indirect leader
Skills & Characteristics	Visionary & Doer	
	Leader & Team Player	Bad communication within the team
		Challenge of being an indirect leader
	Communicator & Negotiator	Unclear communication between Product Owner and team
		Bad communication within the team
		Product Owner not documenting request properly for the team
Maintaining good relationships	Scrum Master relationship	Lack of communication between Product Owner and Scrum Master
	Stakeholder relationship	New unnecessary requirements
		Difficulty including long term quality tasks
	Development Team relationship	Difficult relationship with team
Background for the Product Owner	Business background	
Common mistakes	Not enough trust between management and Product Owner	Unrestricted authority to make decisions in projects
		Product Owner not having a strong relationship with upper management
	Distance from the Team	Coordinating meetings
		Product Owner unavailable
	Product Owner Committee	

Figure 4: Challenges related to different elements of a Product Owner role

Area	Category	Challenge
Organisational management	Task dependencies	Requirement dependencies
		Inter-dependencies on other systems
	Unclear definition of Product Owner role	Unclear responsibility for management activities
		Product Owner not the actual decision maker
		Not clearly defined responsibilities
	Disrupting autonomy of the Scrum team	Product Owner micromanaging
		Disruption of the team by asking for statuses
		Documentation required by upper management
		Overcommitment caused by pressure from the product management
		Insufficient understanding about the development team autonomy
		Urgent fix or request from Product Owner
		Ad-hoc requirements from the management

Figure 5: Challenges which do not relate to any of the identified elements of the Product Owner role

7 CONCLUSION

Through two literature reviews, one on the peer-reviewed articles with Product Owner challenges, the other on relevant books on the Product Owner role, we have come up with several findings.

In the review of academic articles, we have identified 40 concrete challenges connected to the Product Owner role, split into eight categories of challenges, in three different areas: *Communication with different actors*, *Organisational management* and *Working with requirements*. However, there is a low amount of evidence found in the articles supporting each challenge as most of the articles did not have Product Ownership as a main topic. The was sparse information about the Product Owner per article shows the low focus on Product Owners and their perspective. Out of the 30 reviewed research articles, only 14 had relevant data to our research question.

We reviewed three books focusing on the Product Owner role in Scrum and identified five main aspects within the Product Owner role. It became clear that the books were often in accordance on the role and that they contained far more information than the Scrum Guide, as visible in figure 2, which focuses mostly on the work with Backlog. They thus expanded the description of the role immensely by adding to the responsibilities of the Product Owner in addition to specifying his desired characteristics, relationship to maintain and most common mistakes related to him.

The combined result of the two reviews provides a wide overview of the aspects and challenges of the Product Owner role, but it also sheds light on areas which might not be covered by one or other of the sources. Overall, we can conclude that there need to be more research done on the Product Owner role and the information on the role needs to be expanded, as it is a role binding the different actors in the development process.

7.1 Limitations

We acknowledge the limitations connected to our narrative review. We set up an inclusion criteria, that the considered books needs to have Product Ownership as a main topic, which excluded the books written on Scrum in general.

Further in connection to the systematic literature review, even though it was conducted to our best knowledge, we must acknowledge potential limitations. We limited our search for relevant papers to only include peer-reviewed articles, which may have excluded some sources of information, however we argue that it was necessary to keep a quality factor in otherwise broad search.

Moreover we chose to do two types of literature reviews that limits the consistency in our work and comparability between the results of the reviews. We were however clear on our method, which increases the validity.

REFERENCES

[1] Julian M. Bass. 2015. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering* 20, 6 (01 Dec 2015), 1525–1557.

[2] Elizabeth Bjarnason and Helen Sharp. 2017. The role of distances in requirements communication: a case study. *Requirements Engineering* 22, 1 (01 Mar 2017), 1–26.

[3] Maya Daneva, Egbert Van Der Veen, Chintan Amrit, Smita Ghaisas, Klaas Sikkel, Ramesh Kumar, Nirav Ajmeri, Uday Ramteerthkar, and Roel Wieringa. 2012. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. 86 (2012), 1333–1353.

[4] Meghann L. Drury-Grogan, Kieran Conboy, and Tom Acton. 2017. Examining decision characteristics challenges for agile software development. *Journal of Systems and Software* 131 (2017), 248 – 265.

[5] Robert Galen. 2013. *SCRUM Product Ownership - Balancing Value from the Inside Out.* RGCG.

[6] Lucas Gren, Richard Torkar, and Robert Feldt. 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. 124 (2017), 104–119.

[7] Petra Heck and Andy Zaidman. 2016. A systematic literature review on quality criteria for agile requirements specifications. *Software Qual J* (2016).

[8] Ville T. Heikkilä, Maria Paasivaara, Casper Lasssenius, Daniela Damian, and Christian Engblom. 2017. Managing the requirements flow from strategy to

release in large-scale agile development: a case study at Ericsson. *Empirical Software Engineering* 22, 6 (Dec 2017), 2892–2936.
[9] Rashina Hoda and Latha K. Murugesan. 2016. Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software* 117 (2016), 245 – 257.
[10] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband. 2014. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* (2014).
[11] Sigurhanna Kristinsdottir, Marta Larusdottir, and Åsa Cajander. 2016. Responsibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study. In *Human-Centered and Error-Resilient Systems Development*, Cristian Bogdan, Jan Gulliksen, Stefan Sauer, Peter Forbrig, Marco Winckler, Chris Johnson, Philippe Palanque, Regina Bernhaupt, and Filip Kis (Eds.). Springer International Publishing, Cham, 3–16.
[12] Craig Kuziemsky and Francis Lau. 2016. *Handbook of EHealth Evaluation : An Evidence-based Approach.*
[13] Craig Larman. 2007. *Agile and iterative development : a manager's guide* (10th printing ed.). Addison wesley, Boston, Mass.
[14] Janeth López-Martínez, Reyes Juárez-Ramírez, Carlos Huertas, Samantha Jiménez, and Cesar Guerra-García. 2016. Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review. *4th International Conference in Software Engineering Research and Innovation (CONISOFT)* (2016).
[15] Don McGreal and Ralph Jocham. 2018. *The Professional Product Owner.* Addison-Wesley.
[16] Nils Brede Moe, Aybüke Aurum, and Tore Dybå. 2012. Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology* 54, 8 (2012), 853 – 865.
[17] Roman Pichler. 2010. *Agile Product Management with Scrum - Creating Products that Customers Love.* Addison-Wesley.
[18] Ken Schwaber and Jeff Sutherland. 2017. The Scrum Guide. http://www.scrumguides.org/scrum-guide.html. (November 2017).
[19] Scrum Alliance. 2018. State of Scrum 2017-2018. PDF found on: http://info.scrumalliance.org/State-of-Scrum-2017-18.html. (2018).
[20] L. Siddique and B. A. Hussein. 2016. Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective. *Journal of Engineering, Project, and Production Management* 6, 2 (2016), 120–135. Copyright - Copyright Association of Engineering, Project, Production Management 2016; Last updated - 2016-08-01.
[21] Marilyn Sihuay, Abraham Dávila, and Marcelo Pessoa. 2018. Factors Models of Scrum Adoption in the Software Development Process: A Systematic Literature Review. (2018).
[22] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. 2014. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences* 119 (2014), 257 – 267. DOI: http://dx.doi.org/https://doi.org/10.1016/j.sbspro.2014.03.030 Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013.
[23] Maureen Tanner and Angela Mackinnon. 2015. Sources of Interruptions Experienced During a Scrum Sprint. 18 (2015), 3–18.
[24] Mathieu Templier and Guy Paré. 2015. A Framework for Guiding and Evaluating Literature Reviews. 37 (2015).
[25] J. Webster and R. T. Watson. 2002. Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly* (2002), 13–23.

A APPENDIX

List of articles with codes used in our literature review (see figure 6)

[1] Julian M. Bass. 2015. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering* 20, 6 (01 Dec 2015), 1525–1557.
[2] Elizabeth Bjarnason and Helen Sharp. 2017. The role of distances in requirements communication: a case study. *Requirements Engineering* 22, 1 (01 Mar 2017), 1–26.
[3] Maya Daneva, Egbert Van Der Veen, Chintan Amrit, Smita Ghaisas, Klaas Sikkel, Ramesh Kumar, Nirav Ajmeri, Uday Ramteerthkar, and Roel Wieringa. 2012. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. 86 (2012), 1333–1353.
[4] Meghann L. Drury-Grogan, Kieran Conboy, and Tom Acton. 2017. Examining decision characteristics challenges for agile software development. *Journal of Systems and Software* 131 (2017), 248 – 265.
[5] Lucas Gren, Richard Torkar, and Robert Feldt. 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. 124 (2017), 104–119.
[6] Ville T. Heikkilä, Maria Paasivaara, Casper Lasssenius, Daniela Damian, and Christian Engblom. 2017. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. *Empirical Software Engineering* 22, 6 (Dec 2017), 2892–2936.
[7] Rashina Hoda and Latha K. Murugesan. 2016. Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software* 117 (2016), 245 – 257.
[8] Sigurhanna Kristinsdottir, Marta Larusdottir, and Åsa Cajander. 2016. Responsibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study. In *Human-Centered and Error-Resilient Systems Development*, Cristian Bogdan, Jan Gulliksen, Stefan Sauer, Peter Forbrig, Marco Winckler, Chris Johnson, Philippe Palanque, Regina Bernhaupt, and Filip Kis (Eds.). Springer International Publishing, Cham, 3–16.
[9] Timo O. A. Lehtinen, Juha Itkonen, and Casper Lassenius. 2017. Recurring opinions or productive improvements–what agile teams actually discuss in retrospectives. 22 (2017), 2409–2452.
[10] Nils Brede Moe, Aybüke Aurum, and Tore Dybå. 2012. Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology* 54, 8 (2012), 853 – 865.
[11] Nils Brede Moe, Torgeir Dingsøyr, and Tore Dyb. 2009. Overcoming Barriers to Self-Management in Software Teams. 26 (2009), 20–26.
[12] L. Siddique and B. A. Hussein. 2016. Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective. *Journal of Engineering, Project, and Production Management* 6, 2 (2016), 120–135. Copyright - Copyright Association of Engineering, Project, Production Management 2016; Last updated - 2016-08-01.
[13] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. 2014. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences* 119 (2014), 257 – 267. https://doi.org/10.1016/j.sbspro.2014.03.030 Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013.
[14] Maureen Tanner and Angela Mackinnon. 2015. Sources of Interruptions Experienced During a Scrum Sprint. 18 (2015), 3–18.

Figure 6: List of articles with codes used in our literature review

Tasks & competencies of Product Owners in Scrum practise - A case study of Software Development with multiple stakeholders

Marie Jiříčková
Jonas Nielsen
mjiiko17@student.aau.dk
jnie14@student.aau.dk
Aalborg University
Aalborg, Denmark

ABSTRACT

Scrum is the most used agile framework nowadays and there is a growing amount of research and publications on its different parts and aspects. This article widens the research available on the role of the Product Owner, which carries immense responsibility, but has been given so far, the least attention of the Scrum Roles. Through nine interviews with Product Owner practitioners in Software Development, who deal with multiple stakeholders, and with the use of Grounded Theory as an analysis method, this article uncovers what the Product Owner role entails. The result are tasks and competencies of Product Owners in practise which are categorised into three main areas: Managing priorities, Cultivating relationships and Being visionary.

KEYWORDS

Product Owner, Scrum Role, Software Development, practise, case study, Grounded Theory.

1 INTRODUCTION

At the moment Scrum is the most used agile framework [18]. It originated from a Japanese article in 1986 and was then refined by Jeff Sutherland and Ken Schwaber into the framework known today [10]. Scrum builds on simple practices in managing work, products, customer participation, openness and in self-organising teams [10]. Scrum is meant to be simple to understand, but it is hard to master [10], as it is a framework, and not a method to follow.

The Scrum Guide describes that the Scrum framework supports developers in addressing complex and adaptive problems, while having a high level of productivity and creativity and at the same time delivering a product with maximum value for the client [14]. It is the opposite of a big collection of connected mandatory components [14], and therefore more in line with a framework. It leans on the scientific method of empiricism [14], as it is explanatory and experience based, with a focus on people and self-organisation. Scrum is built upon a series of values and commitments [13], were communication is a key value binding everything together.

Scrum is based on *Sprints* of work, lasting between two to four weeks, and have small teams of 3-9 persons [13]. It embodies different artefacts, including a *Product Backlog* and a *Sprint Backlog*, which serves to estimate and set goals for the overall product as well as the current Sprint. It utilises three different roles: a Development Team, Scrum Master and Product Owner. For an overall picture of the Scrum Framework see figure 1.

As described in the Scrum Guide [13], the Product Owner is responsible for maximising the value of the product through the work of the Development Team [13, 16]. The Product Owner is the sole person responsible for managing the Product Backlog [5, 13]. The Product Backlog is an artefact used by the Product Owner, to list and prioritise the requirements for the Development Team.

Scrum is usually not applied and used precisely as written in the Scrum Guide, but assimilated with other agile methods, tools or entire frameworks on top. The state of agile [2018] specifies that 70% of organisations using agile methods are using a combination of Scrum with some other agile method [18]. Many of the organisations are using a mixture, as Scrum does not provide sufficient tools when scaling up to larger organisations with multiple teams.

This article will be focusing on the Product Owner role in Scrum, as they are the least described role, both in the Scrum Guide, but also in literature [7], even though they are a key person in the success of a project [11]. To help focus our research, we have worked with the following research question:

What does the Product Owner role entail in a software development practise from the Product Owner's perspective?

The article is built as a case study on the Product Owner role with nine interviews with Product Owners from various companies. It is organised as follows: First, we summarise the related work, then we introduce the method - our data collection and data analysis. Next, we present our results and discuss some related points, and finally we conclude on our findings.

2 RELATED WORK

In this section, we present an overview of the research done on the Product Owner role. To find the literature, we used Aalborg University's set of databases and we searched for the keyword "Product Owner" (without quotation marks). We limited our results to the field of Computer Science and we only included full articles. This resulted into identifying five articles presented below. Looking through related work gave us the insight into the state of research

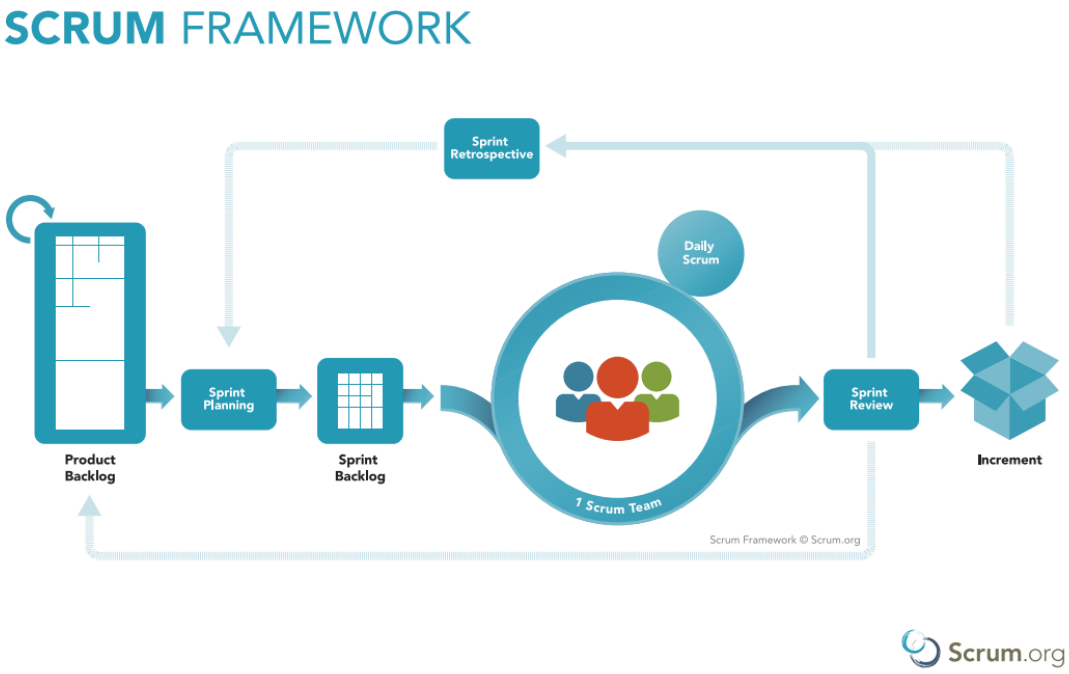

Figure 1: The Scrum Framework [14]

in this area.

Article [16] is investigating the role and responsibility of Product Owners in Software Development and is trying to answer their research question: "Is there a conformance between the understanding of product owners of their own role and responsibility, and the ideas put forward by the initiators of Scrum?". The knowledge and skills a Product Owner should have, according to the Scrum framework, is compared to the perception of the role by actual Product Owners. The exploratory paper is conducting semi-structured interviews with five active Icelandic Product Owners working in five different organisations to find out how they define the role and responsibilities. The result shows that Scrum application varies highly between the organisations themselves as well as inside of them. The application is also often very different from the Scrum methodology, for instance when there is more than one Product Owner for a project. It is also clear that the majority of Product Owners uses different project management methods within their operations.

Article [1] focuses on how Product Owners and Product Owner teams scale up in larger offshore software enterprise development programmes [1]. The research question presented is: "How do practitioners describe enhancement and expansion of functions within the product owner role, to meet the needs of large-scale offshore enterprise software development programmes?". The article is a qualitative study of software engineer practice in eight international companies within which 46 practitioners was interviewed. The main contribution represents the identification the nine Product Owner team functions - groom, prioritiser, release, master, technical architect, governor, communicator, traveller, intermediary and risk assessor - which are used to scale agile methods to large projects. The functions are mapped into scrum-of-scrum process revealing two function classes: client-side and production-side.

On scaling Product Ownership, the same author recently co-wrote one more article [2]. 93 practitioners who are working in cross-border teams were interviewed to map their activities. On top of their conventional responsibilities like prioritising requirements, Product Owners in large-scale projects must also perform three additional group of activities. First, manage scale and handle large number of stakeholders and long-term software release timescale. Second, manage distance, which is about the global software development, and lastly manage governance, which points to having the right consistency between the agile teams. In connection to that, the authors also describe the valued Product Owner behaviour.

Another article [12] is trying to answer a research question related to the scaling of the Product Owner role: "How has the Product Owner role been scaled in large-scale distributed Scrum projects?". The results are based on 58 semi-structured interviews and include having a local Product Owner available at each site, creating a Product Owner team, having frequent communication between the Scrum teams and Product Owners, and communicating clear priorities to all the stakeholders.

Last article we will mention, is an exploratory case study of responsibilities and challenges of Product Owners [8]. The case study was done in the company Spotify, where five employees were interviewed, three of which were Product Owners. The results divide the responsibilities between daily work connected to the Development Team functions and the long-term activities pointing to vision of the product. The main challenge was identified as the inspiration and encouragement of the team members to communicate openly within the team and with the stakeholders.

In summary, only two of the articles deal directly with Product Owner role and their responsibilities [8, 16]. The rest focuses on the scalability of Scrum in connection with the role [1, 2, 12]. With this article we want to add to the limited research done on the Product Owner's practise, which have been proven wider and different [8, 16] from the official Scrum Guide description [13].

3 METHOD

The empirical evidence we present in this paper was obtained by conducting a qualitative case study. In the following we describe in detail the participants in our study, our data collection as well the process of our data analysis.

3.1 Data collection

For data collection, we have chosen to do semi-structured interviews, to help us explore the practitioners and their experiences. The semi-structured interview process can help extracting the latent knowledge from the interviewees, by going down interesting leads by allowing the interviewer to ask follow-up questions and grant the opportunity to elaborate on certain points [3].

Participants
The practitioners participating in our study were eight Product Owners and one a Product Manager. The participants were from different industry sectors, as seen in figure 2, and seven of them are in projects with multiple clients.

Setting & Procedure
Prior to interviews, we made an interview guide with questions. It was split into four sections: Background information on the participant, Background information on the company, Role/job specification and Company relationships. Under each section we had multiple questions for instance in the Role/job specification we asked: "What would you say is your single biggest responsibility?". The questions gave us a tentative structure to follow under the interviews and were refined as more interviews were performed.

With each participant we conducted one interview. The interviews were conducted through Skype, through telephone or in person. The interviews were done in English with one exception which was done in Danish due to the language barrier of the participant. We were both present at the interviews except from the one in Danish, which was done only by one. Before each interview was conducted, the participants were asked about their permission to be audio recorded and transcribed, which they all agreed to. The average length, both of interview time and transcription pages of the interviews, can be seen in figure 3.

3.2 Data Analysis

Transcription
To help us analyse our interviews, we transcribed them according to Steinar Kvale, who emphasises the need to transcribe all empirical data, so it can be analysed [9]. We chose to do full transcriptions, in order not to lose any information, furthermore it was necessary for our data analysis. Kvale states that there is not a universal method to transcribe interviews - the most important thing is to be consistent in the style of transcription chosen and to get as close to the oral language, through written language, without losing on the validity and reliability [9]. As we did not need the specifics of our informants, we chose to make them anonymous in the interview transcriptions and throughout the paper. There were in total 128 pages of transcription as seen in figure 3.

Grounded theory
For analysing of our data, we chose to work with Grounded theory, as it is very fitting for our research. It complements research in social interactions and human behaviour, and in areas which are under explored [5, 6, 15], which we consider our topic to be as illustrated in the Related work section 2. Grounded theory does not work with a hypothesis, which needs to be validated, instead the main concerns are uncovered in the process [5], which makes it one of the few methods which focus on theory generation [6]. Grounded theory is also increasingly used in papers studying agile teams [6] some of which we used for drawing inspiration [1, 5, 6, 15].

Initial coding
The analysis started with *Initial* coding, where we coded segment by segment. Initial codes are provisional and are meant to explore whatever theoretical potential of our data [4]. When coding we followed some basic rules as suggested by Charmaz [2006]:

- Remaining open
- Keeping codes simple, short and analytic
- Comparing codes between each other
- Working quickly through the data to evoke fresh thinking

To create initial codes we used 'in vivo' categories where area under exploration is named with the use of words directly from the source [19]. The advantage of 'in vivo' codes lies in the fact, that they are free from prior assumptions of the researchers [1]. For the coding we used QSR's software, NVivo 12, which is a qualitative data analysis software tool.
The initial coding was done separately by both the authors of this article (with the exception of one interview done in Danish). The resulting codes were not compared directly but served as a basis for concept creation via discussion which is described and used in the next phase. In figure 3 the number of initial codes in total, in all the interviews, and per interview is presented - these numbers are counted as an average from both authors.

Focused coding
After *initial* coding we continued with *focused* coding, were we

Participant	Industry	Role	Experience	Agile method	Certification	Scrum Use
P1	Financial services	Product Owner	3 years	Scrum	Product Owner	Scrum
P2	Financial services	Product Owner	2 years	Scrum, Safe	Scrum Master	Scrum
P3	Government services	Product Owner	5 years	Scrum, Kanban	Product Owner	Mixed agile
P4	Financial services	Product Owner	1 year	Scrum, Kata, Lean	Scrum Master	Scrum
P5	Government services	Product Owner	1 year	Scrum	-	Scrum
P6	Energy sector	Product Owner/ Team Lead	8 months	Scrum	-	Customized Scrum
P7	Financial services	Product Owner	3 years	Scrum, Kata	Product Owner	Scrum
P8	Financial services	Product Manager	1,5 years	Scrum, SAFE	Scrum Master/Product Owner/ Product Manager	Customized Scrum
P9	Consumer Products services	Product Owner/ Product Manager	2 years	Scrum, SAFE	Scrum Master	Customized Scrum

Figure 2: Interviewees' demographics

9 interviews	Length in time	Length in pages	Number of intial codes
All interviews in total	5 hours and 12 minutes	128	374
Average per one interview	35 minutes	14	41

Figure 3: Length of interviews an number of initial codes

identified the most frequent and important initial codes, to create concepts, and organised and synthesised larger parts of the data into those concepts [4] by conducting a second round of coding, meaning, that both authors coded all 128 pages of transcriptions once more. We merged our results from the second round of coding to compare between us and find a consensus on understanding the data, as well as increase validity. In this phase a *core* category was identified - Tasks & competencies of Product Owners with multiple stakeholders. It is a category which points to the most significant and frequent concepts or codes, which is also connected to as many other codes[17]. This *core* category guided our further coding and created higher focus [17]. Furthermore, all codes and concepts were being compared and revisited in order to create a higher level of abstraction - categories and their groups [15]. With the use of constant comparison, which refers to continuous search for similarities as well as differences when creating categories, we refined our category structure [1].

4 FINDINGS

In this section we present the findings of our study, the skills & traits of Product Owners in Scrum practise. They are represented by three main categories: *Managing priorities*, *Cultivating relationships* and *Being visionary*. The categories were raised based on 12 concepts, as seen in figure 4, where the category structure is visible. It shows for each concept, the number of participants mentioning the given concept in their interview and the total number of references from all interviews together pointing to this concept.

4.1 Managing priorities

When asked about their single biggest responsibility, four of our participants answered it is prioritisation of the Product Backlog or the inputs they receive from different stakeholders. This category's concepts are about the different aspects of prioritisation, that Product Owners do: *Managing different Stakeholders with different priorities*, *Effective Product Backlog prioritisation*, and *Balancing technical debt and progress.*

Managing different Stakeholders with different priorities

As described in the data collection section, participants in our study, with one exception, deal with multiple stakeholders and therefore also have to manage their priorities which may differ. The difference in priorities does not come only from in between the single stakeholders but also from the different levels of stakeholder representation, meaning the leaders who deal with long term strategy and users who work with the system every day. That this can be

Grounded Theory	Category	Concepts	Total number of participants	Total number of references
Tasks & Competencies of Product Owners with multiple Stakeholders	Managing priorities	Managing different Stakeholders with different priorities	7	18
		Effective Product Backlog prioritisation	8	29
		Balancing technical debt and progress	3	4
	Cultivating relationships	Building a network	2	3
		Being available for the team	6	10
		Having good relationship and frequent communication with SM	5	12
		Cooperating with upper-management	6	9
		Maintaining good relationship with Stakeholders	7	34
	Being visionary	Being autonomous	5	6
		Having an overview	4	8
		Being adaptable and capable of fast decision making	3	4
		Being unofficial leader	2	2

Figure 4: Concepts and categories

difficult and surprising in the beginning, and one of the participants expresses it:

"*It's hard to go from theoretical to hands on. I know we have a lot of stakeholder handling, it's a highly political point, so you have to learn that outside of the theory. Without any clear information, that can drain some energy, that's not part of the certification or training*"

- Participant 8

Another prioritisation input can come from the different law regulations, which need to be implemented depending on the nature of the developed software. Although Product Owners are not necessary alone on these prioritisation decisions, some participants get help from their supervisors, others from their developers:

"*The important thing is to say here, that I'm not alone in this part (...) I of course talk to my developers, I of course talk to support, maybe I also ask them about what they actually think regarding it.*"

- Participant 8

It also needs to be noted, that it is common among the interviewees, that the Development Team are in contact with the stakeholders as well, so some of the small individual requests or fixes get solved on the developers level without the Product Owners need to intervene nor prioritise.

Effective Product Backlog Prioritisation

The Product Backlog is an essential artefact for the Product Owner, which is described in the Scrum Guide [13] and the Product Owner is the sole person responsible for it. The Product Owners are refining the items in it (most of our participants work with User Stories), making sure there is enough documentation, and also ensuring that there is a plan for the team on what to develop next. One participant also described the importance of breaking down requirements going into the Product Backlog:

"*I see a lot my role as breaking that down, because when IT goes wrong it's often because, perhaps you have this whole.. it's too huge what you're trying to estimate, both in cost and time. So we try to break it down.*"

- Participant 3

The Product Backlog contains many items, which change and develop all the time, however there is a limit to the number of items it is possible to handle in one Sprint. In order to progress efficiently and satisfy the stakeholders, the Product Backlog has to be effectively prioritised and items picked with care for each Sprint as described by one participant:

"*We have a lot of items and not a lot of time. So it's every single user story is solely picked out for the next sprint. And it changes all the time, that's one of the great things about being agile.*"

- Participant 4

There are also tensions connected to the number of items in the Product Backlog, unexpected bugs adding to this pressure, which emphasises the need for *Effective Product Backlog Prioritisation*:

"*Yeah, the workload is a general problem. And that's why the prioritising is kinda delicate matter sometimes. You have to be real sharp in ... Okay, this one is the most important, that gives the banks the most value for money.*"

- Participant 4

The ultimate goal is to prioritise all the inputs to get the right thing for the right time in the desired quality.

Balancing technical debt and progress

The concept *Balancing technical debt and progress* is about the struggle between delivering the most value from the business point of

view, but at the same time to limit the creation of technical debt. The struggle gets harder when there are fixed deadlines for delivering some tasks and the project has to proceed, as one participant describes:

"*Often we have a deadline that we cannot do anything about and so it's a tension because sometimes we need, it's necessary to do like quick and dirty solutions because there is these deadlines and then when will we get back to this and get it fixed in a proper way.*"

- Participant 1

It can also become a point of tension between the Product Owner who promised delivery to the customer, and the team which is feeling more urgently the pressure of the technical debt created. It is important to keep technical debt in mind and find a suitable compromise for solving it.

4.2 Cultivating relationships

The Product Owner role is positioned in between others, working like a connection between the Development Team, the stakeholders and management in the company he is working for. It is important to maintain good relationships, to gain trust and ensure good cooperation. The concepts in this category present the ways that Product Owners cultivate these relationships: *Building network*, *Being available for the team*, *Having good relationship and frequent communication with Scrum Master*, *Cooperating with upper-management* and *Maintaining good relationship with Stakeholders*.

Building network
A Product Owner often serves as a connector between different actors in the development process. This is a major part of a Product Owner's job, which one participant confirms, as he estimates spending 85% of his time on stakeholder management. This makes it important to build one's network and orient in the relationships around:

"*I think, that's for a Product Owner, I think that, that's a very important thing, is the network and also stakeholder management, but also understanding the political atmosphere or relations that exists.*"

- Participant 6

Participant 6 also talks about the need to be outgoing, which gives an advantage when building relationships and trust.

Being available for the team
In the relationship towards the team, our participants stressed the need to *be available for the team* in some form, on a daily basis, though not necessarily in person. One describes, that they comment around issues in tasks, in the software development tool Jira. The same Product Owner also talks about the importance of dialog and being open to the inputs from the developers.

Two other Product Owners also talked about being ready to help their team in tasks which are not normally their responsibility if needed, e.g. with some testing when the responsible person is missing or with writing release information.

Having good relationship and frequent communication with Scrum Master
Scrum Master role is important part of the Development team and even though their work was a bit different in each case, it was clear, that the Product Owner and Scrum Master need to work in agreement. Two of our participants talked about not having a good relationship due to different points of view, on how things should be done and the subsequent tensions:

"*But what I experienced was that in a planning session with the Product Owner actually left the room in anger because she did not have the opportunity to plan for them anymore and that was quite hard*"

- Participant 5

In the quote above, the participant had, in his previous experience as a Scrum Master, a disagreement with the Product Owner, who was overruling the team and had an expectation on how much they should deliver, which is something the team should decide through their own Sprint Backlog.

A positive cooperation between Scrum Master and Product Owner was connected to descriptions of frequent communication between the two, either by sharing the same office space or setting up several meetings a week to coordinate and discuss.

Cooperating with upper-management
The Scrum Guide [13] does not take into specific consideration the relationships the Development team has towards the rest of organisation it is part of. But Product Owners in practise need to actively cooperate with other parts of the organisation, if not with other teams, then with their upper management. One part of the cooperation is about updating, reporting and consulting on further approach:

"*So I usually each week have a status meeting with my business line manager which is my leader (...) and we talk about what is going on and do we need more, what is the status and etc.*"

- Participant 5

Another important part is cooperating on priorities, one interviewee described that he is not always alone on difficult prioritisation decisions:

"*But let's say that I have five User Stories and they are very important, equal important to the business, to the stakeholders, then the Chief of department can help me to say: okay, this one is number, two, three and four. So we agree on what's the most important.*"

- Participant 4

Furthermore Product Owners need to stay in line with higher strategy decisions made about products as well as deal with occasional pressure put on their prioritisation:

"*Sometimes it could be like when some of the clients would like us to do something individual for them, then of course management is their customers so they'd like to please them. And then they come, you must put this on top of your backlog and I say, no I cannot do that, because I have this and this and this and this and there are deadlines and.. you know. So there is a tension.*"

- Participant 1

If tensions are to be avoided there must be cooperation and respect towards the decisions made on both sides. The Product Owners have a better detailed overview on the work and are qualified to make more informed decisions in some case, but there also need to be space for including decisions made on the higher levels of the organisations.

Maintaining good relationship with Stakeholders
Though stakeholders are mentioned in Scrum Guide [13] in context of the ceremonies they attend, the work with them is not further described. The concept *Maintaining good relationship with Stakeholders* is about managing their needs, while building trust. Managing their needs should be distinguished from prioritising their needs, as it is more about finding out what they want and making sure that the final solution is useful. One Product Owner put it like this:

"Because you can spend lot of money and time on things that are not like expected or there is a say-do problematic where customers say one thing but actually want something else. How do you understand that actually, what they actually want for what they say."

- Participant 3

In order to attain their needs frequent contact, feedbacking and dedication is required. One Product Owner pointed out though that it can be tricky to balance the time spent ensuring, that customers have a good experience with producing something visible for which the customers pay.

Communication is the key to building trust between the Product Owner and his stakeholders. Most Product Owners have set up some form of regular meetings, for instance monthly meetings, or the stakeholders are attending the Sprint Reviews. It is important to keep active dialog also between these meetings. Decisions need to be clearly and honestly communicated and explained in order to build trust and meet expectations, even when things are not going as planned. This is summarised by one participant this way:

"*Being clear and honest about what you're doing and also telling why, advising them, why we do the thing we think - this is the way to go.*"

- Participant 3

The same participant also stresses the importance of seeing the project as shared, where everyone is committed and invested.

4.3 Being visionary

The category *Being Visionary* consists of four concepts: *Being autonomous*, *Having overview*, *Being unofficial leader* and *Being adaptable and capable of fast decision making*. In general, this category refers to the ability of the Product Owner to be able to be on his own, keeping the bigger picture in mind while following through with everyday work. It is also about the trust that the Product Owners have from their supervisors that they will be able to progress and make decisions on their own, without supervision.

Being autonomous
In context of this concept our interviewees talked about the freedom they have in their day to day work, decisions and contact with stakeholders. It is partially caused by the nature of the work in an agile environment, where fast thinking and decisions are required when work needs to keep progressing. The need for autonomy can also be enhanced when the Product Owner is not from the same company, but a representative from a client, as one interviewee put it:

"*I'm [an] employee in a bank but actually working at another company, so you're very much on your own, very very much on your own, and we have more Product Owners which have stopped. They didn't like it. They were too far away from their leader and from their colleagues (...) So with this set up, it's important that you can do on your own, and that you can make your own decisions, that's very important.*"

- Participant 1

The autonomy is also given by the mandate the Product Owners have from their supervisors. They are the ones with the full picture of the product and all things connected to it and therefore they are trusted to be in the best place to make the decisions about it.

Having overview
In order to be able to make effective decisions it is important for the Product Owner to know what is going on. One aspect is seeing into the Development Team's work - to see what task is in what stage, if anything is not going according to plan and what can be done to help. One participant explained the need to be close to the team to keep up:

"*So if you are very detached to the team then you always feel that you are left out (...) decisions are made on the fly also in the team, right, and technical solutions can be changed or something, or merged or so on. So you need to know what is going on, all the time in order to not feel left out.*"

- Participant 9

Another aspect of *Having overview* is on the higher level - to see what are the priorities, and work in a longer term, not just the next Sprint ahead.

Being adaptable and capable of fast decision making
An important part of doing agile development is that things can change, and development work can take a sudden different turn. It is not just the client that changes their mind, but there can also be

unexpected issues in parts of the system already running live which need to be fixed fast, or delays caused by tasks being dependent on the work of other teams. One interviewee summarises the need to be adaptable like this:

"*So things change just along the way and all the time and we need to adjust where you are and where we are and where the company [is] and where our customers are ... So you should be very adaptable.*"

- Participant 9

The Product Owners need to fit into the agile environment and be adaptable to sudden changes, so a vision and not a concrete plan is preferable

Being unofficial leader
The Product Owner is not officially the supervisor to the Development Team, that role is usually the Department Chief or Head of Development, but in the daily work they are the ones making the final decisions, as they are responsible for the product. That *Being unofficial leader* can be challenging, is something that the Product Owners are discussing in one of the companies:

"*What (...) more Product Owners are talking about is that you are not officially the leader of the Scrum Team but in a way you are their leader, so that can be little hard to manage, because they have a leader - department chief or something like that - but in a way we act on daily basis, you are kind of a leader too. (...) but I can't fire them. I can't hire anyone, I can't decide whether they will have a bonus or a higher pay check or something like that. But I lead them on daily basis in some way.*"

- Participant 1

So it is about finding the balance of leading the team successfully further in the development process to bring the best value to the customer without falling for being too directive and restricting the self-organisation of the Development Team.

5 DISCUSSION

In this section we discuss some further observations from our participants we find interesting though not part of our theory described in the Findings section. We also summarise implications of our work for practise and how our research correlate with the related work described in the section 2.

Scrum bubble
Participant 3 talked about the extremes of doing the waterfall method PRINCE2 and the agile framework Scrum, both of which he had the chance to try without any customisation. He described PRINCE2 as a heavily document based approach which is focusing on the overall organisation of a project while it is missing any guidance in the everyday development work. On the other hand, Scrum focuses on the day to day run of a project and encloses the team's work into Sprints, which he describes as follows:

"*The developers in the Scrum team have that.. that they can actually say: Eh, I can't answer anything else that what's in the sprint. So the next two weeks you can't ask me anything. So they are like in a bubble. Whereas in my role, yea, I'm in the team but also I cannot just say for two weeks now I'm in a bubble.. I get, got the interruptions all the time.*"

- Participant 3

Later, he also adds:

"*I think we're perhaps reaching a better way, where it's more not completely Scrum, not completely Prince2, but more the organisation is more into okay, perhaps we can do it more agile and we are also more flexible in also acknowledging that they also have some plans which should follows... so it's not just like: we don't know let's see how we, how far we get in two weeks. That's not enough for huge organisation, they have to have some kind of plans.*"

- Participant 3

The above quote summarises the need to find a compromise between the two approaches, which would make the development effective for everyone, because working solely with one approach can lead to undesirable extremes.

No team related issues
An interesting topic, coming out of the interviews, was the absence of team related conflicts or issues. In our interview guide we had a section on company relationships in which we directed questions specifically on issues in the relationship between the Product Owner and the Development team. One Participant explained that he does not find the work with developers challenging, like with the clients, because everything is more straight forward:

"*Not the developers actually, because it's more straight forward it's more like: what do you mean by that, we don't understand that, and then I go: let me find out. So I don't find that difficult, because then we can just talk about that.*"

- Participant 3

We noticed though that, the participants did not want to mention incidents putting them into bad light, so if we would have interviewed other roles, we would have gotten a different perspective.

Clients in direct contact with Development Team
In some of the cases, the participants explained that the Development Team had direct contact with the clients. In one case, the Development Team took assignments without involvement of the Product Owner:

"*So when [they] need to have something developed, which is for them only, they pay for it themselves, and then first they can contact the team and if it can be made in less than half a day then the team will just do it and if it will take more than half a day they will come to me and I will put it on the backlog and see when we can find the time to make it.*"

- Participant 1

The Product Owner only places it on the Product Backlog, if it is something that the Development Team estimates takes longer. This also shows the amount of trust between the Product Owner and

the Team, as wrong estimates can shift some tasks around in the Product Backlog. One of the other Product Owners encouraged the Development Team to be more in contact with the clients:

"*So what I strive for is actually, that there is as much frequent interaction between the developers and the end-customers as possible, and I'm not stepping in between that. And that's both because it's ... eases effort and the pace on the task that we need to do, but looking as also being Team Lead, that it can also increase business knowledge within the developers minds and it just gives a better collaboration and link in between.*"

- Participant 6

This is one of the ways for the Product Owners to give more value to the product by enabling the Development Team to make better decisions later, when they have more business knowledge and can give more feedback on the Product Backlog.

Different perspectives
Some of the Product Owners also have experience in being Scrum Masters before becoming a Product Owner. This gave them more insight into how a Scrum Master tries to optimise the Team and their everyday work, compared to the Product Owners responsibilities. It can take time for the Scrum Master's work in this are to take affect:

"*(...) the Scrum Master really wants the team to do its best and deliver the most.. but usually it.. what the Scrum Master does here takes a bit of time before the Product Owner really can see the benefit of it.*"

- Participant 5

This experience can help the Product Owner see it from both perspectives, as their roles are opposites. The Scrum Master is the master of the process, a facilitator of the meetings, the usage of artefacts and enabler in the teams, whereas the Product Owner is, as mentioned before, a visionary for the product. The Product Owner further says:

"*I think, in the future, if I come to a new company, it is having the knowledge of really knowing what a Scrum Master [is], how that person sees him, how the Product Owner sees him, because I've tried both, so therefore I have some knowledge and insights into how things will potentially create tension.*"

- Participant 5

The knowledge can help in preventing some tension between the Product Owner and the Scrum Master, and also help in having a good and honest communication between Product Owner and Scrum Master.

Connection to related work
In relation to the research papers described in section 2, we found some similarities in our findings with the results presented by the papers. In connection with the article [16], we can confirm the same findings on all three main points about Scrum application in practise, as the Product Owners in our case are using the Scrum framework mixed with other agile methods, they are admitting the application of Scrum differs team to team, and it is clear the Scrum methodology is often not 100% followed. The article [8] presents that the Product Owners main challenge "*(...) is to inspire and encourage team members to collaborate and communicate within the team and with stakeholders.*", which connects to the discussion point of the Development Team communicating directly with the clients. It also supports our findings on the Product Owner being an unofficial leader and being available for the team. We can say our work is in agreement with the other research done, while still contributing with new findings on the tasks and competencies of Product Owners.

Consequences for practise
Our research, together with some of the other mentioned papers [8, 16], uncovers some consequences for the Scrum practise. The Scrum Guide's definition of the Product Owner is mainly about their management of the Product Backlog [13]. What the Scrum Guide does not cover, are the competencies and some of the surrounding tasks of the Product Owners. The Product Owners have some of the largest responsibilities, as they need to be visionaries for the product and also be unofficial leaders for the team. The Scrum framework is meant to be easy to learn and hard to master, but some refinement is needed to balance the scales between these two characteristics and a new refinement of the Scrum Guide, could help Product Owners, and Scrum Teams, with understanding the role and its tasks more clearly. An immense aspect of being a Product Owner, is the amount of stakeholder management and communication with clients, as we have covered, which is essential for making the best product and that is something which deserves more attention in the Scrum Guide and other main sources on the framework.

6 CONCLUSION

This paper presents an empirical study investigating the role of Product Owner in practise. The results were found through nine interviews with practitioners, who had the Product Owner role. The interviews were analysed with the use of Grounded Theory.

The results show that the Product Owner role entails three types of tasks and competencies: *Managing priorities*, which is about the different aspects of prioritisation the Product Owner has to handle, *Cultivating relationships*, which points to the need for building trust and good relationships with the different actors surrounding the Product Owner, and lastly, *Being visionary*, which is focused on the Product Owner being able to follow a vision of the final product on his own.

With our findings, we enriched the Product Owner role description, which is limited in the Scrum Guide to describing work connected to Product Backlog [13]. We also contributed to the research done into Product Owner's role, as we specified the tasks Product Owners face and competencies, they should possess in their role.

6.1 Limitations

We acknowledge the limitations in doing a case study, as the findings and results could be different in other organisations and development processes. The concepts and categories found through the usage of our qualitative study, can be different in another team, as with all qualitative studies, however we have been clear on the method used and the results found, and can thereby make a stronger validity and replication claim. The number of participants can be argued for being in the low end, but the results are similar to other related studies [8, 16] and our study is still in a new research area, so differing results are to be presumed.

All our participants, except one, is all Product Owners, which might give a one-sided perspective, and a study that have interviews with other roles, such as Scrum Master or Development Team members, would have given new perspectives.

6.2 Future work

To strengthen our research even more, additional longer studies on a larger sample size of Product Owners, would be beneficial to give even more information on the role. An additional field study, where observations and interviews are conducted with a Product Owner, would provide valuable data on the role and on their day to day activities, as we can observe if the Product Owner conducts the tasks, the way he says and thereby have more objective data. This will give research that can help maximise the Product Owner role even more, but also give new Product Owners a better view on the role, its tasks and needed competencies.

Lastly a more focused study on the Product Owner and their usage of artefacts, can show more about the vast usage of their tools, which the Product Owners need to possess, to be effective at their role.

ACKNOWLEDGMENTS

We would like to thank all the participants in our study for their time and willingness to share their experiences.

REFERENCES

[1] Julian M. Bass. 2015. How product owner teams scale agile methods to large distributed enterprises. 20 (2015), 1525–1557.
[2] Julian M. Bass and Andy Haxby. 2019. Tailoring Product Ownership in Large-Scale Agile Projects: Managing Scale, Distance, and Governance. 36 (2019), 58–63.
[3] David Benyon. 2010. *Designing interactive systems: a comprehensive guide to HCI and interaction design* (3rd ed.). Pearson Education Limited.
[4] Kathy Charmaz. 2006. *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis.* Sage Publications.
[5] Rashina Hoda and Latha K. Murugesan. 2016. Multi-level agile project management challenges: A self-organizing team perspective. 117 (2016), 245–257.
[6] Rashina Hoda, James Noble, and Stuart Marshall. 2012. Developing a grounded theory to explain the practices of self-organizing Agile teams. 17 (2012), 609–639.
[7] Marie Jiříčková and Jonas Nielsen. 2019. *The Product Owner role in Scrum development - A review of guideline and research literature.* Department of Computer Science, Aalborg University.
[8] Sigurhanna Kristinsdottir, Marta Larusdottir, and Åsa Cajander. 2016. Responsibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study. In *Lecture Notes in Computer Science (Human-Centered and Error-Resilient Systems Development)*, Vol. LNCS-9856. Springer International Publishing, 3–16.
[9] Steinar Kvale. 2007. *Doing interviews.* Sage Publications Ltd.
[10] Craig Larman. 2007. *Agile and iterative development : a manager's guide* (10th printing ed.). Addison wesley, Boston, Mass.
[11] Dominik Maximini. 2018. *The Scrum Culture : Introducing Agile Methods in Organizations* (2nd ed.). Cham: Springer International Publishing.
[12] Maria Paasivaara, Ville T. Heikkilä, and Casper Lassenius. 2012. Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum. In *Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum.* 174–178.
[13] Ken Schwaber and Jeff Sutherland. 2017. The Scrum Guide. http://www.scrumguides.org/scrum-guide.html. (November 2017).
[14] Scrum.org. 2019. WHAT IS SCRUM? (May 2019). https://www.scrum.org/resources/what-is-scrum
[15] Lubna Siddique. 2016. Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers? Perspective. 6 (2016), 120–135.
[16] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. 2014. The Role of the Product Owner in Scrum-comparison between Theory and Practices. *Procedia - Social and Behavioral Sciences* 119 (2014), 257 – 267. https://doi.org/10.1016/j.sbspro.2014.03.030 Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013.
[17] Robert Thornberg and Kathy Charmaz. 2014. *Grounded theory and theoretical coding in Uwe Flick The SAGE handbook of qualitative data analysis (pp. 153-169).* SAGE Publications, 55 City Road, London.
[18] VersionOne. 2018. VersionOne 12th Annual State of Agile Report. PDF found on: http://tiny.cc/5p1q6y. (April 2018).
[19] Carla Willig. 2008. *Introducing Qualitative Research in Psychology.* Maidenhead: Open University Press.