The Journal of Systems & Software 173 (2021) 110871

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

The role of the project manager in agile software development projects

Yogeshwar Shastri [a,*], Rashina Hoda [b], Robert Amor [c]

[a] School of Computing, Eastern Institute of Technology, New Zealand
[b] Faculty of Information Technology, Monash University, Australia
[c] School of Computer Science, University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 30 August 2019
Received in revised form 17 November 2020
Accepted 19 November 2020
Available online 25 November 2020

Keywords:
Project manager
Agile software development
Agile project management
Scrum

ABSTRACT

Agile teams are not meant to have project managers. Instead, agile methods such as Scrum and XP define roles such as product owner, scrum master, and coach. Studies have uncovered the existence of the project manager in agile projects, pointing to disconnect between theory and practice. To address this gap, a Grounded Theory study with a mixed methods approach was conducted using multiple sources of data including over 45 h of interviews with 39 software practitioners and quantitative data from 57 questionnaire respondents. We present and describe the project manager's role in agile projects in terms of (a) everyday activities: facilitating, mentoring, negotiating, coordinating, and protecting, performed by the project manager using; (b) three management approaches: hard, moderate, and soft; (c) four traditional project management activities continued to be performed by them, including: tracking project progress, reporting on project status, budgeting and forecasting, and managing personnel; and (d) the influence of the presence of the project manager on the frequency with which agile activities are carried out by the teams. Our study highlights the continued presence of the role of the project manager in agile software projects as a part of the transition from traditional to agile ways of working.

1. Introduction

The project manager is considered pivotal in traditional software development projects involving multiple aspects of managing teams such as leadership, team building, motivation, communication, influencing, decision making, planning, and coaching (Pettersen, 1991). In traditional software development methods, such as the waterfall model (Royce, 1987; Benington, 1987), the project manager sits within a well-defined hierarchy in the project team (Pettersen, 1991), playing a role very similar to the one outlined in project management literature (Project Management Institute, 2018).

With its emergence in the late 1990s, Agile Software Development (ASD) introduced self-organizing teams to software engineering (Fowler and Highsmith, 1991). Self-organizing teams have been characterized as teams displaying significant autonomy in taking decisions (Cockburn and Highsmith, 2001), managing workloads and allocating work amongst themselves (Chow and Cao, 2008; Chagas et al., 2014). 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 (Schwaber and Beedle, 2002; Deemer et al., 2012; Sutherl and Schwaber, 2017). XP introduced the role of the *coach* – tasked with process and team guidance (Beck and Andres, 2005). In fact, the job title and role of the project manager simply does not exist in popular ASD methods such as Scrum and XP (Schwaber and Beedle, 2002; Fowler and Highsmith, 1991; Deemer et al., 2012; Sutherl and Schwaber, 2017).

However, there is growing evidence that the job title of the project manager is still in existence in organizations practicing ASD (Shastri et al., 2016; Siddique and Hussein, 2016; Digital.ai Software Inc., 2020; VersionOne Inc., 2018), highlighting a gap between what agile theory recommends and what is implemented in practice. Previous studies have investigated the project manager's preference for agile methods (Bishop et al., 2018), their views on conflicts in the team (Siddique and Hussein, 2016), and the underlying tensions emerging from conflicting expectations about the project manager's role in ASD projects (Taylor, 2016). Additionally, studies on self-organizing teams have hinted that a certain proportion of the erstwhile manager's responsibilities are carried out by the new roles such as scrum master, and product owner (Hoda et al., 2013; Drury-Grogan and O'dwyer, 2013; Bass, 2013). To date there has not been a dedicated study which comprehensively looks at the role of the project manager in agile projects. The ambiguity surrounding the project manager's role in ASD is clearly expressed by a leading practitioner compendium on agile, the Agile Practice Guide, jointly published by the Project

* Corresponding author.
E-mail address: ysha962@aucklanduni.ac.nz (Y. Shastri).

https://doi.org/10.1016/j.jss.2020.110871
0164-1212/

Management Institute (PMI) and Agile Alliance, when it says that "*the role of the project manager in an agile project is somewhat of an unknown, because many agile frameworks and approaches do not address the role of the project manager*" (Project Management Institute, 2017b).

The ambiguity surrounding the project manager role can be broken down into the following questions: Do agile projects have identified project managers (alongside agile roles)? What does the project manager do on a regular basis in agile projects? What management styles or approaches does the project manager adopt when carrying out their role? What influence, if any, does the project manager exert on the agile practices carried out by the team? Are there any traditional project manager activities still carried out in agile projects? We address and attempt to answer these and a critical overarching question: "*What is the role of the project manager in agile projects?*"

An earlier publication of the preliminary results (Shastri et al., 2017), based on 21 interviews, had presented the roles played by managers in agile teams: the mentor, coordinator, negotiator, and process adapter. Our current paper is focused on the project manager role in depth. In this paper, we present a theory of the role of the project manager in agile projects resulting from a Grounded Theory study with a mixed method approach based on multiple and varied data including over 45 h of semi-structured interviews with 39 software practitioners and quantitative data from 57 questionnaire respondents. This theory explains how the project manager performs five different activities – *facilitating, mentoring, negotiating, protecting*, and *coordinating* – on an everyday basis, by using three different management approaches – *hard, moderate,* and *soft* – and the possible influence of the project manager on the frequency with which agile practices are carried out by teams. The project manager continues performing some of the traditional project management duties such as tracking project progress, reporting on project status, budgeting and forecasting, and managing personnel even in ASD projects. As far as we are aware this is the first Grounded Theory study to present a comprehensive picture of the project manager in agile projects.

In the following sections we present the literature review, the research methodology, the key findings, discussion and implications of the findings, a comparison to relevant contemporary research, limitations, and the conclusion.

2. Literature review

The following section presents a literature review which traces the evolution of traditional software development, the role of the project manager, the introduction of agile methods in the software industry, and the status of the project manager in agile software development projects.

2.1. Traditional software development

Waterfall, a popular traditional software development model, is a specification-driven approach characterized by extensive planning, upfront requirements gathering, detailed documentation, and a focus on immaculate execution of the process (Royce, 1987; Benington, 1987). Waterfall incorporates many aspects of traditional project management such as the sequential arrangement of different development steps, emphasis on extensive upfront planning and a phase-based approach (Rowen, 1990).

The main criticisms of the waterfall approach is linked to its poor flexibility to changes in the project environment (Boehm, 2006; Robey et al., 2001). The fact that the sequential waterfall model did not fit well into a fluctuating software development environment was recognized early on, resulting in an alternate spiral model proposed by Boehm (Boehm, 1988). This model focused on addressing risks and involving the customer in the development process. To address the human aspects of software development, Boehm and Ross also proposed a people centric theory of software project management (known as Theory-W) which proposed that the project would be successful only if the project manager was responsible for generating a win–win situation for users, customers, team members, and concerned stakeholders (Boehm and Ross, 1989).

2.2. The traditional project manager

The earliest conceptualization of the project manager was given by Gaddis in the 1950's as someone who: "*manages a team of professionals, whose job is finite in duration, who recruits the project team, conducts project planning and is able to sell the project to stakeholders*" (Gaddis, 1959). More recently, the 2018 edition of the Project Management Body of Knowledge (PMBOK), defines the project manager "*as the person assigned by the performing organization to lead the team that is responsible for achieving the project objectives*" (Project Managerment Institute, 2018). A project manager is envisioned as a connection between the project team and the stakeholders. In recent years, there has been a surge in the demand for project managers across different industry sectors (Bredillet et al., 2015). This is reflected in the burgeoning membership of professional bodies such as PMI and an increase in their members who hold project manager certifications (Project Management Institute, 2017a).

The PMBOK emphasizes that the project manager should possess some key competencies and a range of interpersonal skills (Project Managerment Institute, 2018). A project manager is expected to be proficient in leadership, team building, motivation, communication, influencing, decision-making, political and cultural awareness, negotiation, trust building, conflict management, planning, effective supervision, budgeting, and coaching (Project Managerment Institute, 2018; Pettersen, 1991). Research studies have proposed individual competencies such as leadership and communication skills (Stevenson and Starkweather, 2010), stress management (Müller and Turner, 2007), courage, and temperance (Bredillet et al., 2015).

This model of the project manager role is not universally accepted (Bredillet et al., 2015; Morris et al., 2006; Pinto and Winch, 2016; Cicmil and Hodgson, 2006; Loufrani-Fedida and Missonier, 2015; Hodgson and Paton, 2016). Based on their qualitative study of large project-based organizations, Loufrani-Fedida and Missonier (Loufrani-Fedida and Missonier, 2015) suggest it is unrealistic to expect the project manager to display all the competences ascribed in the literature. They recommended that some of the individual competencies assumed for an ideal project manager should be shouldered by functional managers (Loufrani-Fedida and Missonier, 2015; Hodgson and Paton, 2016), who have operational responsibilities for a department within the organization (Project Managerment Institute, 2018), and are typically not associated with project teams.

Other criticism of the PMBOK's view of the project manager's role include Cicmil et al. (Cicmil et al., 2006), objecting to its portrayal as a "*skilled technician*" i.e. a person whose role primarily centers on controlling time, cost and the scope of the project, or as the "*implementer*" of a large body of standards, practices and techniques prescribed by professional project management literature (Cicmil and Hodgson, 2006), echoing the criticisms of Morris (Morris, 1997) and Morris et al. (Morris et al., 2006). Overall, the current definition of the traditional project manager in theory is seen to be at odds with practice even in traditional software development projects. When looking to the current understanding of the project manager's role in agile software projects, things are more ambiguous.

2.3. 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), and feature-driven development (FDD) (Palmer and Felsing, 2001). ASD has a clear emphasis on people and on rapid response to change (Cockburn and Highsmith, 2001; Fowler and Highsmith, 1991). In the last decades, the adoption of ASD has been extremely rapid in the software industry worldwide (Dybå and Dingsøyr, 2008; Larman and Basili, 2003; VersionOne Inc., 2018).

In the recurring industry-based "*state of agile*" survey (Digital.ai Software Inc., 2020; VersionOne Inc., 2018, 2019) Scrum was identified as the most commonly used agile method. In the most recent iteration of the survey, 58% of respondents indicated that their project used Scrum as a standalone agile method and over 76% when used in combination with other methods (Digital.ai Software Inc., 2020). The next popular standalone method was Kanban, with 7% of respondents while XP was reported as being used by a minority (<1%) of the respondents. While Scrum focuses on the project management aspects of agile such as estimation and planning, XP focuses on development practices such as test-driven development and pair programming (Pikkarainen et al., 2008).

In recent years there has been a flowering of research on "*large scale agile*". This is in part a reflection of the growing need in the software industry on clear guidelines regarding implementing agile in large projects with multiple teams. Dikert et al. in their systematic literature review (SLR) have defined large scale agile as, "*software development organizations with 50 or more people or at least six teams* (Dikert et al., 2016)". In their SLR, Dikert et al. (Dikert et al., 2016) discovered that Scrum was the most popular method used in organizations which were undergoing large scale agile transformation.

2.4. The project manager in Agile software development

In ASD methods, such as Scrum, and XP, the job title and role of the project manager simply does not exist (Beck and Andres, 2005; Deemer et al., 2012; Sutherl and Schwaber, 2017). Scrum introduced two new roles, namely that of the product owner and the scrum master (Schwaber and Beedle, 2002). The product owner is the customer representative, whereas the scrum master is primarily the internal facilitator (Highsmith, 2004). XP introduced different roles such as the coach, consultant, tracker, programmer, customer, tester, and the big boss (Beck and Andres, 2005). Of these, the coach is responsible for the process, guiding the team and learning from other XP teams (Beck and Andres, 2005). It seems that the scrum master, product owner and the XP coach roles share some characteristics and responsibilities of the traditional project manager (Project Managerment Institute, 2018; Beck and Andres, 2005). Critically, the job title and role of the project manager does not exist in agile methods.

Industry surveys (VersionOne Inc., 2018) and recent research (Shastri et al., 2016, 2017) point to the continued existence of the job title of the project manager in agile projects. The recent industry surveys have indicated that nearly 14% to 23% (VersionOne Inc., 2018) of their respondents were project or program managers while a research study shows nearly 67% of the respondents' projects had a project manager present (Shastri et al., 2016).

Recognizing the reality of the project manager's co-existence with self-organizing agile teams, the Agile Practice Guide, a joint publication of PMI and the Agile Alliance, has re-fashioned the project manager's role as a "*servant leader*" (Project Management Institute, 2017b). As a servant leader, the project manager's focus is on coaching, collaboration, and stakeholder management.

While considerable research has been done into the various aspects of agile teams (Siddique and Hussein, 2016; Bishop et al., 2018; Taylor, 2016; Nkukwana and Terblanche, 2017), research on the project manager's role in agile is still scarce.

Taylor's ethnographic study (Taylor, 2016) on the experience of project managers in ASD identified that underlying "*tensions exist as PMs may still be held responsible for project outcomes, yet they are expected to delegate decision making to the team*" (Taylor, 2016). This tension is due to differing expectations of management and the team (Nkukwana and Terblanche, 2017). While senior management expects the project manager to take responsibility for project delivery and adopt a controlling approach if necessary, on the other hand, the teams expect a light touch "*servant leader*" type approach (Nkukwana and Terblanche, 2017).

A recent study by Siddique and Hussein (2016) addressed conflict within agile teams from a project manager's perspective, attributing it to a lack of experience of the project manager, lack of customer involvement, budgetary issues, and ego conflicts within teams. The consequences of such conflicts can be a drop in productivity, lowering of motivation, and poor decision making.

Bishop et al. (2018) study on the reasons behind project manager preferences for agile methods revealed that the project managers had a pragmatic approach when selecting an agile method. Project managers appreciated the practical benefits of agile adoption such as adaptability, increased efficiency and faster delivery of features but also identified negative factors, such as a desire for fixed outcomes.

Overall, we find that the role of the project manager in traditional software development as defined by traditional standards and the body of knowledge (Project Managerment Institute, 2018) is not universally accepted (Morris et al., 2006; Cicmil and Hodgson, 2006; Cicmil et al., 2006; Morris, 1997; Loufrani-Fedida and Missonier, 2015). Furthermore, there is ample ambiguity around the role of the project manager in agile software development as proposed in theory (Highsmith, 2004; Schwaber and Beedle, 2002; Beck and Andres, 2005; Sutherl and Schwaber, 2017) and identified in practice (Shastri et al., 2016; Digital.ai Software Inc., 2020; VersionOne Inc., 2018; Shastri et al., 2017). This leaves a critical gap in research, to define and describe the role of the project manager in agile software development.

2.5. Recent trends in Agile research

Implementing agile on a large scale has been the topic of extensive research in the last few years (Scheerer et al., 2014; Smite et al., 2019; Paasivaara and Lassenius, 2019; Gustavsson, 2019; Conboy and Carroll, 2019; Moe et al., 2019; Dingsøyr and Moe, 2013; Bass and Haxby, 2019; Dikert et al., 2016). As early as 2013, Dingsøyr and Moe (2013) had identified large scale agile transformation, inter-team coordination, and scaling of agile practices as some of the key topics for future research in large-scale agile. This call to research has been answered by several researchers in the last few years and one of the persistent focal points for research has been inter-team coordination in large scale agile projects. Researchers have studied: coordination in large scale agile using the multi-team systems perspective (Scheerer et al., 2014); the manner in which agile practices are tailored for the large scale implementation (Smite et al., 2019; Gustavsson, 2019; Dikert et al., 2016); adoption and sustenance of agile frameworks (Conboy and Carroll, 2019); roles of the product owner (Bass and Haxby, 2019); and team autonomy (Moe et al., 2019).

The coordination aspect was also studied by Gustavsson (2019) who looked at the manner in which coordination routines such

as the scrum of scrums are enacted and the reasons for which they are enacted in a particular manner. Paasivaara and Lassenius (Dikert et al., 2016) investigated the ways in which large firms such as Ericsson dealt with coordination challenges over large multi-team projects. A similar approach was reported at Spotify by Smite et al. (2019) who conducted a study of Spotify's culture of guilds to understand how coordination was carried out across multiple teams. Conboy and Carroll (2019), based on their long running research on adoption and sustenance of agile frameworks, identified nine challenges associated with adopting large scale agile frameworks. Researchers have also studied the role of the product owner in large scale agile (Bass and Haxby, 2019). Team autonomy is another area of large scale agile that has been touched upon by researchers (Moe et al., 2019).

Some of the works cited above are particularly relevant to our findings and are discussed in detail in Section 6.

3. Research methodology

The starting point of our Grounded Theory (GT) study was to investigate the role of the project manager in agile projects. To this end we conducted a quantitative study to see if the job title of the project manager still existed. The results indicated the job title of project manager is still widely used (Shastri et al., 2016).

This led to the next step where we designed the qualitative questionnaire to investigate the role of the project manager. Before piloting the qualitative questionnaire to the first set of 10 participants, it was felt necessary to gather information about the participant's project and professional background prior to the face-to-face interview. Thus, a quantitative questionnaire (the *pre-interview questionnaire*) was created which was sent to the participants a few days before the interview. The complete pre-interview questionnaire has been uploaded to Dataverse and the link has been provided in the references (Shastri and Hoda, 2020). The pre-interview questionnaire was extremely useful as we could use the information to tailor the interview questions appropriate to the participant's context. Additionally, as the qualitative interviews had fixed duration (usually under an hour), we could focus in the interview on asking questions relevant to the participants project, rather than using the precious interview time to gather demographic information.

After the first 10 interviews the pool of qualitative and quantitative questions was broadened to include questions regarding agile practices and involvement of the project managers in selected practices.

3.1. Grounded theory as the research methodology

GT is defined as, "*a general methodology of analysis linked with data collection that uses a systematically applied set of methods to generate an inductive theory about a substantive area*" (Glaser, 1992). GT owes its genesis to the work of two sociologists, Barney Glaser and Anslem Strauss, who developed it in the mid-1960s (Glaser and Holton, 2004; Charmaz, 2006).

The growing attractiveness of GT to software engineering researchers (Stol et al., 2016) lies in two factors. Firstly, GT is well suited to generate a theory for a relatively new discipline such as software engineering. Secondly, GT is an inductive process rather than a deductive one. The inductive process plays a critical role in uncovering the underlying concerns of software engineering practitioners. Stol et al. (2016) review of grounded theory application in computer science and software engineering found 98 journal articles published between 1995 and 2015 that used GT or used selected techniques from GT, including a large number of studies which have used GT to study the human and social aspects of software engineering (Hoda et al., 2013; Siddique and Hussein, 2016; Hoda et al., 2012, 2011; Stray et al., 2016; Dybå and Dingsøyr, 2008; Hoda and Noble, 2017; Waterman et al., 2015). An additional benefit is GT's flexibility to accommodate both qualitative and quantitative data, although this has not been exploited fully thus far in software engineering research.

For this study, we have used the Glaser's GT method with a mixed method approach using both (qualitative and quantitative) data collected via interviews and questionnaires, as explained below. While there is some level of mixing of qualitative and quantitative data in our study, there is also a degree of separation between both the strands. A GT study by its very emergent nature makes pre-determined and systematic mixing of qualitative and quantitative data challenging (Charmaz, 2006).

3.2. Quantitative data collection and analysis

The quantitative data collection was conducted in two phases: The first phase was designed to investigate whether the job title of the project manager still existed in agile projects. This survey was posted on LinkedIn in several groups which had participation by software engineering practitioners and the survey had 94 respondents. The analysis of this survey demonstrated that the job title of the project manager was still extant in a majority of the respondent's projects (Shastri et al., 2016). The second phase of quantitative data collection involved creating and administering a "*pre-interview questionnaire*" to participants of the face-to-face interview. Participants for the interview process were recruited through a call for participation posted on LinkedIn, and social media accounts of agile groups such as Agile Auckland.

Additionally on LinkedIn, in order to shortlist participants for the interviews, BOOLEAN search operators were used to identify software practitioners who were working in an agile environment. For example, search terms such as "*Agile & New Zealand*" were used to generate lists of prospective participants. Relevant participants were identified based on these searches and a connection request was sent to them. Participants were made aware of the focus of the research, that is, project managers, and, where requested, additional information pertaining to the research project was sent. This was a time-consuming process and resulted in approaching between 300–400 practitioners. Out of these around 57 agile practitioners responded to our request and agreed to participate in the research process. Once consent was obtained from the participants, they filled in the pre-interview questionnaire. However, only 39 participants proceeded to the subsequent interview process which formed the basis of the qualitative data.

The pre-interview questionnaire created using Google forms was used as it was felt essential to build a picture of the participants' backgrounds and demographics prior to the interviews. In this way, the interview could focus more on the open-ended questions. The link was emailed to participants at least one week prior to the interview. The response options to the questions included a mixture of multiple choice questions such as "*please select the relevant agile methodologies you have used*", closed ended questions such as "*is there a project manager on your project*?" and Likert scale type questions such as "*please rate the frequency of using the practices listed below*".

The quantitative data obtained from 57 pre-interview questionnaire responses was mostly Likert scale data. This was first cleaned by applying consistent labeling in the manual entry responses, and the responses were compared to the interview transcripts to cross-check for any errors. Once the survey data was validated and its integrity verified by running Cronbach's Alpha test, it was transferred into IBM SPSS v.24. 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 0.939 for the section of our questionnaire which asked participants to grade the involvement of different roles in agile practices. While for the section which asked respondents to rate how frequently agile practices were used in their projects, the Cronbach 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.

For example, a survey section asked 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., 2018). The full list of practices can be seen in Table 2 later. 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 project manager (PM) 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 PM is present) - (HFU as % of cases in which PM is absent)

A positive PDL value indicates that the frequency of a practice goes up when the project manager role is present. Similarly, a negative PDL value indicates that the frequency of a practice goes down when the project manager is present. The results of the cross-tabulation analysis are discussed in the findings section.

3.3. Qualitative data collection and analysis

The qualitative data was collected from 39 agile practitioners who agreed to an interview after filling in the pre-interview questionnaire (summarized in Section 3.2). Thirty-two of these participants were from New Zealand, five from India and one participant each from Australia and the USA. Most of the interviews were on average an hour long and were conducted face to face, except for four, which were conducted over Skype as the participants were unavailable in person. All of the recorded interviews were transcribed using a professional transcriptionist who was approved by the University. Prior to outsourcing of the transcription, the transcriptionist signed a confidentiality agreement to protect participant data.

The interview included questions such as:

Please tell me briefly about your professional background and your current role in this organization;

How would you describe the Project Managers role on the project?

Who was responsible for negotiating with stakeholders and how was negotiation done?

Were there any obstacles in the way of team functioning/ performance? If yes, how were they resolved?

What are the major challenges you have faced while working in the agile project?

How did you overcome those challenges?

Table 1 shows a breakdown of the participant demographics and project information. To ensure confidentiality, the participants have been assigned code numbers beginning with a "P" i.e. P01, P02, etc. During the interview process, a number of participants chose to speak about more than one project in which they had worked in different capacities. For example, in one project they might have been a team member, while in another they could have been a project manager. Thus, they would be providing information from different perspectives and talking about different roles. In such a scenario to avoid mixing up the data, we adopted a simple scheme to indicate the job title and the project sector of the participant across more than one project. For example, in Table 1, for participant P13, in the "*Job Title*" column, the job title "*Developer*^{1}*-Agile Coach*^{2}" indicates that the participant was referring to a developer role in an earlier project and an agile coach role in a later project. Similarly, moving to the column titled "*Project Sector*", the label "*Accounting*$^{1-2}$" with the numbers in superscripts indicates that the project sector was the same for both the projects. In addition to the data obtained from participants who were project managers, others such as testers and developers provided substantial information regarding the role of project managers in their projects. These project managers referred to by participants are identified by an alphanumeric code (for example, PM1, PM2 etc.).

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 primary researcher in consultation with the co-authors. Fig. 1 shows an overview of the GT analysis process and the emergence of the key concepts, categories, and the theory. As an example, we have given snippets of raw data followed by a couple of key codes and concepts which constituted two key categories, that is, the activities of facilitating and coordinating. In the following paragraphs we explain some of the key steps of performing a Grounded Theory analysis.

Open coding: This is the process of coding the interview transcripts line-by-line. This is necessary to identify substantive codes emergent within the data (Glaser, 1992) . The first stage of analysis involved sifting through the raw data (interview transcripts) and extracting snippets of data from the transcript. Glaser and Holton have explained the process as, "*The process begins with line-by-line open coding of the data to identify substantive codes emergent within the data. The analyst begins by coding the data in every way possible—running the data open*" (Glaser and Holton, 2004). This data was then assigned a code, which is a phrase that summarized the data snippet in a short and clear description, usually between 2–5 words long. An example of data analysis from the raw data stage to the codes, concepts and category is presented below.

Raw data: "*The business analyst was extremely blunt when he was dealing with the product owner in terms of decisions that were being made and whether something was worth doing or not doing, which caused friction within the relationship. I would basically then facilitate a smoother transaction in terms of stepping in to make sure we continue to work together*". – P07, Project Manager, New Zealand.

Code: Resolving conflicts

Constant comparison method: This is a continuous and iterative process of generating codes by analyzing interviews and then comparing the codes to those generated within the interview and within other interviews in the dataset. This process results produces higher levels of abstraction such as concepts and categories (Glaser, 1992). The code "*resolving conflicts*" was found to share similarities with two other codes, namely, "*facilitating issue resolution*" and "*clearing obstacles*". These codes were grouped to a higher-level concept of "*facilitating minesweeping*" where minesweeping refers to identifying and removing obstacles for the team.

Concept: Facilitating minesweeping

The concepts were further grouped to the category of the "*Facilitating*" as an informal activity carried out by the project manager. The concepts which were grouped included: facilitating minesweeping, facilitating team functioning and facilitating processes.

Category: Facilitating

Table 1
Participant demographics.

Participant number	Job title	Experience in Agile	Team size	Country	Project sector
P01	Developer	2	11–15	USA	Banking
P02	Project manager	<1	11–15	New Zealand	Local government
P03	Project manager	4	6–10	New Zealand	Telecommunications
P04	Project manager	4	>25	New Zealand	Local government
P05	Programme manager	10	0–5	New Zealand	Insurance
P06	Software product manager	3	>25	New Zealand	Banking
P07	Project manager	5	6–10	New Zealand	Insurance
P08	Project manager	12	11–15	New Zealand	Telecommunications
P09	Senior project manager	5	16–20	India	Banking
P10	Product owner	3	6–10	New Zealand	Telecommunications
P11	Programme manager	10	16–20	New Zealand	Finance
P12	Scrum master	4	>25	New Zealand	Local government
P13	Developer[1]-Agile coach[2]	10	0–5	New Zealand	Accounting[1–2]
P14	Agile coach	10	6–10	New Zealand	Telecommunications
P15	Developer	4	0–5	New Zealand	Finance
P16	Scrum master	5	6–10	Australia	Utilities
P17	Project manager & (Scrum master)	3	21–25	New Zealand	Accounting
P18	Scrum master	1	6–10	New Zealand	Finance
P19	Scrum master	5	6–10	New Zealand	Education
P20	Technology consultant & (Product owner)	4	11–15	India	Telecommunications
P21	Scrum master & (Agile coach)	7	>25	New Zealand	Finance
P22	Developer	6	6–10	New Zealand	Finance
P23	Software engineer & (Scrum Master)	1	6–10	New Zealand	Taxation
P24	Product manager & (Product owner)	5	16–20	New Zealand	Software
P25	Quality analyst	9	6–10	New Zealand	Education
P26	Scrum master	6	16–20	New Zealand	Entertainment
P27	Senior director product management	12	6–10	New Zealand	Human resources
P28	Project manager[1]-Scrum master[2]	10	6–10	New Zealand	Retail[1&2]
P29	SAP delivery team manager & (Scrum Master)	8	21–25	New Zealand	Retail
P30	Solutions architect	1	16–20	New Zealand	Retail
P31	Programme manager[1]-Product owner[2]	5	6–10	New Zealand	Tourism[1]-Healthcare [2]
P32	Tribe lead	8	0–5	New Zealand	Healthcare
P33	Scrum master[1]-Software engineer[2]	3	6–10	New Zealand	Human resources[1]-Software [2]
P34	Software engineer	5	0–5	New Zealand	Healthcare
P35	Senior test engineer	9	16–20	New Zealand	Healthcare
P36	Team lead & (Scrum master)	5	6–10	New Zealand	Healthcare
P37	Project manager[1]-Scrum master[2]	2	6–10	India	Software[1]-Software[2]
P38	Scrum master	4	6–10	India	Security
P39	Scrum master[1]-Scrum master[2]	9	6–10	India	Finance[1]-Payment solutions[2]

(The numbers "*1*" & "*2*" in superscripts are used to differentiate the job title and project sector of the participants who spoke about their role in multiple projects).

Other categories which emerged in a similar way included mentoring, negotiating, coordinating, and protecting. Together they represent the '*everyday activities performed*', a key dimension of the role of the project manager on agile projects.

Through the analysis, differences in how project managers approached the same activities emerged. In particular, we found that project managers adopted from three management approaches when performing these everyday activities: *hard, moderate* and *soft*.

The *hard* aspect reflects the assertive posture adopted by a project manager in the project. This means a firm and assertive approach in the best interests of the team and the project given the context. This does not signify an uncompromising and controlling attitude. The moderate approach is a mixture of assertiveness and subtle persuasion. This can also be thought as the "middle path" approach. The facilitation activity abounds with examples of this approach.

An approach was classified as *soft* when the thrust was on subtle and gentle persuasion. This does not mean indecision or irresoluteness. The project manager only takes over the team controls in exceptional circumstances. The soft approach is also a vote of confidence in the team's capabilities.

Each of these approaches is discussed in the context of the everyday activities performed by the project manager in the findings section. Finally, project managers were also seen to continue performing some traditional project management activities, such as tracking project progress, reporting on project status etc.

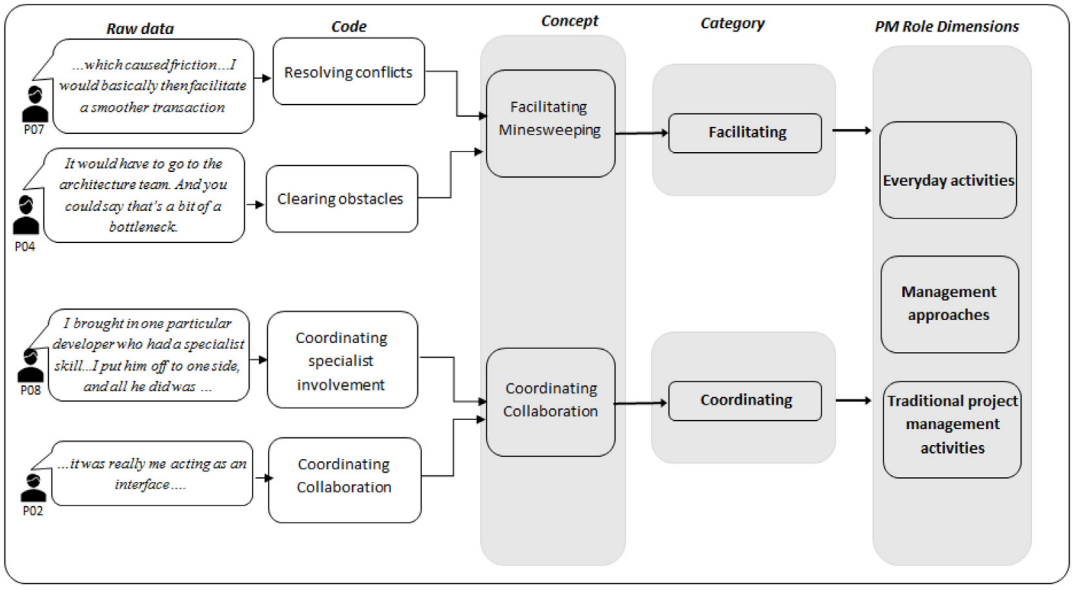

Fig. 1. Emergence of key concepts and categories leading to the formulation of the theory of the project manager in agile projects, using Grounded Theory data analysis.

Memoing: Memos are a concrete record of the researcher's decisions throughout the research process. Glaser has defined memos as, "*a theoretical note about the data and the conceptual connections between categories written down as they strike the researcher*" (Glaser, 1978). Memos by nature are free-flowing chunks of text which allow the researcher to informally capture a snapshot of the research in time.

Selective coding: In contrast to open coding, selective coding focuses only on the key categories and concepts which are related to the emerging theory. Once the underlying key categories had emerged we stopped open coding and began selective coding for the key categories. The emergence of the key categories and the overall theory of the project manager in agile projects is shown in Fig. 1 and the final theory with its different dimensions is shown in Fig. 2. Glaser and Holton have emphasized that, "*Selective coding means to cease open coding and to delimit coding to only those variables that relate to the core variable in sufficiently significant ways as to produce a parsimonious theory. Selective coding begins only after the analyst is sure that he/she has discovered the core variable*" (Glaser and Holton, 2004).

Theoretical coding: This is the process of using a theoretical structure to visualize the relationships between the categories. GT has several theoretical coding families which helps researchers to present their theory (Glaser, 1978). Theoretical coding has been explained as, "*theoretical codes conceptualize how the substantive codes may relate to each other as hypotheses to be integrated into the theory. Theoretical codes give integrative scope, broad pictures and a new perspective*" (Glaser and Holton, 2004). Thus, theoretical coding is the process of using a theoretical structure to visualize the relationships between the categories. In the GT milieu there are different types of theoretical coding families. Some of the theoretical coding families are: the Six C family (causes, contexts, contingencies, consequences, covariances, and conditions); the process family (stages, phases, phasing's, transitions, passages, careers, chains, sequences); the strategy family (strategies, tactics, techniques, mechanisms, management); and the dimensions family (social norms, social values, social beliefs) (Glaser, 1978). It is important that the theoretical coding model should fit the findings of analysis, rather than forcing the findings into a theoretical family (Glaser, 1978).

As the data findings emerged, initially the Six C model seemed like a good fit to visualize relationships. However, our findings did not map onto the Six C model, primarily because of the lack of '*covariance*' related findings, which is one of the six C's. We referred back to literature to look for a more appropriate model. Thus, further reflection led us to conclude that the theoretical coding family best suited to our findings was the dimension family (Glaser, 1992; Glaser and Holton, 2004) which enables the findings to be presented a facets or aspects of a phenomenon, in this case dimensions of the role of the project manager. Thus, the role of the project manager in agile projects is described in terms of:

(a) the everyday activities: *facilitating, mentoring, negotiating, coordinating*, and *protecting*, performed by the project manager using
(b) three management approaches: *hard, moderate*, and *soft*,
(c) four traditional project management activities: *tracking project progress, reporting on project status, budgeting, forecasting,* and *managing personnel*, and
(d) the influence of the presence of the project manager on the frequency with which agile practices are carried out by the teams.

A diagrammatic visualization is given in Fig. 2.

Theory: The role of the project manager in agile projects

The dimensions (a), (b), and (c) represent findings of the qualitative data analysis while the dimension (d) represents the results of the quantitative data analysis. The qualitative data were also seen to supplement and help explain some of the results of the quantitative data analysis. All the constituent elements of our theory are discussed in detail in the following sections.

4. Findings

In this section, we describe each dimension of the role of the project manager in agile projects: the influence of the presence of the project manager on the frequency with which agile practices are carried out by the teams, the project managers' management approaches to everyday activities, and the traditional project management activities performed by the project manager. While the findings in Sections 4.2 and 4.3 emerged from the GT analysis of the qualitative data, the findings presented in Section 4.1 are those that were obtained from the analysis of the pre-interview

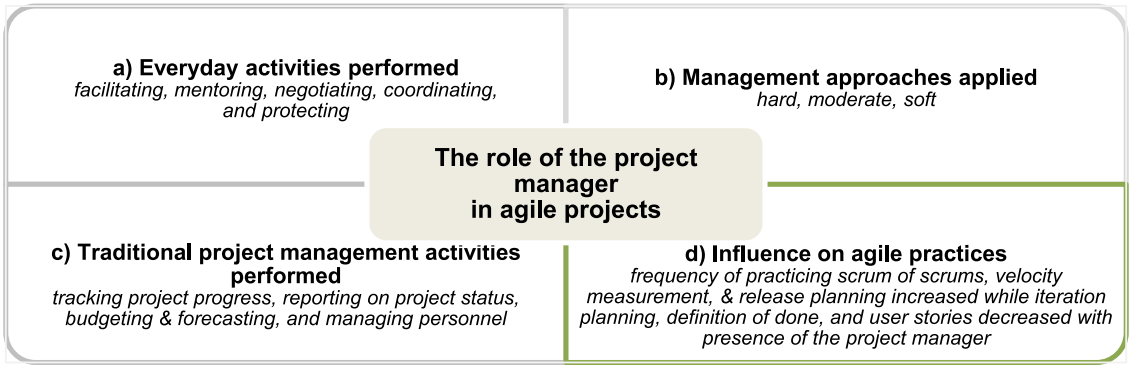

Fig. 2. A grounded theory of the role of the project manager in agile projects described in terms of four dimensions.

Table 2
Influence of the project manager on frequency of use of agile activities by the team.

Agile activities	HFU[a] as % of cases in which PM is present (N=40)	HFU as % of cases in which PM is absent (N=17)	Percentage Difference Level (PDL)
Scrum of scrums meeting	53% (N=21)	29% (N=5)	24%
Project velocity Measurement	68% (N=27)	53% (N=9)	15%
Release planning	80% (N=32)	65% (N=11)	15%
Team based estimation	63% (N=25)	54% (N=9)	9%
Daily scrum/daily standup	90% (N=36)	83% (N=14)	7%
Short iterations/sprints	68% (N=27)	65% (N=11)	3%
Kanban board	75% (N=30)	77% (N=13)	−2%
Product backlog	88% (N=35)	91% (N=16)	−3%
Iterations/sprint reviews	78% (N=31)	83% (N=14)	−5%
Sprint backlog	83% (N=33)	88% (N=15)	−5%
Information visualization	83% (N=33)	94% (N=16)	−11%
Retrospectives	78% (N=31)	94% (N=16)	−16%
Agile games	32% (N=8)	54% (N=7)	−22%
User stories	63% (N=25)	89% (N=16)	−26%
Iterations/sprint planning	43% (N=17)	71% (N=12)	−28%
Definition of done	60% (N=15)	100% (N=13)	−40%

[a]HFU=High Frequency of Use. The percentages in the table correspond to N=40 in cases where project manager was present and N=17 where project manager was not present. The PDL was calculated by the formula PDL= *(HFU as % of cases in which PM is present) - (HFU as % of cases in which PM is absent).*

questionnaire questions. This is an example of the treatment of qualitative and quantitative data in a mixed methods approach to an exploratory GT study.

4.1. The project manager's influence on agile activities

In response to the question, *do agile projects have identified project managers (alongside agile roles)?* We found evidence of the existence of project manager in 70% (N=40 of 57 pre-interview questionnaires) including both primary participants and referred project managers. The product owner was identified to be present by 85% and the scrum master by 83% of the participants.

In response to the question, *what influence, if any, does the project manager exert on the agile practices carried out by the team?* The cross-tabulation analysis results are summarized in Table 2, where the last column "*Percentage Difference Level (PDL)*" shows the difference between the HFU percentages in the second and third columns. A positive value of the PDL indicates that if a project manager is present, there is an increase in the frequency of an agile activity. Whereas a negative PDL value indicates that the frequency goes down when a project manager is present.

As can be seen from Table 2, the activities which exhibited the greatest increase in their frequency of use when a project manager was present were the scrum of scrums meeting, project velocity measurement, and release planning. These three agile activities showed a positive PDL of 15% to 24%. Each of the above agile activities has coordination and facilitation requirements. For example, scrum of scrums is an agile activity which involves holding scrum meetings with representatives from different agile teams working on a common project. This requires someone who can coordinate between different teams and ensure that the meeting takes place. This is supported by our qualitative data analysis where we found that the project manager was highly involved in coordinating across different teams in agile projects. It was also clear that a part of the project manager's role involved traditional project management duties such as tracking project progress, budgeting, and forecasting. This explains why the activities of project velocity measurement and release planning saw a noticeable positive effect in their frequency of use.

On the other hand, activities such as iterations or sprint planning, definition of done, and user stories saw a marked negative PDL when the project manager was present. The magnitude of negative PDL ranged from negative 22% to 40%. We did not find any direct support of this result from our qualitative data analysis of the interview data. Further investigation is needed to understand this indication of a negative influence of the project manager on these agile activities.

4.2. The project managers' approaches to everyday activities

In response to the questions, *what does the project manager do on a regular basis in agile projects?* And, *what management styles or approaches does the project manager adopt when carrying out their role?* we present a mapping of the everyday activities of the

Table 3
Mapping of the project managers' activities with management approaches.

Activity	Description	Management Approach	Examples
Facilitating	Clearing obstacles and issues, facilitating the project teams functioning, increasing process efficiency, and ensuring quality control.	Moderate to Hard	P02,P03, P04,P07, P08,P20, PM7
Mentoring	Empowering the team on the path of self-organization, educating the team and stakeholders in agile practices, and ensuring team adherence to agile practices.	Hard, Moderate, and Soft	P02,P03, P04,P07, P08, P37
Negotiating	Negotiating project funding, issues, scope and commitment to work, with customers, vendors and the development team.	Moderate to Hard	P07,P10, P13, P28, P31
Coordinating	Coordinating project logistics, such as release of deliverables and human capital, and coordinating collaboration between the customers, teams and technical specialists.	Moderate	P02, P07, P17, PM2, PM3, PM6, PM8, PM9
Protecting	Shielding the team from external interference and pushing back on scope creep.	Hard	P06, P08, P28,PM2

project manager with the management approaches in Table 3. These are described in detail below, with examples.

Overall, based on the qualitative interview data and instances of the everyday activities identified, it was seen that nearly all project managers were involved in facilitating, while over half were involved in coordinating and negotiating, less than half were involved in mentoring, and few (approximately one quarter) were involved in protecting. We have defined the management approaches as follows:

- The *soft* approach: This approach is characterized by the project manager adopting a "*hands-off*" approach to the teams day-to-day functioning, encouraging the team to become self-sufficient by acting as a coach, and creating an atmosphere which facilitates open discussions.
- The *moderate* approach: This approach is characterized by the project manager facilitating and coordinating across issues and people in the agile projects. The project manager is ready to jump in and lend a hand to the team whenever necessary.
- The *hard* approach: This approach is characterized by the project manager adopting an assertive and uncompromising stance when the project manager feels that the team or stakeholders attention needs to be focused in a particular channel. This approach is particularly used when there is a persistent obstacle to the teams functioning.

4.2.1. Project managers' approaches to facilitating

Facilitating involves the project manager clearing obstacles and issues, facilitating the project teams functioning, increasing process efficiency, and ensuring quality control. Based on the qualitative data analysis, the project manager's approach to facilitation can be classed as somewhere between *moderate* and *hard*. The hard aspect of facilitation was when an assertive posture was adopted by the project manager. Some of the terms used by participants which illustrated the hard approach were "*driving them*" and "*poke and prod*". While terminology such as "*discuss*", "*jump in*", "*lend a hand*" showed the more moderate to soft approach. However, based on a majority of participant reports (N=17), the project manager's attitude towards facilitation was more aligned between the moderate and hard aspects. The project managers' moderate to hard management approach to facilitating is explained with examples below.

The project managers' approach was determined by who they were dealing with. For example, in a government project, the government body had outsourced development of a key component to a third-party vendor. The vendor used agile while the government organization used waterfall. Before the vendor could start development, technical specialists at the customer end had to clear the documentation. This became a bottleneck and delayed the vendor's development team, who were eager to get started. The project manager (P02) who was an external contractor brought in by the government body, pushed the technical specialists to clear the documentation as soon as possible.

On the other hand, when dealing with the team, the project manager's (P03) attitude was to assist the team. Even if the stakeholders were using a waterfall approach, the project manager (P03) let the team practice documentation at a level where it satisfied stakeholder demands but did not become cumbersome for the team.

Several project managers (e.g. P04, P07, and P08) maintained good relations with the team but did not hesitate to be firm on certain issues. If the project manager (P04) observed that the team's collective voice was being muffled by a particular dominant member, the project manager did not hesitate to address the issue even if it led to possible conflict with the dominant team member. However, P04 ensured that every team member had a say in the meetings as it was essential to empower the whole team. This was very much the hard approach in action. With the rest of the team, P04's approach can be best characterized as moderate.

Transition to agile from waterfall can be confusing to team members and cause conflicts with team members used to a stage-gate model of development. In this situation, transition "*pangs*" amplified minor issues. As was appropriate in this scenario, the project manager (P08) adopted a moderate stance and brought the "*warring parties*" to the negotiating table. P08 was successful in getting a compromise between the developers and testers in this case.

The only instance we came across of a project manager with a soft approach to facilitation was in the case of an entertainment sector project. The project manager (PM7) was a first-time project manager and had a complete hands-off approach towards the team. PM7 let the scrum master (P26) facilitate at the team level, while he took care of the dealings with senior management.

Thus, overall, the project manager's approach to facilitation can be described as between moderate to hard.

4.2.2. Project managers' approaches to mentoring

Mentoring involves the project manager empowering the team on the path of self-organization, educating the team and stakeholders in agile practices, and ensuring team adherence to agile practices. The project managers' approach to mentoring moved between the hard, soft and medium approaches depending on the situation, as explained below.

There was an interesting demonstration of the soft aspect when the project manager was a motivated "*early adopter*" of agile. Here, the project manager (P37) came across the concepts of agile while looking for ways to improve the productivity and gradually "*injected*" agile practices such as the daily stand up without introducing them as agile practices. The team became convinced of the benefits of agile once they had experienced some.

But the soft aspect of *mentoring* really came into play when the project managers doubled up as informal scrum masters. Some indicative terms used by project managers to flag this approach were, "*not constantly checking up*", "*open discussion*", "*like a coach*", and "*becoming self-sufficient*".

On a telecommunications project, the project manager (P03) acted as the scrum master as well. While the team was beginning to mature in agile, the customer organization operated in a very traditional waterfall way. With the team, P03 acted as the scrum master and saw his role more as a coach. The team was enthusiastic about agile with good all-round participation in activities such as planning poker. When interfacing with the customer, P03 assumed the role of the project manager and educated the customer on the rationale behind agile.

The hard aspect of mentoring was when an assertive posture was adopted by the project manager. *Pushed*", "*move along*" were some of the terms used by project managers to describe this assertiveness. For example, on a local government project, the project manager (P02), intervened to keep meetings on track and within time. In this scenario, the team was taking its first steps in agile and needed a lot of oversight. Product demonstrations were one area where the team became lax. P02 had to push them to hold regular demonstrations. P02 also informally set up one of the end users as the product manager and facilitated regular team interactions with them.

This hard aspect was also seen in a banking sector project where although the organization was mature in agile, the team was essentially composed of beginners in agile. Initially, enforcing agile practices remained a large part of the project manager's (P07) repertoire. As the team matured in agile, P07 lessened his degree of control until a stage was reached when the team became more self-organizing. In P07's own words, "*they could have operated without me needing to be there*".

A mixture of the hard and soft aspect gave rise to the middle path or the moderate approach. This was seen on another local government project, where the project manager (P04), was managing three different teams. P04 personally set up and ran the daily stand up meeting and did a combined stand up with all the teams once a week. The team was transitioning to agile from a waterfall environment. The project manager gave the team a lot of freedom to challenge existing practices including agile and gave them a free hand with the project documentation. Here, P04's role was evolving into a hybrid of the scrum master and project manager roles. In fact, P04 saw his role more as a coach than as a controlling project manager.

4.2.3. Project managers' approaches to negotiating

Negotiating involves the project manager negotiating project funding, issues, scope and commitment to work, with customers, vendors and the development team.

In terms of negotiation, the project manager's approach fell between hard and moderate approaches. Negotiation is part of the project manager's traditional responsibilities (Benington, 1987) and even in some agile projects the project manager oversaw negotiations. This view was supported by several participants (P02, P08, P10, P13, P28, P31, and P35) who saw negotiation as one of the key responsibilities of the project manager. Some terms indicative of this approach were: "*get to the point*", "*negotiate*", "*plan*", "*convince*", and "*talk*".

A good example of the likely need for the project manager to exist alongside the product owner in an agile project was demonstrated in the case of P07. While the product owner's role is to act as the customer representative, the project manager's key focus was to ensure that all project deliverables were completed successfully. The project manager could adopt a hard stance on behalf of the team when negotiating funding with the customers. In the case of P07, the project manager told the senior management that the project deliverables would not be met without further funding, backing this up with a strong business case. It was generally seen that scrum masters, though sometimes involved in negotiating certain aspects with the customers such as project scope, steered clear from tracking finances and negotiating funding. These were seen to fall under traditional project management activities covered by the project manager (described further in the next sub-section).

This moderate to hard continuum was also useful when dealing with vendors, who sometimes needed to be chased up for delivery of features.

4.2.4. Project managers' approaches to coordinating

Coordinating involves the project manager coordinating project logistics, such as release of deliverables and human capital, and coordinating collaboration between the customers, teams and technical specialists.

The project manager's approach to coordinating was moderate. "*Come together*" "*discuss*", "*coordinate*", "*overall delivery*", "*release process*", and "*right resources*" were some of the terms used by project managers to indicate coordination.

Coordination usually involved coordinating the release of deliverables, resourcing and staffing the teams, liaising with senior management, and sprint coordination. A majority of the project managers (P02, P07, P17, PM2, PM3, PM6, PM8, and PM9) were responsible for coordinating the delivery of features completed by multiple teams. The project manager was responsible for ensuring that the deliverables met the standards and the release was done in synchronization. One of the project managers (P17) termed the entire multiple team environment as an "*ecosystem*". P17 identified his key responsibility was to keep the ecosystem functioning. In projects with multiple teams, the project manager (PM2, PM9) looked after sprint coordination. The project manager's role as the coordinator also finds indirect confirmation from our quantitative data (see Section 4.1). The project manager's presence possibly led to an increase in the scrum of scrums meeting, which exhibited a PDL of nearly 24%. This could mean that the project manager is leading multi-team projects which necessitate agile activities such as the scrum of scrums meeting, which in turn are key coordination mechanisms in large scale agile. This aspect will be further discussed in Section 5.

4.2.5. Project managers' approaches to protecting

Protecting involves the project manager shielding the team from external interference and pushing back on scope creep. The project manager adopted the hard approach to protect the team from external disruptions. The hard stance was particularly on display when the project manager was trying to prevent scope creep (e.g. P08, P28). The project manager acted as a barrier against last minute changes by the customers. This naturally called for a hard or firm approach. However, this hard

Table 4
A comparison of project manager activities from our study with the scrum guide.

	Role responsible for the activity in the scrum guide
Project Manager activities from our study which map to the activities in the scrum guide	
Clearing obstacles and issues	scrum master
Facilitating the project teams functioning	scrum master
Increasing process efficiency	scrum master
Ensuring quality control	scrum master
Empowering the team on the path of self-organization	scrum master
Educating the team and stakeholders in agile practices	scrum master
Ensuring team adherence to agile practices	scrum master
Coordinating collaboration between the customers, teams and technical specialists	scrum master
Shielding the team from external interference	scrum master
Pushing back on scope creep	scrum master
Project Manager activities not in the scrum guide	
Tracking project progress	Not in the scrum guide
Reporting on project status	Not in the scrum guide
Budgeting and forecasting	Not in the scrum guide
Managing personnel	Not in the scrum guide
Negotiating project funding, issues, scope and commitment to work, with customers, vendors and the development team	Not in the scrum guide
Coordinating project logistics, such as release of deliverables and human capital	Not in the scrum guide

approach did not emerge in isolation. The project manager built up good collaborative relationships with the stakeholders by frequent meetings and clear communications. It was this moderate approach during business-as-usual which gave the project manager confidence to take a hard stance when protecting the team.

The project manager (PM2), who oversaw external and internal project dependencies, actively pushed back on the scope creep. P06 had a very good collaborative association with the project manager and consulted PM2 frequently regarding the impact of changes on the project.

4.3. Traditional project management activities

In response to the question, *are there any traditional project manager activities still carried out in agile projects?* We found the project manager performing some traditional project management activities. As several participants pointed out, even with the team being self-organizing, there was the need for a role to tackle external dependencies and administrative overheads.

Tracking project progress: One of the most common activities of the project manager was to keep an eye on the project's progress. This was done by observing the agile project management tools such as JIRA (P05) and burn up charts, and by tracking project velocity (P06, P07, P27, and P31) which the teams maintained, and creating custom project documents to track progress. Tracking progress was done in either "*active*" or "*passive*" modes. The *active* mode involved talking with team members and gaining insight into the bottlenecks which were holding up project velocity. The *passive* mode involved tracking project velocity using tools like those used in Kanban.

Reporting on project status: Project reporting was generally driven by traditional management layers which overlaid onto the agile teams (P04, P08, P10, and P14). In the case of a public-sector project, the project manager (P04) clearly identified that fixed deadlines (delivery dates) and budgets drove the reporting. This necessitated periodic generation of status reports destined for a variety of stakeholders.

"*From a reporting point of view, I had to follow their reporting methods. I had to kind do all the finances, kind of like a burnup chart*". – P04, Project Manager, New Zealand.

Budgeting and forecasting: Interestingly, one of the possible reasons for the continued existence of the project manager in agile projects was given by a scrum master, who opined that the project manager was needed to carry out project management activities such as financial management and reporting, as these fell outside the domain of the team. Financial tracking along with generating regular project status reports was identified as part of the daily administration responsibility of the project manager (P07, P08). Additionally, dealing with customer invoices was also a part of the project manager's responsibility (P08, P14).

Managing Personnel: Another administration aspect was personnel management which included conducting routine performance reviews and appraisals (P09, P39), career progression planning, resolution of outstanding personnel issues (P20), and holidays (P39).

5. Discussion and implications

As mentioned in Section 2.4, cornerstone practitioner literature on Agile, particularly that on Scrum, does not envision any role for the project manager in its framework (Sutherl and Schwaber, 2017). However, as we have shown in our present and previous study (Shastri et al., 2016), the role of the project manager is still in existence even in agile projects. Thus, our findings are particularly relevant as they allow us to present an approximation of where exactly the project manager fits in the agile scheme of things in the software industry. In Table 4 we have presented a mapping of the activities of the project manager (from our study) and the similarity or point of difference with those given in the scrum guide. From Table 4, it can be seen that most of the activities performed by the project manager in our study should be, as per the scrum guide, performed by the scrum master. Additionally, Table 4 also shows the traditional project management activities that are still carried out by project manager even in agile projects. These activities do not have any analogous activity assigned to any of the scrum roles (scrum master, product owner, and development team) in the scrum guide. This gap between practitioner literature and our findings, does seem to indicate that there is a need for a role (like the project manager) who can track project progress, do budgetary reporting, and conduct personnel management.

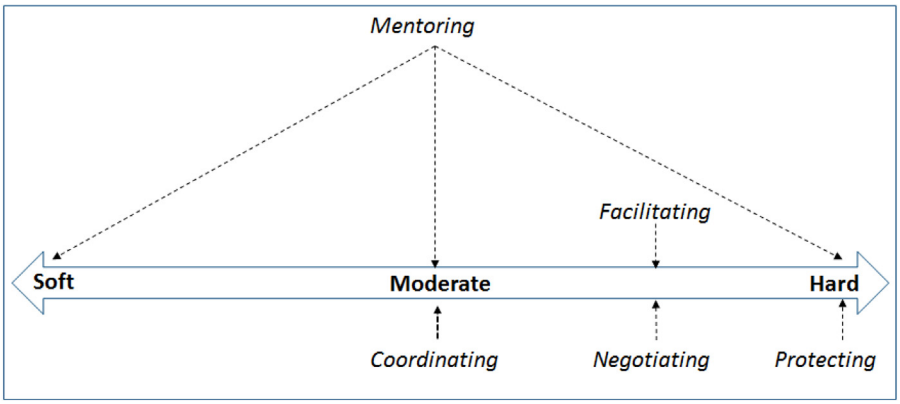

Fig. 3. The relation between the activities and the management approaches adopted by the project manager.

Additionally, there were some important considerations which dictated what stance the project manager adopted: (a) Was the project manager playing a double role, that is, acting as the scrum master as well; (b) Was the project manager an enthusiastic early adopter of agile; and (c) Was the team transitioning to agile? Each of these points is discussed below.

The duality of the project manager's role is one of the findings of our research. We found conclusive evidence of the project manager simultaneously performing the scrum master role. This is also reflected in the mapping presented in Table 4. Depending on the organizational context and the circumstances of the project, the project manager could be called upon to perform the role of the scrum master (P06, P09, P10, P14, P17, P20, P28, and P35). The project manager thus had to adopt a dual approach to deal with different entities such as the development team and stakeholders. One of the common reasons for the duality of the project manager's role was organizational resource constraints and the organization transitioning from waterfall to agile. This also seems to suggest that the project manager role might be interim while the organization grapples with the transition to agile. So long as the need for traditional project management activities continue in an organization (for example, due to their unique contexts), the project manager role is likely to be present.

Another key finding was the management approach adopted by the project managers when dealing with customers, team members, and suppliers. The inter-relationship between the activities and the management approaches is shown in Fig. 3. A majority of the project managers were found to adopt a moderate approach, closely followed by the number of project managers adopting a hard approach. In fact, the soft approach was adopted by very few project managers. It was found that some project managers had a consistent approach across different activities (such as facilitating and mentoring). The mapping of the approach with the activity is given in Table 3.

Another aspect was that a few project managers had a consistent approach across *different activities*. For example, P07 and P02 adopted a hard approach across different activities such as facilitating, mentoring, and negotiating. The key point is that in these cases it was the project manager who was taking the initiative to introduce agile to the team members.

On the process side of things, some project managers took care of the scrum ceremonies such as running the daily standups, the sprint, and could also write user stories. While the traditional project management aspect consisted of personnel management, negotiation, project delivery and planning. This suggests that the project manager's role is still needed in performing non-agile (but essential) activities such as performance reviews and multiple team coordination.

The project manager playing the role of the scrum master has certain advantages and disadvantages. The advantage is that the project manager is aware of the constraints faced by the team and in this formal title can effectively liaise with stakeholders to resolve those constraints. On the flip side, the roles of the scrum master and project manager calls for different perspectives on certain issues. It is also possible that if the project manager is an early adopter of agile he might convince the team to transition to agile. This is an aspect which merits further investigation.

With traditional organizations looking to experiment with agile, the managers in the pilot agile teams often find themselves in a difficult situation. This is due to limited freedom to innovate, a large traditional working overhead, and the feeling of impermanence stemming from the agile implementation. Our recommendation to senior management is to be clear as to the purpose of adopting agile. Additionally, senior management needs to avoid the scenario of project manager's trying to satisfy waterfall and agile requirements simultaneously.

This study will help guide new and existing project managers to better understand the various aspects and boundaries of their new roles on agile projects and enable them to better facilitate self-organizing teams. Our classification of the approaches *(hard, moderate, and soft*) to performing the everyday activities will help managers to implement a behavioral pattern depending on the context. Our statistical data also provides project managers with tips on which agile practices are likely to benefit from their presence and which practices are best handed over to the team. Gradually withdrawing from controlling agile practices will help the team become self-organizing by putting trust in the team's ability to take decisions. Another more concrete implication is that our findings can be used to create a job description for the agile project manager. For example, the person specification in the job description could read as follows:

"*We are looking for versatile and adaptable Project Manager (PM) with strong facilitation skills. The PM should be the grease that drives the team machine as it sets new benchmarks in quality and delivery while staying true to agile principles. The PM needs to be a mentor to whom team members can look upto, a great wall behind whom the team can carry on their work uninterrupted, a lithe negotiator who can negotiate the best deal for the company and the team, and a master coordinator who can juggle often conflicting requirements with ease*".

The above job description is quite rudimentary but it does give an idea of the ways in which the findings of this study can be used by practitioners.

6. Related work

Different roles in ASD (such as product owner, scrum master, the customer and the team) have been studied to varying extents

in the past few years. Researchers have identified different activities carried out by the scrum master (Noll et al., 2017; Bass, 2014). The role of the product owner has also been in the spotlight and researchers have investigated the role in the industry vis-à-vis the theoretical definition (Bass, 2013; Bass and Haxby, 2019; Bass et al., 2016; Kristinsdottir et al., 2016; Oomen et al., 2017), and (Sverrisdottir et al., 2014). The team in ASD projects has been studied from the viewpoint of self-organization (Hoda et al., 2012, 2013), the transition to self-organization (Hoda and Noble, 2017), project management challenges from the perspective of the self-organizing team (Hoda and Murugesan, 2016), and the effect of customer collaboration (Hoda et al., 2011).

However, there is a scarcity of prior empirical work dealing decisively with the role of the project manager in agile projects. Existing literature has focused on understanding the reasons behind project managers' preferences in agile contexts (Bishop et al., 2018), the effect of agile adoption on the working styles of the project manager (Taylor, 2016), and the project manager's perspective on conflicts in agile teams (Siddique and Hussein, 2016).

Our finding of the project manager's dual role as the scrum master supports a recent study on the role of scrum masters that reported a high frequency of project manager acting as the scrum master (Noll et al., 2017). The project manager acting as the customer proxy was observed by Taylor (Taylor, 2016) but we did not find evidence of this in our study.

Interestingly, one research area which corroborates some of our findings is recent research on scaled agile (Scheerer et al., 2014; Smite et al., 2019; Gustavsson, 2019; Conboy and Carroll, 2019; Moe et al., 2019), and (Dikert et al., 2016). As mentioned in Section 2.5, intra-team coordination is one aspect of scaled agile which has received attention from researchers. One of the key challenges identified by Dikert et al. (2016) in extant literature was that of multiple team coordination, where traditional and agile methods existed side by side. Another challenge, in our view linked to the coordination one, identified by Dikert et al. (2016) was the conflict between the need for additional management positions to manage the scaling-up with the need to encourage self-organization. One of the studies included by Dikert et al. (2016) in their SLR has noted the need for project managers to act as change agents. Our activity of coordination, carried out by the project manager using the moderate approach, reflects the project manager's approach slowly undergoing a metamorphosis into a more agile approach.

Another finding from our study is that the project manager possibly had a high level of influence on the scrum of scrums meeting (see Section 4.1). Gustavsson (2019) looked at the manner in which coordination routines such as the scrum of scrums are enacted and the reasons they are enacted in a particular manner. Gustavsson (2019) found that the scrum of scrums was tailored by organizations to suit their project context. This could involve varying the frequency of meetings or throwing open participation to a wider section of stakeholders. Our study also suggests that the high level of influence of the project manager on Scrum-of-Scrum (SOS) meetings could be due to the project manager coordinating multiple teams.

Paasivaara and Lassenius (2019) reported that one of the ways in which large firms such as Ericsson dealt with coordination challenges over large multi-team projects was to facilitate community based decision making by setting up communities of practice (COP's). In this scenario the teams followed a democratic decision making process and even product owners had very limited say in this process. One of the key characteristics of the COP identified was the presence of facilitator who kept the teams on track (Paasivaara and Lassenius, 2019). A similar approach was reported at Spotify by Smite et al. (2019), who identified Spotify's culture of guilds as one of the mechanisms for facilitating "*lightweight coordination*" across regions. In both the studies neither Ericsson nor Spotify had any project managers. However, our findings suggest that in project where the project manager is present, he or she is involved in performing facilitation activities.

Conboy and Carroll (2019), based on their long running research on adoption and sustenance of agile frameworks, identified nine challenges associated with adopting large scale agile frameworks. These challenges included foundational issues such as developing a definition of large scale agile appropriate to the organizational context and maintaining the spirit of self-organization when undertaking the transformation. Researchers have also studied the role of the product owner in large scale agile (Bass and Haxby, 2019) and reported on the creation of product owner teams. The members of these teams undertook a range of activities including communication and risk assessment.

Team autonomy is another area of large scale agile that has been touched upon by researchers (Moe et al., 2019). The key challenges the researchers reported were that of goals being set without the involvement of the teams and interference in the teams work due to external dependencies. Our activity of protecting performed by the project manager with its attendant *hard* approach is a possible solution to the above mentioned challenge.

The focus of all the studies mentioned above is on a particular aspect of agile or the project manager's perspective on issues which affect the team and the project in ASD. None of the above-mentioned studies have explored the project manager's day-to-day role and responsibilities. Our study fills this critical research gap.

7. Limitations

A Grounded Theory study generates a "*mid-ranged*" theory that is limited to the contexts studied and remains open to modification based on new data to suit new contexts (Glaser, 1992). Our participants represent a wide range of project sectors, ranging from telecommunications and banking, to government, tourism and retail. Through theoretical sampling, we managed to include a variety of roles such as developers, solutions architects, and test engineers in addition to various management roles to provide multiple perspectives. Our theory is open to modification and extension based on future research work in different contexts.

In using a mixed data Grounded Theory method, we have made use of a pre-interview questionnaire which provided quantitative data while the subsequent interviews provided qualitative data. The modest sample size of the quantitative data (N=57) is supplemented with a decent sample size of qualitative data (over 45 h of interviews with 39 participants).

To ensure the reliability of the data and to prevent mixing up of participant experiences from different projects, these were carefully coded (as explained in the section on research methodology). The responses to the questionnaire were kept at hand during the interview process and any contradictions (for example, regarding project details) were clarified with the participants directly.

In a Grounded Theory study, the verifiability of the theory generated can be assessed by the rigor of the research method and evidence that the theory has emerged from the collected data (Glaser, 1992; Glaser and Holton, 2004). To ensure verifiability and to keep our coding procedures as transparent as possible, we have provided explanations of our coding procedures and the process of derivation of concepts and categories in the research methodology section.

One of the key limitations of the quantitative study is that it looked at a limited set of variables. Given the exploratory nature of the study, variables were limited to those that emerged as significant across the participant base. Factors such as size of project

and whether it is a multi-team project were not factored into the analysis or the questionnaire. The above factors can influence whether management decides to utilize the project manager to coordinate multiple project teams and even dictate the level of involvement of the project manager in practices such as scrum of scrums. These aspects can be explored in more structured studies in the future.

The possibility of sampling and response bias in quantitative data collection was minimized by adopting targeted strategies to minimize bias. To reduce the likelihood of sampling bias we recruited not only project managers, but a diverse sample of participants such as developers, testers, product owners, and scrum masters in addition to Project Managers.

Response bias was minimized in a majority of cases by verifying the completed pre-interview questionnaire with the interview participants during the interview. As with most such studies, response bias in terms of not being able to include those who did not wish to participant in the study remains.

8. Conclusion

In this paper we have presented the role of the project manager in ASD projects. The findings are based on interviews with 39 agile practitioners and the pre-interview questionnaire with 57 respondents.

The key contribution of this study is the mixed data Grounded Theory of the role of the project manager. We have presented the theory in terms of the dimensions of the project manager's role in agile projects. One of the dimensions is the project manager performing activities such as facilitating, mentoring, negotiating, coordinating, and protecting. Some of these activities fall in the domain of the scrum master as per classic agile literature (Highsmith, 2004; Sutherl and Schwaber, 2017). We observed this unique arrangement reported by nearly one-fifth of our participants.

Another closely related dimension is that of the project manager adopting a *hard, moderate*, or *soft* management approach depending on the activity performed by the project manager. As illustrated in Fig. 3, the project managers applied all three management approaches to mentoring. While in the case of coordinating, the project manager's approach was moderate. The nature of negotiations meant that the project managers' approach was between moderate and hard approaches. Their approach to facilitating was also between moderate and hard. To protect the team from external disruptions project managers adopted the hard approach. The management approach used was partly influenced by different factors such as: the project manager also playing the scrum master role; the project manager being an enthusiastic early adopter of agile; and the team was transitioning to agile. The project manager while carrying out the activities also carried out traditional duties such as project tracking, project reporting, budgeting & forecasting, and personnel management. Thus, our findings suggest that the project manager's role is needed in performing non-agile activities which are essential within an organizational framework. Additionally, the phenomenon of the project manager acting as the scrum master suggests that this arrangement is transitory till the time the organization can implement agile.

Additionally, the pre-interview survey data provides evidence that the project manager's presence on an agile project can possibly influence the frequency with which agile practices are carried out by the team. The agile practices which exhibited the greatest increase in their frequency of use in the presence of a project manager included scrum of scrums meeting, project velocity measurement, and release planning. Practices such as sprint planning, definition of done, and user stories saw a marked negative trend when the project manager was present. This aspect in particular merits further investigation.

Our findings show that the role of the project manager in agile projects involves performing everyday activities such as *facilitating* and *mentoring*, using hard, moderate, and soft management approaches, while retaining some the traditional project management duties such as tracking project progress and budgeting. Understanding the role of the project manager in agile projects will help practitioners better manage expectations of this role and ease their agile transitions.

CRediT authorship contribution statement

Yogeshwar Shastri: Conceptualization, Methodology, Investigation, Formal analysis, Writing - original draft. **Rashina Hoda:** Conceptualization, Methodology, Resources, Writing - review & editing, Supervision. **Robert Amor:** Conceptualization, Methodology, Resources, Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

References

Bass, J.M., 2013. Agile method tailoring in distributed enterprises: Product owner teams. In: IEEE 8th Int. Conf. on Global Software Eng., Bari, pp. 154–163.

Bass, J.M., 2014. Scrum master activities: Process tailoring in large enterprise projects. In: 2014 IEEE 9th International Conference on Global Software Engineering, pp. 6–15.

Bass, J.M., Beecham, S., Noll, J., Razzak, M.A., 2016. All Hands to the Pumps: The Product Owner Role in Small Companies Lero Technical Report: 2017.

Bass, J.M., Haxby, A., 2019. Tailoring product ownership in large-scale agile projects: managing scale, distance, and governance. IEEE Softw. 36 (2), 58–63.

Beck, K., Andres, C., 2005. Extreme Programming Explained : Embrace Change. Addison-Wesley, Boston, MA.

Benington, H.D., 1987. Production of large computer programs. In Proc. of the 9th Int. Conf. on Software Eng. Monterey, CA, March 30–April, pp. 299–310.

Bishop, D., Rowland, P., Noteboom, C., 2018. Antecedents of preference for agile methods: A project manager perspective. In: Proc. of the 51st Hawaii Int. Conf. on System Sciences.

Boehm, B.W., 1988. A spiral model of software development and enhancement. Computer 21 (5), 61–72.

Boehm, B., 2006. A view of 20th and 21st century software engineering. In: Proc. of the 28th Int. Conf. on Software Eng. Shanghai, pp. 12–29.

Boehm, B.W., Ross, R., 1989. Theory-W software project management principles and examples. IEEE Trans. Softw. Eng. 15 (7), 902–916.

Bredillet, C., Tywoniak, S., Dwivedula, R., 2015. What is a good project manager? An Aristotelian perspective. Inter. J. Proj. Manage. 33 (2), 254–266.

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: Int. Conf. of Software Process Improvement and Capability Determination, pp. 177–189.

Charmaz, K., 2006. Constructing Grounded Theory. Sage Publications Ltd., Thousand Oaks, CA.

Chow, T., Cao, D.-B., 2008. A survey study of critical success factors in agile software projects. J. Syst. Softw. 81 (6), 961–971.

Cicmil, S., Hodgson, D., 2006. New possibilities for project management theory. Proj. Manage. J. 37 (3), 111–122.

Cicmil, S., Williams, T., Thomas, J., Hodgson, D., 2006. Rethinking project management: researching the actuality of projects. Inter. J. Proj. Manage 24 (8), 675–686.

Cockburn, A., Highsmith, J., 2001. Agile software development, the people factor. Computer 34 (11), 131–133.
Conboy, K., Carroll, N., 2019. Implementing large-scale agile frameworks: challenges and recommendations. IEEE Softw. 36 (2), 44–50.
Deemer, G.B.P., Larman, C., Vodde, B., 2012. The Scrum Primer. [Online]. Available: http://scrumprimer.org/.
Digital.ai Software Inc., 2020. VersionOne 14th Annual State of Agile Report. Digital.ai Software Incorporated., [Online]. Available: https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494.
Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. J. Syst. Softw. 119, 87–108.
Dingsøyr, T., Moe, N.B., 2013. Research challenges in large-scale agile software development. ACM SIGSOFT Softw. Eng. Notes 38 (5), 38–39.
Drury-Grogan, M.L., O'dwyer, O., 2013. An investigation of the decision-making process in agile teams. Int. J. Inform. Technol. Decis. Making 12 (06), 1097–1120.
Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. Inform. Softw. Technol. 50 (9–10), 833–859.
Fowler, M., Highsmith, J., 1991. The agile manifesto. Softw. Develop. 9 (8), 28–35.
Gaddis, P.O., 1959. The project manager. Harvard Bus. Rev. 37 (3), 89–97.
Glaser, B.G., 1978. Theoretical Sensitivity : Advances in the Methodology of Grounded Theory. Sociology Press, Mill Valley, Calif.
Glaser, B.G., 1992. Basics of Grounded Theory Analysis. Mill Valley, Calif.: Sociology Press.
Glaser, B.G., Holton, J., 2004. Remodeling grounded theory. Forum: Qual. Soc. Res. 5 (2), 47–68.
Gustavsson, T., 2019. Dynamics of inter-team coordination routines in large-scale agile software development. Dynamics 5, 15–2019.
Highsmith, J.A., 2004. Agile Project Management : Creating Innovative Products. Addison-Wesley, Boston.
Hoda, R., Murugesan, L.K., 2016. Multi-level agile project management challenges: A self-organizing team perspective. J. Syst. Softw. 117, 245–257.
Hoda, R., Noble, J., 2017. Becoming agile: a grounded theory of agile transitions in practice. In: Proc. of the 39th Int. Conf. on Software Eng., Argentina, pp. 141–151.
Hoda, R., Noble, J., Marshall, S., 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. Inform. Softw. Technol. 53 (5), 521–534.
Hoda, R., Noble, J., Marshall, S., 2012. Developing a grounded theory to explain the practices of self-organizing Agile teams. Empir. Softw. Eng. 17 (6), 609–639.
Hoda, R., Noble, J., Marshall, S., 2013. Self-organizing roles on agile software development teams. IEEE Trans. Softw. Eng. 39 (3), 422–444.
Hodgson, D.E., Paton, S., 2016. Understanding the professional project manager: Cosmopolitans, locals and identity work. Inter. J. Proj. Manage. 34 (2), 352–364.
Kristinsdottir, S., Larusdottir, M., Cajander, Å., 2016. Responsibilities and challenges of product owners at spotify-an exploratory case study. In: Human-Centered and Error-Resilient Systems Development. Springer, pp. 3–16.
Larman, C., Basili, V.R., 2003. Iterative and incremental development: A brief history. Computer 36 (6), 47–56.
Loufrani-Fedida, S., Missonier, S., 2015. The project manager cannot be a hero anymore! understanding critical competencies in project-based organizations from a multilevel approach. Inter. J. Proj. Manage. 33 (6), 1220–1235.
Moe, N.B., Dahl, B., Stray, V., Karlsen, L.S., Schjødt-Osmo, S., 2019. Team autonomy in large-scale agile. In Proceedings of the 52nd Hawaii International Conference on System Sciences.
Morris, P.W., 1997. The Management of Projects. Thomas Telford.
Morris, P.W., Crawford, L., Hodgson, D., Shepherd, M.M., Thomas, J., 2006. Exploring the role of formal bodies of knowledge in defining a profession–The case of project management. Inter. J. Proj. Manage. 24 (8), 710–721.
Müller, R., Turner, J.R., 2007. Matching the project manager's leadership style to project type. Inter. J. Proj. Manage. 25 (1), 21–32.
Nkukwana, S., Terblanche, N.H.D., 2017. Between a rock and a hard place: Management and implementation teams' expectations of project managers in an agile information systems delivery environment. South Afr. J. Inform. Manage. 19 (1).
Noll, J., Razzak, M.A., Bass, J.M., Beecham, S., 2017. A study of the Scrum Master's role. In: Int. Conf. on Product-Focused Software Process Improvement, Innsbruck, pp. 307–323.
Oomen, S., De Waal, B., Albertin, A., Ravesteyn, P., 2017. How can scrum be succesful? Competences of the scrum product owner. In: Proceedings of the 25th European Conference on Information Systems (ECIS), Guimarães, Portugal, pp. 5–10.
Paasivaara, M., Lassenius, C., 2019. Empower your agile organization: Community-based decision making in large-scale Agile development at ericsson. IEEE Softw. 36 (2), 64–69.
Palmer, S.R., Felsing, M., 2001. A Practical Guide to Feature-Driven Development. Pearson Education.
Pettersen, N., 1991. What do we know about the effective project manager?. Int. J. Project Manage. 9 (2), 99–104.
Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J., 2008. The impact of agile practices on communication in software development. Empir. Softw. Eng. 13 (3), 303–337.
Pinto, J.K., Winch, G., 2016. The unsettling of settled science: The past and future of the management of projects. Inter. J. Proj. Manage. 34 (2), 237–245.
Project Management Institute, 2017a. Annual Report. [Online]. Available: https://www.pmi.org/annual-report-2017/at-a-glance.
Project Management Institute, 2017b. Agile Practice Guide, first ed. PMI, Newtown Square Pennsylvania.
Project Managerment Institute, 2018. A Guide to the Project Management Body of Knowledge, sixth ed. PMI, Newtown Square, PA.
Robey, D., Welke, R., Turk, D., 2001. Traditional, iterative, and component-based development: A social analysis of software development paradigms. Inform. Tech. Manage. 2 (1), 53–70.
Rowen, R.B., 1990. Software project management under incomplete and ambiguous specifications. IEEE Trans. Eng. Manage. 37 (1), 10–21.
Royce, W.W., 1987. Managing the development of large software systems. In: Proc. of IEEE WESCON, 26, Los Angeles.
Scheerer, A., Hildenbrand, T., Kude, T., 2014. Coordination in large-scale agile software development: A multiteam systems perspective. In: 2014 47th Hawaii International Conference on System Sciences. IEEE, pp. 4780–4788.
Schwaber, K., Beedle, M., 2002. Agile Software Development with Scrum. Prentice-Hall, Upper Saddle River, NJ.
Shastri, Y., Hoda, R., 2020. Survey of roles in agile project management and their involvement in agile practices. http://dx.doi.org/10.7910/DVN/QEASRH, Harvard Dataverse, V1.
Shastri, Y., Hoda, R., Amor, R., 2016. Does the project manager still exist in agile software development projects? In: 23rd Asia-Pacific Software Eng. Conf., Hamilton, pp. 57–64.
Shastri, Y., Hoda, R., Amor, R., 2017. Understanding the roles of the manager in Agile project management. In: Proc. of the 10th Innovations in Software Eng. Conf., Jaipur, pp. 45–55.
Siddique, L., Hussein, B.A., 2016. Grounded theory study of conflicts in norwegian agile software projects: the project managers' perspective. J. Eng. Project Prod. Manage. 6 (2), 120–135.
Smite, D., Moe, N.B., Levinta, G., Floryan, M., 2019. Spotify guilds: how to succeed with knowledge sharing in large-scale agile organizations. IEEE Softw. 36 (2), 51–57.
Stapleton, J., 2003. DSDM : Business Focussed Development. Addison-Wesley, Boston.
Stevenson, D.H., Starkweather, J.A., 2010. PM Critical competency index: IT execs prefer soft skills. Inter. J. Proj. Manage. 28 (7), 663–671.
Stol, K.-J., Ralph, P., Fitzgerald, B., 2016. Grounded theory in software engineering research: a critical review and guidelines. In: Proc. of the 38th Int. Conf. on Software Eng. Austin, pp. 120–131.
Stray, V., Sjøberg, D.I.K., Dybå, T., 2016. The daily stand-up meeting: A grounded theory study. J. Syst. Softe. 114, 101–124.
Sutherl, J., Schwaber, K., 2017. The scrum guide: the definitive guide to scrum. Scrumguides.org. [Online]. Available: https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100.
Sverrisdottir, H.S., Ingason, H.T., Jonasson, H.I., 2014. The role of the product owner in scrum-comparison between theory and practices. Proc.-Soc. Behav. Sci. 119, 257–267.
Tavakol, M., Dennick, R., 2011. Making sense of Cronbach's alpha. Int. J. Med. Educ. 2, 53–55.
Taylor, K.J., 2016. Adopting agile software development: the project manager experience. Inform. Technol. People 29 (4), 670–687.
VersionOne Inc., 2018. The 12th Annual State of Agile Report. VersionOne Incorporated., [Online]. Available: https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report.
VersionOne Inc., 2019. The 13th Annual State of Agile Report. VersionOne Incorporated., [Online]. Available: https://stateofagile.com/#ufh-i-613553418-13th-annual-state-of-agile-report/7027494.
Waterman, M., Noble, J., Allan, G., 2015. How much up-front?: a grounded theory of agile architecture. In: Proc. of the 37th Int. Conf. on Software Eng. Florence, pp. 347–357.

Yogeshwar Y. Shastri completed Ph.D. degree from the Department of Electrical, Software and Computer Engineering at The University of Auckland, New Zealand. He received his Master's degree from the University of Sheffield, UK and his Bachelor's degree from Amravati University, India. Email address: ysha962@aucklanduni.ac.nz.

Rashina Hoda, Ph.D. (Victoria University of Wellington), B.Sc. Hons (Louisiana State University), is an Associate Dean (Academic Workforce) and an Associate Professor in software engineering at the Faculty of Information Technology at Monash University. Her research focuses on human-centered software engineering, agile software development, and grounded theory. Rashina received a distinguished paper award at ICSE2017 and a distinguished reviewer award at ICSE2020. She serves on the IEEE Transactions on Software Engineering reviewer board and IEEE Software advisory panel, as associate editor for the Journal of Systems and Software and on the program and organizing committees for ICSE and XP conferences. Rashina is currently writing a book on Grounded Theory for Software Engineering. For more information please visit: www.rashina.com.

Robert Amor received his B.Sc. (Hons) and M.Sc. degrees from Victoria University of Wellington, New Zealand. After receiving his Ph.D. from the University of Auckland, New Zealand he spent five years working at the Centre for Construction IT at the BRE in the UK. He is currently in the School of Computer Science at the University of Auckland and a professor in that department. Prof Amor undertakes research in the field of Construction Informatics. Achieving interoperability is his core research interest and to achieve this he investigates integrated environments which covers information modeling (e.g., BIM), process modeling, user interaction, implementation frameworks, information mapping, and communication strategies. Since 2003 he has coordinated the working group W78 (IT for Construction) for the International Council for Research and Innovation in Building and Construction. He is also Editor-in-Chief of the Journal of Information Technology in Construction.