Empirical Software Engineering (2021) 26: 3
https://doi.org/10.1007/s10664-020-09899-4

Spearheading agile: the role of the scrum master in agile projects

Yogeshwar Shastri[1,2] · **Rashina Hoda**[3] · **Robert Amor**[4]

Accepted: 6 November 2020 / Published online: 6 January 2021

Abstract
Scrum innovated the role of the scrum master in software engineering. The scrum master is envisioned in agile literature as the "*servant leader*" who serves the team in a multitude of different ways, which include promoting scrum, facilitating the team's functioning, and removing obstacles. However, empirical studies focusing on the role of the scrum master in practice are scarce. To address this gap, a Grounded Theory study with a mixed methods approach was carried out which included semi-structured interviews with 39 software practitioners and a questionnaire with 47 respondents. In this study, we present and describe the scrum master's role in agile projects in terms of (a) the grounded theory of the role of the scrum master which involves everyday activities of *facilitating, mentoring, negotiating, process adapting, coordinating,* and *protecting*; (b) the varying involvement of the scrum master in selected agile practices carried out by the team; and (c) a positive association between the presence of the scrum master and the frequency with which agile practices are carried out by the team. This study presents for the first time a multifaceted study of the multiple dimensions of the scrum master role and will enable practitioners to better manage expectations of this role in practice.

Keywords Grounded theory · Mixed methods · Scrum master · Agile software development · Agile project management · Scrum · Agile

Communicated by: Hakan Erdogmus

✉ Yogeshwar Shastri
ysha962@auckland.ac.nz

Rashina Hoda
Rashina.Hoda@monash.edu

Robert Amor
trebor@cs.auckland.ac.nz

Extended author information available on the last page of the article

1 Introduction

The scrum master is a key player in implementing and sustaining scrum, the most popular agile software development method (Sutherland and Schwaber 2017), (VersionOne Inc. 2018), (VersionOne Inc. 2017). The scrum master has been characterized in classic agile literature as a "*servant leader*" who is responsible for promoting and sustaining scrum within the project (Sutherland and Schwaber 2017).

With its emergence in the late 1990s, agile software development introduced self-organizing teams to software engineering (Fowler and Highsmith 2001). Self-organizing teams have been characterized as teams displaying significant autonomy in taking decisions (Cockburn and Highsmith 2001) and managing workloads and allocating work amongst themselves (Chow and Cao 2008), (Chagas et al. 2014). In addition to the development team, scrum introduced two new roles: the scrum master – primarily responsible for facilitating team functioning and removal of impediments, and the product owner – mainly responsible for representing the customer and guiding the product vision (Sutherland and Schwaber 2017), (Schwaber and Beedle 2002), (Deemer et al. 2012).

While key practitioner literature has stressed the centrality of the scrum master in agile projects (Sutherland and Schwaber 2017), studies which investigate the role of the scrum master are limited. Previous studies have investigated the tailoring of the scrum master role in enterprise level projects (Bass 2013) and the different activities such as impediment removal and ceremony facilitation carried out by the scrum master in agile projects (Noll et al. 2017). However, the above studies are limited in their scope as they leave several questions unanswered such as: *What are the everyday activities carried out by the scrum master? Is there an association between the presence of the scrum master and the frequency with which agile practices are carried out by the team? What is the extent of the scrum master's involvement in agile practices carried out by the team?* There is also a lack of quantitative data which can complement the qualitative data and lend further insight into the scrum master's role. We attempt to answer these and a critical overarching question: *What is the role of the scrum master in agile projects?*

Answering these questions allows us to address the lack of knowledge regarding the role that scrum masters play in agile projects. Such a lack of knowledge compromises not only researchers but also practitioners. For instance, organizations that look to hire scrum masters do not have a baseline to set their expectations. Given such a scenario, our goal is to produce a rich descriptive body of knowledge to enable both scrum masters and organizations looking to employ scrum masters, to have a clear view of what the scrum master actually does in agile projects. This clear perspective will be invaluable for people who are performing the role for the first time as it will enable them to set realistic expectations of their role. Even experienced scrum masters who are looking for guidance on a particular activity such as negotiation will find this study to be useful. For hiring organizations, our study will enable them to set job descriptions which are realistic and grounded in their organizational context. This will help avoid disillusionment on part of both the scrum master and the employing organization.

Thus, in an endeavour to answer some of the questions around the scrum masters role, we present the results of a Grounded Theory study with a mixed methods approach to employing supplementary quantitative data in addition to the primary qualitative data. This study on the role of the scrum master in agile projects utilized: a) Grounded Theory to

analyze qualitative data from over 45 h of semi-structured interviews with 39 software practitioners; and b) Statistical analysis such as crosstabulation and Chi-Square to analyze quantitative data from 47 questionnaire respondents who gave insights into the role of the scrum master. Our study explains how the scrum master performs six different activities – *facilitating, mentoring, negotiating, process adapting, coordinating,* and *protecting*– on an everyday basis– and has varying involvement in agile practices and can possibly influence the frequency with which agile practices are carried out by teams. To the best of our knowledge, this is the first study to present a multifaceted picture of the scrum master role.

The rest of the paper is structured as follows: Section 2 presents a background of agile software development and the role of the scrum master in the agile context. Section 3 describes the research method, including an explanation of the qualitative and quantitative data analysis procedures. The findings of the study are presented in Section 4. Section 5 presents a comparison to related work; section 6 presents evalaution of the study; section 7 presents the implications for practice and research; section 8 presents the limitations; and section 9 concludes the study.

2 Background

The following section presents a literature review which traces the introduction of agile methods in the software industry and the role of the scrum master in agile software development projects.

2.1 Agile software development

Agile software development (ASD) is an umbrella term for a set of incremental and iterative development methods such as Scrum (Schwaber and Beedle 2002), eXtreme Programming (XP) (Beck and Andres 2005), Dynamic Software Development Method (DSDM) (Stapleton 2003), Crystal (Cockburn and Highsmith 2001), Kanban (Anderson and Carmichael 2016), and Feature-Driven Development (FDD) (Palmer and Felsing 2001). ASD has a clear emphasis on people and on rapid response to change (Fowler and Highsmith 2001), (Cockburn and Highsmith 2001). In the last decades, the adoption of ASD has been extremely rapid in the software industry worldwide (VersionOne Inc. 2018), (Dybå and Dingsøyr 2008).

The popularity of scrum is reflected in the "*state of agile*" surveys (VersionOne Inc. 2018), (VersionOne Inc. 2017), where 56% of the survey respondents' projects used scrum as a standalone agile method. In the 2018 iteration of the survey, over 70% of the respondents projects used scrum in combination with other methods (VersionOne Inc. 2018). Kanban was the next most popular standalone method with 5% of respondents, while a minority (<1%) of the respondents used XP.

Each of these three agile variants has its own special focus. Scrum focuses on the project management aspects of agile such as team coordination via meetings and project planning (Sutherland and Schwaber 2017), XP focuses on development practices such as test-driven development and pair programming (Beck and Andres 2005), and Kanban is a process improvement approach visualizing work-in-progress and bottlenecks (Anderson and Carmichael 2016).

2.2 The scrum master's role

Schwaber and Sutherland (2017), the co-creators of scrum, in the latest iteration of the Scrum Guide™, split the scrum master's role into servicing three entities: 1) the product owner (PO); 2) the development team; and 3) the organization. Each "*service*" has different activities assigned to it. The scrum master serves the team by facilitating agile adoption, removing impediments, facilitating performance of scrum ceremonies, and coaching teams which are starting out in scrum (Sutherland and Schwaber 2017). The scrum master serves the PO by communicating the importance of product backlog management to the team, assisting the PO in maximizing the value derived from the product backlog, and facilitating scrum events. Lastly, the scrum owner serves the organization by leading and planning the organization's scrum adoption efforts, helping stakeholders understand scrum, and collaborating with other scrum masters (Sutherland and Schwaber 2017).

There are few empirical studies focusing on the role of the scrum master. In most studies, the scrum master is part of a larger research thrust and is discussed on the side-lines of the central topic. Some recent studies have started exploring the role of the scrum master in different industrial settings (Bass 2013), (Noll et al. 2017).

Bass' (2013) empirical study identified six overarching activities performed by the scrum master: process anchor; stand-up facilitator; impediment remover; sprint planner; scrum of scrums facilitator; and integration anchor. The process anchor ensures adherence to agile methods, standup facilitator facilitates information sharing during each sprint, impediment remover clears obstacles, sprint planner assists in the planning process prior to each sprint, the scrum-of-scrums coordinator coordinates work with other scrum masters, and the integration planner coordinates the integration of codebases by multiple teams, the latter two applying in a large-scale context.

Noll et al.'s (2017) study sought to answer the question, "*who performs the role of the project manager in a scrum team*?" They performed a review of extant literature on the subject and conducted an empirical case study of a software team. From the literature survey, they identified ten activities carried out by scrum masters. These activities include those identified by Bass (2013) and five additional activities extracted from published literature.

However, there is a lack of an overarching study which takes a holistic view of the scrum master's role in agile projects and which takes into account the different dimensions of the role. Hence, there still exists ambiguity around the role of the scrum master in agile software development as proposed in theory (Sutherland and Schwaber 2017), (Schwaber and Beedle 2002) and identified in practice (Bass 2013), (Noll et al. 2017). This ambiguity centres on important aspects such as: *the informal activities performed by the scrum master on a daily basis; the extent of the scrum masters involvement in agile practices; is there an association between the presence of the scrum master and the frequency with which agile practices are carried out by the team?* This leaves a critical gap in research, to define and describe the role of the scrum master in agile projects.

3 Research design

This section begins with a brief explanation of the key research philosophy, followed by a detailed explanation of the research methodology, and its practical application to data collection and analysis.

3.1 The research philosophy

Creswell (2012) has defined research philosophy as "*the use of abstract ideas and beliefs that inform research*". The philosophical stance assumed by the researcher can thus be accurately characterized as the lens through which the world is perceived by the researcher. There are different philosophical approaches the researcher can adopt, which include amongst others, the *positivist* approach, the *constructivist* approach, and the *pragmatic* approach. The positivist approach is based on the premise that an external reality exists which can be objectively measured (Denzin and Lincoln 2017, Easterbrook et al. 2008). Quantitative methods are typically guided by a positivist deductive approach which begins with a pre-determined hypothesis to be validated with evidence. The constructivist approach on the other hand posits that the experiences, meanings and models created by people are unique and dependent on their perspective of the context and is most commonly associated with qualitative methods (Creswell 2012). Using one, either a positivist or a constructivist approach, was not ideal for our study as our study was exploratory in nature and we wanted the flexibility to use both qualitative and quantitative methods.

In this study, we have adopted a *pragmatic* philosophical perspective as it is open to multiple methods of data collection and analysis. *Pragmatism* has also been called as the "*what works"* philosophical approach to research (Johnson and Onwuegbuzie 2004). True to its nomenclature, pragmatism gives the researcher freedom to use a combination of qualitative and quantitative methods and treats the results as valid. The roots of pragmatism can be traced back to the late nineteenth century in the principles of Charles Pierce (James and Burkhardt, 1975). However, pragmatism as a formal philosophy was articulated by Howe (1988), who presented an original thesis against the perceived incompatibility of the qualitative and quantitative approaches. Howe's central argument was that both approaches can be congruently used to achieve research objectives. This philosophical position was supported by other researchers (Brewer and Hunter 1989) who advocated the use of multiple methods to gather and interpret data.

3.2 Research method

We conducted a Grounded Theory (GT) study, into the role of the scrum master, with a mixed methods approach to employing supplementary quantitative data in addition to the primary qualitative data, within a pragmatic philosophical perspective. This was part of a larger study whose starting point was to study the role of the project manager in agile projects. The other outputs of the larger study (such as the project manager's role) are in different stages of review for publication. Our study is primarily qualitative in nature and has used GT as the overarching research method. We have incorporated a mixed methods approach to incorporating supplementary quantitative data as noted by Charmaz (2006).

Mixed methods research has been defined as, "*the class of research where the researcher mixes or combines quantitative and qualitative research techniques, methods and approaches, concepts or language in a single study"* (Johnson and Onwuegbuzie 2004). In this research, we have used GT to analyze qualitative data and descriptive statistics to analyze quantitative data. As we have further clarified in section 3.3, there is limited mixing of qualitative and quantitative data, as our study was primarily a GT study, which by its very emergent nature makes pre-determined and systematic mixing of qualitative and quantitative data challenging

(Charmaz 2006). The overarching philosophical perspective was pragmatism, most commonly associated with mixed methods usage (Easterbrook et al., 2008).

3.3 Research design

Our study utilized an *embedded mixed methods* design, where both qualitative and quantitative strands are utilized to shed greater light on the research topic. The key difference between the embedded design and other mixed methods designs is the fact that one strand is dominant (for example, the study might be primarily qualitative in nature) and the other strand plays a secondary role (Creswell and Plano Clark 2011). In our study, it was the qualitative strand which was dominant and formed the backbone of the study. While the quantitative strand informed and supported the qualitative one in some respects, it essentially played a secondary role.

Depending on the type of mixed methods design used by a researcher, the level of interaction between the qualitative and quantitative strand can vary. As our study looked to understand the intangible aspects of the scrum master's role in agile projects, it was primarily qualitative in nature. However, it was felt by the authors that diverse data sets would allow a deeper insight into the research topic (Tashakkori and Teddlie 1998), (Creswell and Plano Clark 2011). Thus the quantitative strand was embedded in the form of a pre-interview questionnaire which each interview participant completed a few days prior to the face-to-face interview. It was decided to embed the quantitative strand for the following reasons:

- The questionnaire allowed us to form a picture of the interview participant's project and the role of the scrum master in the project. This gave us the flexibility to modify the interview questions to the participant's context.
- As the face-to-face interviews were time constrained (usually under an hour), considerable time was saved by using the questionnaire to collect demographic and project related information. This allowed us to more deeply explore the participant's experience of working in agile projects.

3.4 Data collection

The first step in recruiting participants was to approach practitioners via forums such as LinkedIn and special interest groups such as Agile Auckland. Overall, 47 agile practitioners expressed interest in participating in our research and filled in a pre-interview questionnaire. Due to reasons beyond our control, out of the 47 questionnaire respondents only 39 participants agreed to be interviewed subsequently.

Table 1 shows the participant demographics for the qualitative data collection and project information for these participants. Thirty-one of these participants were from New Zealand, six from India, and one each from the USA and Australia. To ensure confidentiality, the participants have been assigned code numbers beginning with a "P" i.e. P01, P02, etc. If a participant spoke about multiple projects that they were part of, their job titles on each project and the project sector are indicated by the numbers "1" and "2" in superscripts. For example, "Dev[1]-AC[2]" indicates that the participant was referring to a developer role in an earlier project and an agile coach role in a later project. For the project sector, "1" and "2" in superscripts indicate the different sectors of the projects participants spoke about. In addition to the data

obtained from participants who were scrum masters, others such as testers and developers provided substantial information regarding the role of scrum masters in their projects. These scrum masters referred to by participants are identified by an alphanumeric code (e.g. SM1, SM2 etc.). This data which we have labelled as "*referred data*" is given in Table 6, Part B of the Appendix.

Table 1 Participant Demographics

Participant Number	Job Title	Agile Experience in Years	Team Size	Project Sector
P01	Developer	2	11–15	Banking
P02	Project Manager	<1	11–15	Local Government
P03	Project Manager	4	6–10	Telecommunications
P04	Project Manager	4	>25	Local Government
P05	Programme Manager	10	0–5	Insurance
P06	Software Product Manager	3	>25	Banking
P07	Project Manager	5	6–10	Insurance
P08	Project Manager	12	11–15	Telecommunications
P09	Senior Project Manager	5	16–20	Banking
P10	Product Owner	3	6–10	Telecommunications
P11	Programme Manager	10	16–20	Finance
P12	Scrum Master	4	>25	Local Government
P13	Developer[1]-Agile Consultant[2]	10	0–5	Accounting[1–2]
P14	Agile Consultant	10	6–10	Telecommunications
P15	Developer	4	0–5	Finance
P16	Scrum Master	5	6–10	Utilities
P17	Project Manager & Scrum Master	3	21–25	Accounting
P18	Scrum Master	1	6–10	Finance
P19	Scrum Master	5	6–10	Education
P20	Technology Consultant & (Product Owner)	4	11–15	Telecommunications
P21	Scrum Master & (Agile Consultant)	7	>25	Finance
P22	Developer	6	6–10	Finance
P23	Software Engineer & (Scrum Master)	1	6–10	Taxation
P24	Product Manager & (Product Owner)	5	16–20	Software
P25	Quality Analyst	9	6–10	Education
P26	Scrum Master	6	16–20	Entretainment
P27	Senior Director Product Management	12	6–10	Human Resources
P28	Project Manager[1]-Scrum Master[2]	10	6–10	Retail[1&2]
P29	SAP Delivery Team Manager & (Scrum Master)	8	21–25	Retail
P30	Solutions Architect	1	16–20	Retail
P31	Programme Manager[1]- Product Owner[2]	5	6–10	Tourism[1]-Healthcare[2]
P32	Tribe Lead	8	0–5	Healthcare
P33	Scrum Master[1]-Software Engineer[2]	3	6–10	Human Resources[1]--Software[2]
P34	Software Engineer	5	0–5	Healthcare
P35	Senior Test Engineer	9	16–20	Healthcare
P36	Tribe Lead & (Scrum Master)	5	6–10	Healthcare
P37	Project Manager[1]-Scrum Master[2]	2	6–10	Software[1]-Software[2]
P38	Scrum Master	4	6–10	Security
P39	Scrum Master[1]-Scrum Master[2]	9	6–10	Finance[1]-Payment Solutions[2]

3.4.1 Qualitative data collection

The qualitative data was collected from 39 agile practitioners who agreed to an interview after filling their respective pre-interview questionnaires (summarized in section 3.4.2). As described previously, the data collection and analysis process was iterative in nature, with each participant filling in a pre-interview questionnaire, followed by the face-to-face interview and analysis of a few participants' data before more was collected.

Most of the interviews were around an hour long and were conducted face-to-face, except for four, which were conducted over Skype as the participants were unavailable in person. The interview questions included questions such as: *Please tell me briefly about your professional background and your current role in this organization; What are the major challenges you have faced while working in the agile project? How did you overcome those challenges?* The full list of interview questions is provided in Part-A of the Appendix.

3.4.2 Quantitative data collection

The pre-interview questionnaire, created using Google Forms, was used to not only collect demographic data but also information on agile practices and involvement of scrum masters in these practices. A link to the questionnaire was emailed to willing practitioners at least one week prior to the interview. The responses in the pre-interview questionnaire were used to tailor the interview questions to the participant's experience and background. As this was part of a larger study which also looked at other roles in agile (such as project managers and product owners) the survey includes questions which probe the involvement of other roles in different activities. The entire survey questionnaire has been made available on Dataverse (Shastri et al. 2020).

The response options included multiple choice questions such as "*please select the relevant agile methodologies you have used*" and Likert scale type questions such as "*please rate the frequency of using the practices listed below*". There was no direct question in the questionnaire which asked respondents if the role of the scrum master was in existence in their project. This information was elicited from participants during the research interview via a more general frame setting question: "*Could you give a brief background of the project and describe your role on the project?"* The information given by the participants to this question led to further probing questions about the scrum master's role. The extent of participation and involvement of the scrum master in ten selected agile activities was gathered from the pre-interview questionnaire.

The pre-interview questionnaire generated a total of 47 valid responses. In the initial stages the survey instrument's focus was on gathering information regarding the presence or absence of the scrum master in the respondent's projects. The analysis of the initial questionnaires and interviews suggested an opportunity to inquire about the involvement of the scrum master in the listed agile practices. Therefore, this was added as a section to the pre-interview questionnaire, later in the research project after around 12 interviews and it was filled out by a total of 35 respondents. These 35 respondents are a sub-set of the overall 47 respondents. Hence, the statistical results in Table 2 are from 47 responses and in Table 3 are from 35 responses.

Table 2 Frequency of use of agile practices by the team when a scrum master is present

Agile Practice	HFU[a] as % of cases in which Scrum Master is present ($N = 39$)	HFU as % of cases in which Scrum Master is absent ($N = 8$)	Percentage Difference Level (PDL)
Definition of Done	79% ($N = 25$)	25% ($N = 1$)	54%
Iterations/Sprint Planning	64% (N = 25)	12% (N = 1)	52%
Project Velocity Measurement	67% ($N = 26$)	26% (N = 2)	41%
Short Iterations/-Sprints	75% ($N = 29$)	50% (N = 4)	25%
Team Based Estimation	66% (N = 26)	50% (N = 4)	16%
Release Planning	75% (N = 29)	63% ($N = 5$)	12%
Agile Games	37% ($N = 12$)	25% (N = 1)	12%
User Stories	75% (N = 29)	63% (N = 5)	12%
Sprint Backlog	82% ($N = 32$)	76% ($N = 6$)	6%
Retrospectives	80% (N = 31)	75% (N = 6)	5%
Iterations/Sprint Reviews	77% ($N = 30$)	75% (N = 6)	2%
Information Visualization	87% ($N = 34$)	88% ($N = 7$)	−1%
Kanban Board	69% ($N = 27$)	75% (N = 6)	−6%
Daily Scrums/Daily Standup	90% ($N = 35$)	100% (N = 8)	−10%
Product Backlog	85% ($N = 33$)	100% (N = 8)	−15%
Scrum of Scrums Meeting	43% ($N = 17$)	60% (N = 4)	−17%

(HFU[a] = High Frequency of Use.)

3.5 Data analysis

In the following sub-sections we have described the data analysis methodology and techniques applied to the qualitative and quantitative data.

3.5.1 Qualitative data analysis methodology

Qualitative research is concerned with studying the intangible and social aspects of an environment. These are the aspects that cannot be explored using quantitative methods. While the definition of qualitative methods per se is varied, we have selected Creswell's (2012) definition as it captures the essence of qualitative research: "*Qualitative research begins with assumptions and the use of interpretive/theoretical frameworks that inform the study of research problems addressing the meaning individuals or groups ascribe to a social or human problem*". There exist different methodologies to analyze qualitative data such as case study, ethnography, and Grounded Theory (GT). For our study we selected GT due to its suitability to generate a theory for software engineering and due to the inductive nature of GT as a key to uncovering the underlying concerns of software engineering practitioners. The lack of a comprehensive theory on this topic made GT a suitable research method for our study.

Table 3 Scrum master's influence on frequency of use of agile practices by the team

Agile Practice	Significant involvement (S) of the Scrum Master in the Agile practice (% of 31 cases where the scrum master was present)	Marginal involvement (MrI) of the Scrum Master in the Agile practice (% of 31 cases where the scrum master was present)	Not Relevant (% of 31 cases where the scrum master was present)
Daily Scrum	S (81%)	MrI (13%)	7%
Sprint Review Meeting	S (78%)	MrI (6%)	16%
Sprint Planning	S (71%)	MrI (16%)	13%
Definition of Done	S (68%)	MrI (22%)	10%
Project Velocity	S (61%)	MrI (26%)	13%
Sprint Retrospective	S (84%)	MrI (3%)	13%
Estimation	S(32%)	MrI (65%)	3%
Product Backlog	S(23%)	MrI (74%)	3%
Sprint Backlog	S(39%)	MrI (52%)	10%
Product Backlog Refinement	S(45%)	MrI(48%)	7%

GT was developed by two sociologists, Glaser and Strauss, in the 1960s (Glaser and Strauss 1967), (Glaser and Holton 2004), (Charmaz 2006). In recent years GT has been gaining popularity amongst software engineering researchers (Stol et al. 2016). The popularity of GT is reflected in the large number of studies which have used GT to study the human and social aspects of software engineering (Siddique and Hussein 2016), (Hoda et al. 2012), (Hoda et al. 2011), (Stray et al. 2016), (Dybå and Dingsøyr 2008), (Hoda and Noble 2017), (Waterman et al. 2015).

For analyzing qualitative data, collected via interviews, we have used Glaser's classic GT method. In the following sub-section, we have provided more details of applied GT analysis and a concrete example of the emergence of theory from data.

Application of GT to analyze data The key techniques used to identify patterns within the interview data collected from 39 participants were the GT procedures of open coding and the constant comparison method (Glaser 1992), (Glaser and Holton 2004). The software used for data analysis was QSR nVivo v.11, a software tool for qualitative data analysis. The analysis was performed by the first author in consultation and regular reviews with the co-authors. Regular consultations helped to minimize the possibility of bias introduced into the analysis by the worldview of the first author. For example, during the sorting of the theoretical memos, the memos were printed out and cut into different slips, each slip having a key theoretical memo. The theoretical memos were then spread out on a desk and simultaneously visualization was done on a whiteboard. The sorting of memos was carried out by the first author and the co-authors in a marathon session running for several hours. This exercise led to the identification of the theory of the scrum master. Another effect was

the elimination of one informal activity, that is "*translating"*, and the merging of the "*educating*" activity with "*mentoring*".

The first stage of the qualitative analysis involved sifting through and extracting snippets of data from the raw interview transcripts. The raw data was then abbreviated into a code which is usually between 2 and 5 words long. An example of data analysis from the raw data stage to the codes, concepts, category, and theory is presented below.

> **Raw data:** "*Reviews are a case in point because we have customers outside of our business hours. This idea of a review with the customer present in terms of looking at the product backlog and inspecting and adapting it in the one meeting is just not possible. Because people can't consistently give up their time with their families. We have a customer review, and we have a team review*." – P18, Scrum Master, New Zealand.

In the above example, the customer was located in the USA and the development team in New Zealand. The scrum master (P18) was alluding to the different time zones which made collaborative working a challenge.

Code Customizing scrum practices.

The code "*customizing scrum practices*" was used to capture the customization of the review practice to accommodate the distributed setup. This code shared similarities with two other codes, namely, "*hybridizing scrum with other agile methods*" and "*integrating agile with waterfall*". These codes were grouped to a higher-level concept of "*adapting agile practices*".

Concept Adapting Agile practices.

A considerable number of participants were found to customize and integrate elements of different agile methods into scrum. The agile practices most frequently customized included sprint retrospectives and sprint reviews, the product backlog, and sprints. In the case of the "*Process Adapting*" category, there was only one underlying concept. However with other categories, there were multiple concepts which were grouped together to yield the overall category.

Category Process Adapting.

Other categories which emerged in a similar way included *facilitating, mentoring, negotiating*, *coordinating*, and *protecting*. For example, the category of facilitating was made up of three key concepts, namely, "*facilitating minesweeping*", "*facilitating team functioning*", and "*facilitating processes*". Together they represent the 'everyday activities' of the scrum master.

The same procedures of coding and constant comparison were applied again, focused on the strongest emerging categories, a process referred to as selective coding.

Memos are a very important part of doing GT research which enable the researcher to discern conceptual links between the seemingly disparate categories. Memos are a lithic record of the researcher's decisions throughout the research process (Glaser 1992). Birks et al. (2008) have opined that memoing leads to the establishment of a very intense relationship between the researcher and the data. Memos were extremely useful during the last phase of data analysis when diverse pieces of information had to be tied together. In our research, we observed the benefits of writing memos as pointed out by researchers (Birks et al. 2008). Initially, memo writing was a difficult exercise as the researcher was not familiar with the concept of memos. The language of the early memos was terse and brief. However, with time and practice, the memos

grew in quality and quantity. One of the extremely fruitful application of memos was in the creation of "*participant bios*". These were one paragraph memos which concisely summarized the overall project environment of the participants and their location within that environment.

Theoretical sorting or the sorting of memos is a crucial step as it leads to the emergence of theory. Glaser and Holton (2004) have explained that "*GT generates the outline through the sorting of memos by the sorting of the categories and properties in the memos into similarities, connections and conceptual orderings*". Following Glaser's advice, we performed sorting as a research team on a conceptual level and not simply a chronological ordering of memos.

After the sorting of the memos, we carried out a Grounded Theory analysis process known as *theoretical coding*. Glaser has explained this as the process of using a theoretical structure to visualize the relationships between the categories and has suggested several theoretical coding families (Glaser 1978) such as: *the six Cs family, the model family, the process family, the strategies family*, and *the dimensions family*. All of these families have been utilized by software engineering researchers to explain the human aspects of ASD (Hoda et al. 2011), (Stray et al. 2016), (Hoda and Noble 2017), (Waterman et al. 2015).

Theory The role of the scrum master in agile projects.

The theoretical coding family best suited to our findings was the dimension family (Glaser 1978), (Glaser and Holton 2004) which enabled the findings to be presented as facets or aspects of a phenomenon. Glaser has visualized the dimensions family as the decomposition of a phenomenon into different pieces (Glaser 1978). Hence, the dimensions family is a useful fit to describe a phenomenon that has different facets, dimensions, aspects or properties (Glaser 1978).

Our theory is described in detail in section 4. The pre-interview questionnaire and interview data were analyzed together for each participant. Using theoretical sampling, data collection and analysis were iterative and interleaved, a set of questionnaire and interview data being analysed before more data was collected and so forth. A holistic view of the scrum master role was achieved by including practitioners in different roles including not only scrum masters and managers but also developers, testers, and senior management. The datasets generated and analyzed during the current study are not publicly available due to confidentiality reasons but are available from the corresponding author on reasonable request.

The last three interviews helped confirm the emerging findings and it was felt that the emergent theory was capable of explaining the phenomenon without the need for fresh data. This was an indicator of theoretical saturation (Glaser 1992) and thus concluded the qualitative data collection phase. For ease of presentation, we describe data collection and analysis for the primary qualitative data first, followed by that for the supporting quantitative data. In practice, they were iterative and interleaved.

3.5.2 Quantitative data analysis

The major part of quantitative data analysis revolved around the data collected from the pre-interview questionnaire. Some of the qualitative data was coded as quantitative data. This is discussed in the following sub-section.

Analyzing the questionnaire data The first step was to apply consistent labeling to the free-text responses in the pre-interview questionnaire and comparing them to the interview

transcripts to cross-check for conflicting information. There were only two questions with free text fields: a) "*What sector or domain the project was in*?"; and b) *What was your job title on the project/product*? Responses to question a) elicited a range of answers from the survey respondents, sometimes the same project domain was described with different spellings. For example, the project sectors could be described by different respondents as both *finance* and *financial* in the free text response. This was rationalized and a uniform label of *finance* was applied in this case.

Once the questionnaire data was validated and its integrity verified by running Cronbach's Alpha test, it was transferred into IBM SPSS v.24. The two survey questions which were tested for reliability were:

a) What is the frequency of agile practices carried out by your team?
   The above question had 16 practices which included definition of done, sprint retrospectives, and sprint reviews. The complete list along with results of the data analysis is presented in Table 2.
b) Please indicate the involvement of the scrum master in the selected agile practices.
   In this question, the survey respondents were presented with a filtered list of 10 agile practices such as sprint backlog, estimation, and the daily scrum.

The test scale of Cronbach's Alpha goes up to 1. In terms of scale reliability values over 0.8 are considered good while over 0.9 are very good (Tavakol and Dennick 2011). The Cronbach's Alpha is around 0.955 for question a) which asked participants to grade the involvement of scrum masters in agile practices. While for question b) which asked respondents to rate how frequently selected agile practices were used in their projects, the Cronbach's Alpha is 0.895. We used the descriptive statistical technique of cross-tabulation as it was found to be most suitable for analyzing Likert scale data. To see if the results were statistically significant the Chi-square test was also done on the data. The Chi-square test results are provided in Tables 4 and 5 in Part B of the Appendix.

One of the sections asked the survey participants about the frequency of use of common agile practices such as daily scrum, user stories, and retrospectives which were derived from the list of common practices used in the state of agile surveys (VersionOne Inc. 2017), (VersionOne Inc. 2018). The full list of practices can be seen in Table 2. These had five response options: *very frequently; frequently; occasionally; not used*; and *not applicable*. For analysis purposes, we combined the results of "*frequently*" and "*very frequently*" into a higher-level classification "*High Frequency of Use (HFU)*". The cross-tabulation was run against two key questionnaire items: the identified presence of the scrum master on an agile project and the frequency of use of a set of common agile practices.

The Percentage Difference Level (PDL) is derived by the following formula:

PDL = (HFU as % of cases in which SM is present) - (HFU as % of cases in which SM is absent).

A positive PDL value indicates the frequency of a practice increases when the scrum master is present and a negative PDL value indicates the frequency of a practice reduces with the presence of the scrum master. These results are discussed in the findings section 4.2.

Coding the qualitative data as quantitative To gain a snapshot view of what percentage of scrum masters were carrying out a particular activity (such as mentoring or facilitating) we aggregated two separate pieces of information: a) Data gleaned from interviews with 39

participants (labelled as "*participant data*") of which 17 were scrum masters; and b) 9 scrum masters whose role was mentioned by participants during the interview (labelled as "*referred data*"). These referred roles were scrum masters involved in the participant's projects. This aggregate data was then used to calculate the percentage of scrum masters involved in an everyday activity. The simple calculation was to divide the number of scrum masters engaged in an activity by the total number of scrum masters. The referred data and the steps used to derive the percentages are given in Tables 6 and 7, Part B of the Appendix.

4 Findings

We have described the findings of our study in two separate sub-sections. Section 4.1 presents the findings from the qualitative strand, the key output of which is the grounded theory of the role of the scrum master in agile projects. The role of the scrum master has been visualized in terms of "*dimensions*" which are the different constituent parts of the scrum masters role.

Section 4.2 presents the findings of the quantitative strand. Results of inferential statistical tests such as cross-tabulation and Chi-Square are given in this section along with the researchers' interpretation.

4.1 Findings of qualitative data analysis

The theory which emerged from the GT analysis was the "*The role of the Scrum Master in agile projects*", in which the role of the scrum master in agile projects is described in terms of the following dimensions, that is, the everyday activities of *facilitating, mentoring, negotiating, coordinating,* and *process adapting*, performed by the scrum master. The different dimensions of the scrum masters role have been visualized in a honeycomb structure and are presented in Fig. 1. Each hexagonal cell represents a particular activity (for example, facilitating) and the arrangement of the cells around the central hexagon symbolises that each activity is a facet of the scrum master's role. These activities are discussed in detail in the following section.

4.1.1 The scrum master's everyday activities

The findings presented in this section are a result of the qualitative data analysis (described in section 3.5.1). Figure 2 shows the mapping of the scrum master's involvement in different activities. The mechanics of the calculation and the underlying data are given in Table 7, Part B of the Appendix.

Overall, it was seen that a majority of scrum masters were involved in facilitating and mentoring. Close to 69% of the scrum masters' in our study performed facilitation and mentoring. While around 31% were involved in process adapting, little more than a quarter were involved in negotiating and protecting, and few were involved in coordinating. If a scrum master was present in the project, there was a very high possibility of the scrum master participating in facilitation and mentoring activities. All these activities have been described to present a complete picture of the scrum master's role.

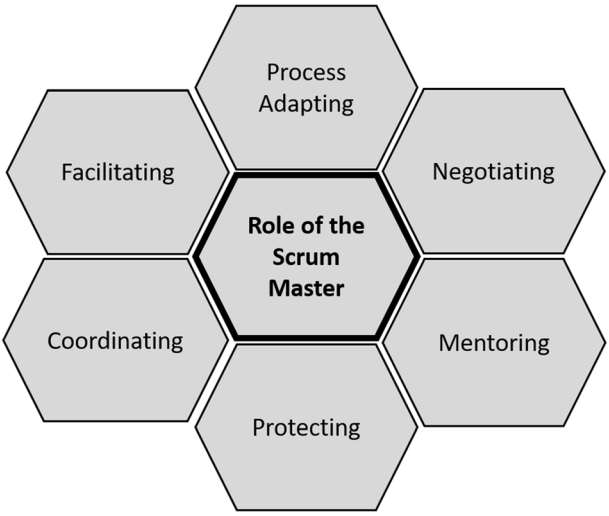

Fig. 1 The role of the scrum master in agile projects presented as dimensions

Mentoring Mentoring involves the scrum master empowering the team on the path of self-organization, educating the team and stakeholders in agile practices, and ensuring team adherence to agile practices. Nearly 69% of the scrum master's performed mentoring as part of their role.

To borrow a term from quality management, the scrum master's approach was a *pull* instead of a *push* approach. Some typical adjectives used by the interviewed scrum masters include: "*observing*"; "*explaining*"; "*coaching*"; "*mentoring*"; "*discussing*"; "*injecting*"; "*educating*"; and "*engaging*".

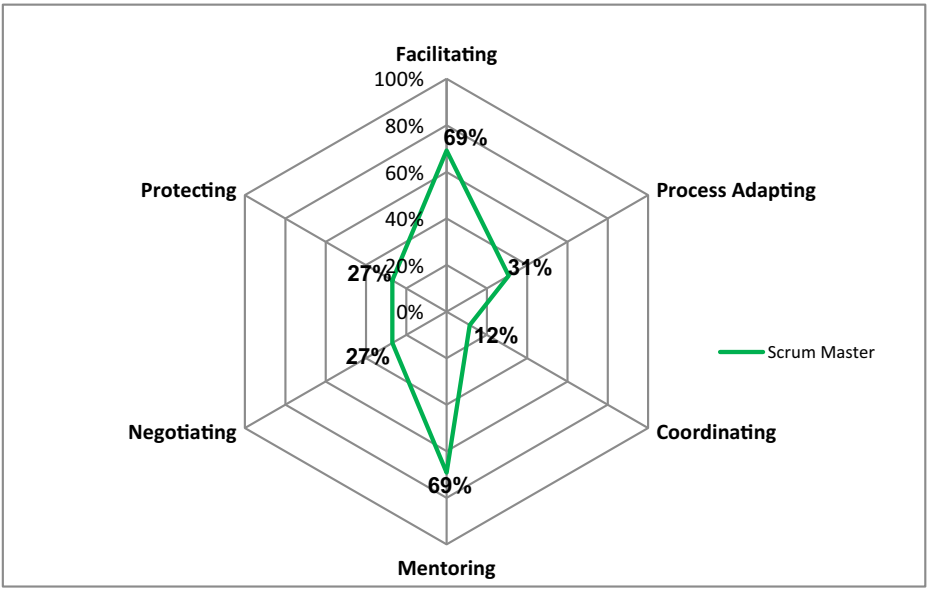

Fig. 2 Mapping the scrum master's involvement in everyday activities

The scrum master's role was well represented in all three key concepts which constituted the mentoring activity, which is, educating stakeholders and team members, ensuring adherence to agile practices, and empowering the team.

The emphasis on minimal hand-holding and minimal interference in the team differentiates the scrum master's mentoring activity from other activities such as facilitating and coordinating. When mentoring, the scrum master's thrust is towards letting the teams solve issues and become more confident in the use of agile. As one scrum master (P21) put it, he was in "*service to the organization*".

None of the interviewed scrum masters saw themselves running agile practices such as daily standups. Instead, they saw their role as observers, who came into the picture only when the team needed to be pointed in a direction or when they ran into a roadblock. The daily stand up was an opportunity to coach the team in agile. Thought-provoking questions were used by the scrum master (P18) to get the team's creative juices flowing. Team members were encouraged to communicate during the daily standup (P16). The scrum masters were the metaphorical grease that drove the wheel and their self-perception was such as well. Both old and new team members were engaged by the scrum master and the focus was on moving ahead with everybody's consent and collaboration.

Irrespective of the sector and type of project, the first step the scrum masters took with any team was to set the frame of reference for agile (P21, P23, P33, P36, P37, P38, and P39). The context was set by making teams understand why they were using agile and its advantages. Convincing teams of the benefits was a better approach than pushing them into agile practices. The frame of reference was usually set in a training workshop dedicated to agile. To take an example, a scrum master (P21) first set the frame of reference by training the team in agile and then let the team run the agile process. P21 only came into the picture when requested by the team to help them in clearing an obstacle, which in this case was the poor accuracy of estimates. Similarly, an agile fundamentals training exercise was conducted by the scrum master (P18) in a financial sector project.

"*With that first team we did a one-day scrum fundamentals course. We took the team out for a day, away from their work for a day, myself and another agile coach, and we went through the fundamentals. It was part team building, and part sort of education*." – P18, Scrum Master, New Zealand.

The scrum master held the team members accountable for their commitments. A sense of "*team play*" was instilled in team members. In the case of an entertainment sector project, the scrum master (P19) got the team members motivated enough to volunteer to take up difficult tasks. Scrum masters (P21, P36) also engaged in one-on-one coaching to correct any deviations from agile practices. Occasionally, the scrum master (P38) had to take an assertive stance to bring the team back on track.

> "*I educated the team not to do that, and I am also monitoring that graph. This is your graph, not mine. You have to make sure this graph stays in a good shape. If it looks ugly, you have to do something, not me.*" – P38, Scrum Master, India.

Facilitating Facilitating involves the scrum master clearing obstacles and issues, facilitating the project teams functioning, and easing the team's transition to scrum. Close to 69% of the scrum masters in our study performed facilitating.

The phrases and keywords used by scrum masters which illustrated this approach included: "*assist team"; "keep container healthy"; "bringing business together"; "collaboration"; "reach out"; "suggested";* and *"sorting out blockers"*.

The scrum master's role was summed up very well by an interviewed product owner (P06), who said, "*Scrum masters by nature tend to be facilitators*". This was supported by other participants (P15, P24) who saw facilitating as the scrum master's key everyday activity (SM4, SM6).

The scrum master's attitude was to let the team solve problems themselves (P16, P18, P21, P26, P36, P37, P38, and P39). P16 saw his role as removing bottlenecks and encouraging the team to take the initiative in problem resolution. P16 only stepped in when he saw that the team's focus needed to be channelled in a particular direction. In this case the team's excessive reliance on email communications with the vendor was causing delays, as the vendor often did not respond to emails. P16 nudged the team to call the vendor whenever they ran into such issues. The organization was experimenting with agile and had given P16 complete freedom to implement agile.

> "*The whole process would become protracted and drawn out. My attitude would be, have you rung up the vendor, have you identified who the main contact person would be for us there and if so let's just get into a room, ring up that person and find out an answer, instead of waiting*." – P16, Scrum Master, Australia.

On a payment solutions project, the scrum master (P39) only came into the picture if the delivery timelines were not being met. In this scenario, P39 helped the team in splitting tasks into smaller bits. This self-perception of the scrum master's role was also confirmed by a software engineer (P34) who was emphatic that the scrum master's involvement was on a guidance level. It was the team which had to take the initiative.

Depending on the project context, the scrum master (P18, P21, and P23) could be responsible for facilitating across multiple teams. For example, P18 was facilitating three teams, one each for development, reporting, and operations. P18 saw his role as keeping the development environment "*healthy*". The scrum master wanted the team to do work and enjoy it as well.

Several organizations had adapted scrum to their own context. Similarly, P21 only stepped in when asked by the team for help. This was to get the team disciplined in agile practices and to resolve a conflict simmering within the team. The conflict between two team members reached a stage where they refused to work with each other. The scrum master (P21) defused the conflict by arranging multiple tripartite meetings where a compromise formula was formulated.

> "*We had multiple meetings where we sat down with each other trying to understand what each other want, and how they want to work together. And what are the rules for them to work together*." – P21, Scrum Master, New Zealand.

If the team was transitioning from another methodology, such as from lean to scrum, the scrum master (P33) helped ease the transition. This involved helping the team understand and get disciplined in scrum practices such as estimation. P33 actively helped facilitate estimation.

Process adapting Process adapting involves tailoring processes to suit project context by customizing agile practices and by integrating traditional project management practices with agile practices.

Compared to other informal activities such as facilitating and mentoring, the process adaptation activity is less about people management skills and more about customization of agile practices. The process adapting category can be further classified into two types:

- *Proactive*: The adaptation was motivated by the observations of the scrum master regarding the value of an agile practice or based on the team's suggestions. This approach has been termed as the "*proactive*" approach.
- *Reactive*: The adaptation was in response to an external stimulus either from the project environment or from external customers. This approach has been termed as the "*reactive*" approach.

Proactive In a proactive approach, an adaptation is made in response to feedback coming from within the team. It could also be the scrum master concluding that certain practices are not working in the team's context and proactively deciding to adapt them.

A good example is the adoption of Kanban boards as task visualization tools. In most cases, the scrum board was dropped as the team found the Kanban board to be a better visualization tool. The Kanban board was ideal to display the entire product backlog in a single location. It was also used to display the tasks within a sprint. These adaptations were carried out due to the team's feedback. However, the use of the Kanban board could also be dictated by reactive pressures, discussed in the next section.

Based on experience and observations, the scrum master carried out innovations in different agile practices. One example was that of a "*snap demo*" created by a scrum master (P17). The snap demo was an ideal forum to enhance customer and team interaction. The snap demo was held weekly and was very informal, with the customers sitting next to the developers and seeing their work, as opposed to a formal demo.

Another adaptation by a scrum master (P09) was to reformat the retrospective into a "*start, stop and continue*" format. This involved the team giving suggestions on what new elements could be added to the process ("*start*"), what elements needed to be removed ("*stop*"), and what should be continued ("*continue*"). P09 realized that the retrospectives were not eliciting useful comments from the team. The introduction of a "*start, stop and continue*" type of retrospective resolved the issue, as team members were motivated to contribute meaningful suggestions. This change was motivated by P09's observation that team members were not contributing any useful suggestions during the retrospective. The change had the desired effect and involvement from the team in the retrospectives improved.

Another scrum master (P21) turned around a redundant sprint review meeting into a forum to enhance communications within the team. As the team did not practice two weekly sprints, there was no need for customer presence in the sprint review meetings. Hence the sprint review meeting had nearly become obsolete, till P21 refocused it.

Reactive In contrast to the proactive approach, the reactive approach is a response to constraints imposed by the project environment and factors outside the realm of the team's influence. The scrum master is essentially reacting to constraints imposed by the organizational structure and processes. This is usually the case with hybrid type of organizations where agility at the team level is enclosed in a larger traditional and waterfall shell. The '*water-scrum*' arrangement led to pressure on the scrum master to try and mould team-level agile practices into a project-level waterfall context. This could be in the form of trying to map sprints onto

milestones, translate agile metrics into traditional project management metrics, or generating regular reports for senior management.

Organizational setup could mean that the scrum master (P26) had to facilitate multiple teams at the same time. This was logistically impossible and led to P26 giving teams independence to run their own retrospectives without P26's presence. P26 tried to attend alternate retrospectives with different teams. Again, this was an adaptation necessitated by organizational constraints rather than being a voluntary choice of the scrum master.

Negotiating Negotiating involves the scrum master aligning customer expectations with the team's actual capacity to deliver working software. From our interview data, around 27% of the scrum masters had a role in negotiation. Hence, the discussion here is brief and limited to the inferences drawn from interviews with seven scrum masters who reported negotiations.

The scrum master's approach to negotiation was characterized by terms such as, "*achieve a trade-off*", "*discuss*", and "*question*". Nearly all the scrum masters identified that it was the product owner who was really in charge of customer-facing and team-facing negotiations. However, on occasions where the product owner was stepping into the scrum master's territory, the scrum master (P39) firmly but politely delineated the roles. In projects where the project manager was present and pushing for adding to the team load, the scrum master (P39) negotiated with the help of historical data. If it was the product owner trying to add extra load, the scrum master (P38) made it clear that tasks would be added only if an equivalent number of tasks were subtracted from the product backlog.

The general tenor was that the scrum masters shared a good working relationship with the product owners. Most of the issues around features and requirements were resolved in meetings with the product owner.

Protecting The protecting activity involved shielding the team from external interference and pushing back on scope creep. Protecting the team from external interference was an interesting aspect of the scrum master's role. For most participants, the scrum master did not manage or control the team nor did the team report to the scrum master. Reporting was usually to a managerial role such as a team lead or a development manager. The situation was further complicated if team members were assigned part-time to the project, in addition to their normal work. The scrum master had to find ways of pushing back on interference and scope creep even in the absence of formal authority.

Even in companies with a very flat organizational structure where the team was highly self-organizing, business needs would take priority over the team's current commitments. Senior management would try to get the team to work on tasks not part of the backlog. In this scenario, the SM4 pushed back and made it clear that the team was already committed to delivering other tasks.

> "*A CTO would come in and say I know you guys have planned for the sprint. But I would want you guys to also work on something else, which you don't account when you create stories. That's where the scrum master [SM4] would come. He would come and he would step in and say no, we can't do it because we have committed these others, and these many tasks for this sprint.*" – P15, Developer, New Zealand.

This boils down to one key point: the scrum master must adopt a hard posture when facing external pressures from senior management and customers. This is exemplified in multiple cases where the scrum masters (P16, P23, P26, P38, P39, and SM4) protected the team. The

scrum master (P23) compromised only in those cases where organizational constraints and business needs meant some tasks had to be taken on board. Even in this scenario, damage control was done by allocating a single team member to the external tasks.

Coordinating The two key coordinating activities carried out by the scrum master were involving specialists in the project and coordinating collaboration between teams from different areas of the organization. This activity involved coordinating collaboration between multiple teams and involving technical specialists in the project.

In the coordinating activity, the scrum master had minimal involvement. Only 12% of the interview participants suggested that the scrum master carried out some kind of coordinating activity. The scrum master (P16) of a utilities project explicitly attributed successful project completion to the involvement of specialists. In addition to the core scrum team, P16 brought in a solutions architect and a security architect when necessary. Their involvement helped in meeting the compliance requirements of the organization.

> "*We wouldn't have got the project across the line without having involvement and validation of the approach the team was taking. There was still a certain amount of compliance that we had to produce to the business to get it across the line, so things like load testing, non-functional testing. We had to produce sign off document for that.*" – P16, Scrum Master, Australia.

In a financial project, the scrum master (SM4) played a key role in facilitating collaboration between teams from the platform and development arenas.

"*Deployments are a big issue. No one considers it. Need to talk to the platform team and need someone to facilitate this. You express your need, desire, to talk to people and this is where the scrum master comes in*." – P15, Developer, New Zealand.

4.2 Findings of quantitative data analysis

In this section, we present the results of the quantitative analysis. Section 4.2.1 looks at the possible correlation between the scrum master being present in an agile project and the frequency with which an agile practice is carried out by the team. Section 4.2.2 deals with the level of involvement a scrum master has in selected agile practices carried out by the team.

4.2.1 The scrum master and the prevalence of agile practices

One of the key aspects we wanted to dig deeper into was to see if there was any association between the presence of a scrum master in the survey respondents agile projects and the prevalence of an agile practice. To this end we ran a crosstabulation between two survey categories: a) *The presence of the scrum master in the respondents project*; and b) *the frequency of agile practices carried out by the team.*

The results of the cross-tabulation analysis (the process has been described earlier in section 3.5.2) are summarized in Table 2, where the last column, Percentage Difference Level (PDL), shows the difference between the High Frequency of Use (HFU) percentages in the second and third columns. A positive value of the PDL indicates that if a scrum master is present, there is an increase in the frequency of an agile practice. Whereas a negative PDL value indicates that the frequency decreases when a scrum master is present. A small positive or negative value (+6% to −6%) indicates that the presence of the scrum master has little effect.

Nearly 83% ($N = 40$ of 47) of the pre-interview questionnaire respondents confirmed that the scrum master was present in their projects.

The practices which seemed to exhibit a notable increase in their frequency of use when a scrum master was present were the *definition of done, iteration (sprint) planning, project velocity measurement,* and *short iterations (sprints)*. These four agile practices showed a positive PDL of 41 to nearly 54%. The next logical step was to assess if the percentage differences were statistically significant. To this end, we carried out Pearsons Chi-Square test, which demonstrated that differences for the above mentioned three practices are statistically significant. This was indicated by their Chi-Square significance values which were below 0.05. For the remaining practices, the differences are not statistically significant. A full list of the Chi-Square values along with additional statistical information is presented in Table 2 in Part B of the Appendix.

The scrum master's role in our study was found to focus largely on facilitating and mentoring (based on qualitative data analysis). The scrum master facilitated the teams transition to agile by encouraging them to take ownership of agile practices such as team-based estimations. Another key aspect of the scrum master's role identified in the qualitative phase of our study was to make sure that the team implemented agile practices and performed them regularly. This explains the high positive upswing in the above-mentioned agile practices.

There does not seem to be a statistically significant association between the scrum master's presence and the frequency of use of the remaining thirteen agile practices such as *sprint backlog, retrospectives, iteration/sprint reviews, information visualization*, and the *kanban board*. A possible explanation is that as teams' transition towards self-organization and take ownership of agile, the scrum master's hands-on involvement is reduced. Thus, in this scenario, it makes little difference if the scrum master is physically present in the day-to-day implementation of agile practices. Another possible explanation is the small sample size of respondents, a larger sample size might yield different results.

What is interesting to note is that agile practices, such as *daily scrums (daily standups)* and the use of the *product backlog*, exhibited a moderate negative PDL of 10 to 15%. The presence or absence of other factors may be influencing this observation. We would like to point out that, from our data and results of analysis, a causal relation cannot be inferred between the presence of the scrum master and the adoption of a particular agile practice. The aspect of causality merits further in-depth investigation.

4.2.2 The scrum master's involvement in agile practices

The survey question, "*what is the extent of the scrum master's involvement in agile practices carried out by the team?"* was answered by 35 respondents as this was added at a later stage of the study. Out of the 35 responses, the scrum master was present in 89% ($N = 31$) of the responses and not present in 11% ($N = 4$) cases. Table 3 lists the agile practices in which the scrum master had significant involvement and those in which there was marginal involvement. The ten scrum practices, against which the involvement of the scrum master was mapped, were selected based on their similarity to project management practices. The results discussed below are the output of the cross-tabulation analysis and the percentages specified are out of 31 respondents where the scrum master was present.

In the pre-interview questionnaire, the survey participants were presented with four levels of role involvement to select: *no involvement, somewhat involved, high involvement,* and *key decision maker*. For analysis, the categories of "*no involvement*" and "*somewhat involved*" were combined and the category was labelled "*Marginal Involvement (MrI)*".

The categories of "*high involvement*" and "*key decision maker*" were combined to a super-category of "*Significant Involvement (S)*". This aggregation of categories allowed us to present a more cohesive picture of the findings. The percentages in round brackets in Table 3 indicate the percentage of respondents who indicated that the scrum master either had significant (S) or marginal involvement (MrI) in the agile practice. For example, in the case of daily scrum, S (81%).

indicates that in case of 81% of the respondents, the scrum master had significant involvement in the daily scrum. The percentage differences for the involvement of the scrum master in the agile practices listed in Table 3 are statistically significant as well. The Chi-Square 2-sided significance (p value) for all the practices was less than 0.05. A full list of the Chi-Square values along with additional statistical information is presented in Table 1 in Part B of the Appendix.

As shown in Table 3, the scrum master's role closely follows the definition laid out in the scrum guide (Sutherland and Schwaber 2017). Activities such as sprint reviews, sprint planning, and daily scrums showed significant involvement by the scrum master. While activities such as product backlog management which are the domain of the product owner (Sutherland and Schwaber 2017) saw marginal involvement by the scrum master. This is keeping in line with practitioner literature and suggests that as agile matures in the software industry; a clear delineation of scrum roles is emerging.

5 Comparing to related work

The wide spread of the scrum master's role as documented in our study is supported in the findings of recent empirical studies (Hoda et al. 2013), (Bass 2013), (Noll et al. 2017), where the scrum master's activities encompassed both software development and project management activities.

Of the different scrum master activities listed by Bass (2013) and Noll et al. (2017), it is that of impediment removal which is the closest match to the facilitation activity of the scrum master we identified. Our finding that the scrum master's highest involvement is in mentoring activities is supported by other studies (Bass 2013), (Noll et al. 2017). As far as we are aware there have not been any studies which detail the protecting activity.

The *Mentor* role presented in Hoda et al.'s (2013) study has significant overlap with our mentoring activity. The *Mentor* role in their study was typically played by a scrum master or an XP coach depending on the agile method being implemented. The key function of their *Mentor* role was to help the team implement agile (Hoda et al. 2013). This corresponds with our mentoring activities such as empowering the team on the path of self-organization and educating the team. Another point of intersection is their role of the *Coordinator* (Hoda et al. 2013), which has aspects of our coordinating and negotiating activities. However, their *Coordinator* role was performed mainly by developers and business analysts. While in our study, the coordinating and negotiating activities formed, however small, part of the scrum master's role.

A more recent study by Hoda and Noble (2017) presented a grounded theory to explain the transition of teams to agile. Their theory elucidated the transition occurring over a period of time across five dimensions: software development practices; team practices; management approach; reflective practices; and culture. In their "*transitioning management approa*ch" as teams became more autonomous, managers adopted an empowering approach, relegating themselves more to the role of mentors,supporters, and obstacle clearance. This finding supports our view of the scrum master as a mentor and a facilitator. Hoda and Noble's (2017) *team practices* dimension

presented teams taking ownership of agile practices such as requirements specification, clarification, prioritization, and estimation. This is supported by our qualitative and quantitative findings, which shows that the team is actively adopting an agile approach.

Martin et al's study (2010) presented the role of the *Technical Liaison* as the person responsible for coordinating between different teams and looking after the involvement of specialists. This corresponds to our coordinating activity. However, in Martin et al.'s study (2010) the technical liaison role was performed by the project manager and the agile coach.

There are certain areas where our findings mirror the recommendations made for the scrum master's role in seminal practitioner literature such as the Scrum Guide™ (Sutherland and Schwaber 2017). Some of the key functions of the scrum master role (as mentioned in the Scrum Guide™) such as coaching, facilitating, ensuring scrum artefacts and events are followed, and removing impediments, are supported by our activities of *mentoring* and *facilitating*. However, our activities of *process adapting* and *negotiating* do not find a mention in the Scrum Guide™. This can possibly be due to the Scrum Guide™ assuming that the scrum implementation is relatively free of the constraints imposed by organizational structures. An interesting point is that the Scrum Guide™ envisions the team negotiating with the product owner regarding the amount of work and product backlog items (Sutherland and Schwaber 2017). However, from our study, it is probably the scrum master in most cases who is negotiating with the product owner. Our protecting activity is supported by the Scrum Guide™ when it says that, "*If others are present, the Scrum Master ensures that they do not disrupt the meeting"*(Sutherland and Schwaber 2017). However, forestalling interference from senior management is an aspect that is unique (the *protecting* activity from our study). Another point of divergence from the Scrum Guide™ is the development of definition of done (DoD). The Scrum Guide™ recommends that the team should be the key player in defining DoD (Sutherland and Schwaber 2017). However, our quantitative analysis reveals that the frequency of this practice increases when the scrum master is present (see Table 2). This suggests that the scrum master is involved in this activity. This is also supported by our finding (see Table 3) that the scrum master has significant involvement in defining DoD.

Our study also presents multiple perspectives of this role beyond everyday activities such as the scrum master's involvement in and possible influence on agile practices, thus providing a rounded perspective on what it means to be a scrum master for the practitioners playing this role in practice and for their teams.

6 Evaluation of the study

In the sub-sections below we present the evaluation of the qualitative and quantitative strands of our study.

6.1 Evaluation of the grounded theory

Foundational texts of Grounded Theory list four criteria for the evaluation of a GT study. These are: fit; work; relevance; and modifiability (Glaser 1992).

Fit & Relevance Glaser and Strauss have emphasized that the grounded theory must fit the area under study (Glaser and Strauss 1967). This means the generated theory should closely reflect the data collected from participants. In our case, the interim results of our study were presented at academic venues (Shastri et al. 2017) and parts of the results were positively received at multiple practitioner-based events.

Work The codes and concepts which emerged from our analysis were closely related to the concerns of the participants, which is, regarding the role of the scrum master in agile projects. The workability was assessed by frequent discussions with the research supervisors.

Additionally, while our theory is specific to our research context, it can act as a predictor tool for the scrum masters role in ASD projects. For example, using our theory we can say that, depending on the project context, a high percentage of the scrum master's role will involve facilitating and mentoring. Negotiation, process adapting, and protecting will also form a not-so-insignificant part of the scrum masters everyday activities.

Modifiability Our grounded theory was modifiable right up to the last stage where it gained its final form. A good example of modifiability is the gradual aggregation and elimination of the large number of codes generated from the open coding process. In open coding, 119 codes were generated, which by the time of the final write-up of the theory had come down to 33. Additionally, our theory continues to be modifiable with future research.

6.2 Evaluation of the quantitative data

Although GT was the key analytical method for qualitative data, some of its key precepts especially those of iterative research were implemented in the quantitative strand as well. Thus, while the pre-interview questionnaire was created prior to the commencement of the qualitative phase, it did not remain static. Content validation was carried out by testing the questionnaire by the authors and then modifying it an iterative manner based on the respondents feedback. The reliability of the questions was measured using Cronbach's Alpha (see section 3.5.2.1).

Sampling bias was minimised by contacting a diverse sample of software professionals during the data collection phase. This covered not only scrum masters but also testers, project managers, and developers. Thus, out of 47 survey respondents, 17% had the job title of scrum masters, whereas the rest were a diverse cohort of job titles. In the qualitative data collection, the percentages are different as participants chose to talk about multiple projects in which they quite often had acted as scrum masters.

The possibility of response bias was minimised by the way our study was structured. In a lot of cases, the survey respondents went ahead and participated in the interview. This allowed us to double-check their responses in the interview and make corrections as appropriate. For example, one respondent had incorrectly rated the scrum masters involvement as "*somewhat involved*" whereas the scrum master was not involved at all in that activity. This was manually corrected in the survey data after the interview. However, due to the fact that we were not able to interview quite a few of the survey respondents the possibility of response bias cannot be completely ruled out.

7 Implications for practice and research

Our study highlights the diversity of informal activities carried out by the scrum master, the varying involvement of the scrum master in agile practices, and the positive association between the scrum master's presence and the frequency with which agile practices are carried out by the team. Depending on the organisational context, the scrum master can perform any of the six everyday activities.

7.1 Implications for practice

Our findings will be very useful for both scrum masters and organizations which are looking to hire scrum masters. For organizations the range and depth of the scrum masters activities gives them a good guide to build a realistic job description when advertising for scrum master roles. It also gives management an understanding of the different, often contradictory, pressures of the scrum master's role. For scrum masters, especially those who are new or transitioning into the role, the everyday activities will help set realistic expectations of their role. These activities are:

Mentoring Scrum masters will need to set the frame of reference for teams who are new to agile. This will also involve conducting regular coaching to ensure the team internalises the agile way of working.

Facilitating In organizations following a hybrid methodology or those transitioning to agile, the scrum master will need to assist the teams going through the transition and let the team take the initiative to resolve issues, rather than the scrum master doing it for them.

Process adapting Depending on the context, the scrum master will need to be ready to incorporate new techniques or processes from other methodologies. A pragmatic stance is advised in this aspect, with a focus on deliverables rather than processes.

Negotiating The scrum master will need to keep their negotiating skills sharp as they may be called upon to conduct informal negotiations with other entities such as the product owner.

Protecting The scrum master will have to assert against possible interference from other teams and senior management. This is very important as continuous interference will impair the team's productivity.

Coordinating While coordinating is a small aspect of the scrum masters role, in some projects, especially those that involve multiple teams or even specialists, this activity can assume a lot of importance.

While all of the above mentioned activities add to the scrum master's role, it is the activities of mentoring and facilitating which are the core activities. As can be seen from Fig. 2, 69% of the scrum masters in our study performed facilitation and mentoring. These activities in turn involve (amongst other sub-activities) empowering the team on the path of self-organization, educating the team and stakeholders in agile practices, clearing obstacles and issues, facilitating team functioning, and easing the team's transition to scrum. For organizations looking to recruit scrum masters these sub-activities can form the

basis of a realistic and accurate job description for the scrum master. Each sub-activity can be translated into its driving skillset. For example, for clearing obstacles, a key skill needed is the ability to think outside the box. Similarly, a higher level to act as an effective mentor, the scrum master should have the ability to inspire trust amongst team members. While other attributes such as the level of education and personality type are important factors in the recruitment of scrum masters, it is the skillsets (derived from the activities) that will ensure that the scrum master is successful in the everyday application of agile.

Also, our findings will help scrum masters better understand and delineate their role from other managerial roles such as those of the product owner or project manager.

7.2 Implications for research

Our grounded theory gives researchers the foundation to build upon and generate a deeper understanding of the contexts in which the scrum master functions in agile projects. A good direction for future research is to conduct mixed methods studies which investigate the effectiveness of the scrum master in agile projects. A good venue for the research setting will be implementations of large-scale agile. We are confident that some of our findings will be applicable to large-scale agile, while focused research will uncover many unseen aspects of the role. Additionally, our questionnaire is open to modification and can be run with larger sample sizes.

8 Limitations

As our GT study utilized qualitative and quantitative strands which had limited mixing, we have presented the limitations separately in the following paragraphs.

A Grounded Theory study generates a "*mid-ranged*" theory which is limited to the contexts studied and remains open to modification based on new data to suit new contexts (Glaser 1992). The participants of our research hailed from a wide range of project sectors including telecommunications, banking, to government, tourism and retail, and performed in various roles such as developers, project managers, scrum masters, and test engineers. Our theory is open to modification and extension based on future research work in different contexts.

The key limitation of the quantitative strand is the relatively small sample size. Additionally, while the quantitative study looked at the level of involvement and possible association of the scrum master with agile practices, it did not measure the effectiveness of the scrum master in those activities. We believe that the topic of scrum master effectiveness is ripe for future research. One other limitation is that there could be factors other than the presence of the scrum master which could lead to certain practices exhibiting an increase in their usage by the team. Some of the other limitations viz. sampling bias and response bias have been discussed previously in section 6.2.

9 Conclusion

In this paper, we have presented a Grounded Theory study with a mixed methods approach to study the role of the scrum master in agile projects. One of the key

highlights of this study is the use of qualitative and quantiatiave data to study an interesting aspect of ASD. We present a grounded theory of the role of the scrum master which explains that scrum masters perform six everyday activities which are:

- Facilitating: This activity involves the scrum master clearing obstacles and issues, facilitating the project teams functioning, and easing the team's transition to scrum.
- Mentoring: Mentoring involves the scrum master empowering the team on the path of self-organization, educating the team and stakeholders in agile practices, and ensuring team adherence to agile practices.
- Negotiating: This involves the scrum master aligning customer expectations with the team's actual capacity to deliver working software.
- Process adapting: This involves tailoring processes to suit project context by customizing agile practices and by integrating traditional project management practices with agile practices.
- Coordinating: The key coordinating activities carried out by the scrum master are involving specialists in the project and coordinating collaboration between teams from different areas of the organization.
- Protecting: The protecting activity involves shielding the team from external interference and pushing back on scope creep.

From our quantitative analysis, it appears that there is a positive association between certain agile practices (definition of done, iteration (sprint) planning, project velocity measurement, and short iterations (sprints)) and the presence of the scrum master. These practices exhibited a notable increase in their frequency of use in the presence of a scrum master.

Understanding the role of the scrum master in agile projects will help practitioners better manage expectations of this role. The key factor which will ensure their longevity in organizations transitioning to agile will be their adaptability to rapidly changing situations. Scrum masters need to be prepared to perform any of the six identified activities in agile projects.

Acknowledgments We would like to thank the participants of this research and the University of Auckland Ethics Committee.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Part A- Interview Questions

During the interview, research participants were requested to use the same project which was used to fill the online pre-interview questionnaire. This list is purely indicative and questions were modified or dropped depending on the flow of the conversation in the interviews.

1. Please tell me briefly about your professional background and how you were introduced to agile and scrum?
2. Could you give a brief background of the project/product and describe your role on the project/product?
3. Was the team self-organizing? What steps were taken to encourage the team to be self-organizing?
4. Was there any conflict in the team and how was it resolved? Please give an example.
5. Were there any obstacles in the way of team functioning/performance? If yes, how were they resolved?
6. Who was responsible for negotiating with stakeholders/vendors/customer representatives? How was the negotiation process done?
7. Were there any distractions to the teams functioning? How were distractions dealt with? Please give an example.
8. Were some of scrum practices adapted or changed to fit your project context? E.g. Using Kanban methods in conjunction with scrum.
9. Were the stakeholders and team well versed in scrum practices?
10. Could you give example of a time when you provided guidance to the team and ensured adherence to agile/scrum practices?
11. Were there any occasions when you had to either make the team understand the strategic business context or make stakeholders understand technical vocabulary?
12. Is there anything you would like to add?

Part B- Supplementary Tables

Table 4 Results of the Pearson Chi-Square analysis between the presence of the scrum master in the project and extent of involvement of the scrum master in agile practices

Agile Activity	Pearson Chi-Square Significance 2 sided (*p*)	Pearson Chi-Square (x^2)	Cramers V	Phi	Valid Cases
Daily Scrum	.000	21.828	.790	.790	35
Sprint Review Meeting	.005	13.047	.611	.611	35
Sprint Planning	.004	15.242	.660	.660	35
Definition of Done	.001	18.065	.718	.718	35
Project Velocity	.004	15.242	.660	.660	35
Sprint Retrospective	.002	15.242	.660	.660	35
Estimation	.000	27.097	.880	.880	35
Product Backlog	.000	27.097	.880	.880	35
Sprint Backlog	.001	18.065	.718	.718	35
Product Backlog Refinement	.000	21.828	.790	.790	35

Table 5 Results of the Pearson Chi-Square analysis between the presence of the scrum master in the project and the fequency with which an agile practice is carried out by the team

Agile Activity	Pearson Chi-Square Significance 2 sided (p)	Pearson Chi-Square (x^2)	Cramers V	Phi	Valid Cases
Definition of Done	.025	11.170	.557	.557	36
Iterations/Sprint Planning	.045	9.744	.455	.455	47
Project Velocity	.028	10.889	.481	.481	47
Short Iterations/Sprints	.073	8.565	.427	.427	47
Team Based Estimation	.319	4.708	.316	.316	47
Release Planning	.148	6.784	.380	.380	47
Agile Games	.671	2.354	.256	.256	47
User Stories	.575	2.901	.248	.248	47
Sprint Backlog	.619	2.645	.237	.237	47
Sprint Retrospective	.814	1.569	.183	.183	47
Sprint Review Meeting	.913	.979	.144	.144	47
Information Visualization	.390	4.117	.296	.296	47
Kanban Board	.382	4.182	.298	.298	47
Daily Scrums/Daily Standup	.336	3.381	.268	.268	47
Product Backlog (maintenance)	.089	6.507	.372	.372	47
Scrum of scrums meeting	.265	5.227	.333	.333	47

Table 6 This data is based on participants who spoke about the role of a scrum master on their team. For example, a participant who is a developer talking about a scrum master on the project

Scrum Master Code	Referred by Participant	Sector of Project
SM1	P01	Banking/ecommerce
SM2	P06	Banking
SM3	P10	Telecommunications
SM4	P15	Finance
SM5	P20	Telecommunications
SM6	P24	Software
SM7	P27	Human Resources
SM8	P32	Healthcare
SM9	P34	Healthcare

Table 7 Calculation of the Percentage of Scrum Masters performing an everyday activity: This data is aggregate of participant data (17 scrum masters) and referred data (9 scrum masters)

Everyday Activity	Participant Data (PD) (Out of 17)	Referred Data (RD) (Out of 9)	Total (PD + RD) (Out of 26)	Percentage Calculation ((PD + RD) ÷ 26) × 100
Facilitating	13	5	18	69%
Mentoring	16	2	18	69%
Process Adapting	7	1	8	31%
Protecting	6	1	7	27%
Negotiating	7	0	7	27%
Coordinating	2	1	3	12%

References

Anderson DJ, Carmichael A (2016) Essential Kanban Condensed. Blue Hole Press

Bass JM (2013) Agile Method Tailoring in Distributed Enterprises: Product Owner Teams. In: IEEE 8th Int. Conf. Global Software Eng, pp 154–163. https://doi.org/10.1109/ICGSE.2013.27

Beck K, Andres C (2005) Extreme programming explained: embrace change. Addison-Wesley, Boston

Birks M, Chapman Y, Francis K (2008) Memoing in qualitative research: probing data and processes. J Res Nurs 13:68–75. https://doi.org/10.1177/1744987107081254

Brewer J, Hunter A (1989) Multimethod research: A synthesis of styles, Sage library of social research (Vol. 175). Sage Publications

Cockburn A, Highsmith J (2001) Agile software development, the people factor. IEEE Computer 34:131–133. https://doi.org/10.1109/2.963450

Chow T, Cao DB (2008) A survey study of critical success factors in agile software projects. J Syst Software 81: 961–971. https://doi.org/10.1016/j.jss.2007.08.020

Chagas L, de Carvalho D, Lima A, Reis C (2014) Systematic literature review on the characteristics of agile project management in the context of maturity models. In: Mitasiunas A., Rout T., O'Connor R.V., Dorling A. (eds) Software Process Improvement and Capability Determination. SPICE 2014. Communications in Computer and Information Science (Vol. 477). Springer, Cham. https://doi.org/10.1007/978-3-319-13036-1_16

Charmaz K (2006) Constructing grounded theory- a practical guide through qualitative analysis. Sage Publications Ltd, Thousand Oaks

Creswell JW (2012) Qualitative inquiry and research design: choosing among five approaches. Sage Publications Ltd, Los Angeles

Creswell JW, Plano Clark VL (2011) Designing and conducting mixed methods research. Sage Publications Ltd, Los Angeles

Deemer GBP, Larman C,Vodde B (2012). The Scrum Primer. http://scrumprimer.org/. Accessed 20 November 2018

Denzin NK, Lincoln YS (2017) The SAGE handbook of qualitative research, 5th edn. SAGE Publications, Thousand Oaks

Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: a systematic review. Inform Software Technol 50:833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Easterbrook S, Singer J, Storey MA, Damian D (2008) Selecting empirical methods for software engineering research. In: In guide to advanced empirical software engineering. Springer, London, pp 285–311

Fowler M, Highsmith J (2001) The agile manifesto. Software Develop Magazine 9:29–30

Glaser BG (1992) Basics of grounded theory analysis. Sociology Press, Mill Valley

Glaser BG, Holton J (2004) Remodeling grounded theory. Forum Qual Soc Res 5:47–68. https://doi.org/10.17169/fqs-5.2.607

Glaser BG, Strauss AL (1967) The discovery of grounded theory: strategies for qualitative research. Aldine, New Brunswick

Glaser BG (1978) Theoretical sensitivity: advances in the methodology of grounded theory. Sociology Press, Mill Valley

Hoda R, Noble J (2017) Becoming agile: a grounded theory of agile transitions in practice. In: Proc. of the 39th Int. Conf. Software Eng (ICSE '17), pp 141–151. https://doi.org/10.1109/ICSE.2017.21

Hoda R, Noble J, Marshall S (2013) Self-organizing roles on agile software development teams. IEEE Trans Software Eng 39:422–444. https://doi.org/10.1109/TSE.2012.30

Hoda R, Noble J, Marshall S (2012) Developing a grounded theory to explain the practices of self-organizing agile teams. Empirical Software Eng 17:609–639. https://doi.org/10.1007/s10664-011-9161-0

Hoda R, Noble J, Marshall S (2011) The impact of inadequate customer collaboration on self-organizing agile teams. Inform Software Technol 53:521–534. https://doi.org/10.1016/j.infsof.2010.10.009

Howe KR (1988) Against the quantitative-qualitative incompatibility thesis or dogmas die hard. Educ Res 17:10–16. https://doi.org/10.3102/0013189X017008010

James W (1975) Pragmatism. Harvard University Press, Cambridge, MA

Johnson RB, Onwuegbuzie AJ (2004) Mixed methods research: a research paradigm whose time has come. Educ Res 33:14–26. https://doi.org/10.3102/0013189X033007014

Martin A, Biddle R, Noble J (2010) An ideal customer: a grounded theory of requirements elicitation, communication and acceptance on agile projects. Agile software development: current research and future directions 111-141. In: Dingsøyr T, Dybå T, Moe NB (eds) . Springer, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12575-1_6

Noll J, Razzak MA, Bass JM, Beecham S (2017) A study of the scrum Master's role. In: Felderer M., Méndez Fernández D., Turhan B., Kalinowski M., Sarro F., Winkler D. (eds) Product-Focused Software Process

Improvement. PROFES 2017. Lecture Notes in Computer Science (Vol. 10611). Springer, Cham. https://doi.org/10.1007/978-3-319-69926-4_22
Palmer SR, Felsing M (2001) A practical guide to feature-driven development. Prentice-Hall, Upper Saddle River
Schwaber K, Beedle M (2002) Agile software development with scrum. Prentice-Hall, Upper Saddle River
Shastri Y, Hoda R, Amor R (2020) Survey of roles in agile project management and their involvement in agile practices, Harvard Dataverse, V1. https://doi.org/10.7910/DVN/QEASRH, Harvard Dataverse, V1
Shastri Y, Hoda R, Amor R (2017) Understanding the roles of the manager in agile Project Management. In: Proceedings of the 10th Innovations in Software Engineering Conference (ISEC '17), Association for Computing Machinery, New York, NY, USA, pp 45–55. https://doi.org/10.1145/3021460.3021465
Siddique L, Hussein BA (2016) Grounded theory study of conflicts in Norwegian agile software projects: the project managers' perspective. J Eng Proj Prod Manage 6:120–135
Sutherland J, Schwaber K (2017) The scrum guide: the definitive guide to scrum. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100. Accessed 22 November 2018
Stapleton J (2003) DSDM: business focused development. Addison-Wesley, Boston
Stol KJ, Ralph P, Fitzgerald B (2016) Grounded theory in software engineering research: a critical review and guidelines. In: ICSE '16: proceedings of the 38th ACM/IEEE international conference on software engineering, ACM, pp 120–131. https://doi.org/10.1145/2884781.2884833
Stray V, Sjøberg DIK, Dybå T (2016) The daily stand-up meeting: a grounded theory study. J. Syst Software 114:101–124. https://doi.org/10.1016/j.jss.2016.01.004
Tashakkori A, Teddlie C (1998) Mixed methodology: combining qualitative and quantitative approaches. Applied Social Research Methods. Sage Publications Inc.
Tavakol M, Dennick R (2011) Making sense of Cronbach's alpha. Int J Med Educ 2:53–55. https://doi.org/10.5116/ijme.4dfb.8dfd
VersionOne Inc. (2018). The 12th annual state of agile report. VersionOne Incorporated. https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report. Accessed 20 November 2018
VersionOne Inc. (2017). The 11th annual state of agile report. VersionOne Incorporated. https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report. Accessed 19 June 2018
Waterman M, Noble J, Allan G (2015) How much up-front?: a grounded theory of agile architecture. In: ICSE '15: proceedings of the 37th ACM/IEEE international conference on software engineering, ACM, pp 347–357. https://doi.org/10.1109/ICSE.2015.54

Affiliations

Yogeshwar Shastri[1,2] · Rashina Hoda[3] · Robert Amor[4]

[1] School of Computing, Eastern Institute of Technology, Napier, New Zealand

[2] Department of Electrical Computer, and Software Engineering, University of Auckland, Auckland, New Zealand

[3] Academic Workforce, Faculty of Information Technology, Monash University, Melbourne, Australia

[4] School of Computer Science, University of Auckland, Auckland, New Zealand

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com