

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/312077636>

Understanding the Roles of the Manager in Agile Project Management

Conference Paper · February 2017

DOI: 10.1145/3021460.3021465

CITATIONS

17

READS

3,116

3 authors:



Yogeshwar Shastri

University of Auckland

7 PUBLICATIONS 47 CITATIONS

[SEE PROFILE](#)



Rashina Hoda

Monash University (Australia)

98 PUBLICATIONS 1,542 CITATIONS

[SEE PROFILE](#)



Robert Amor

University of Auckland

219 PUBLICATIONS 1,574 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Self-organizing Teams [View project](#)



Building Quake & People – a serious game platform for informing life saving strategies [View project](#)

Understanding the Roles of the Manager in Agile Project Management

Yogeshwar Shastri
Department of Electrical and
Computer Engineering
The University of Auckland
Auckland, New Zealand
+642108507449

ysha962@aucklanduni.ac.nz

Rashina Hoda
Department of Electrical and
Computer Engineering
The University of Auckland
Auckland, New Zealand
+6499231377

r.hoda@auckland.ac.nz

Robert Amor
Department of Computer Science
The University of Auckland
Auckland, New Zealand
+6493737599
trebor@cs.auckland.ac.nz

ABSTRACT

Agile project management (APM) does away with the role and the job title of the manager and instead places emphasis on self-organizing teams. However, recent surveys show that the job title of managers, particularly the project manager, is in existence on a significant number of agile projects. At the same time there is very little empirical evidence on the manager's role in an APM framework. To address this issue, a Grounded Theory study involving 20 software professionals from 18 different organizations which employed Agile Software Development (ASD) was carried out. The key finding of this preliminary study is the identification of the four roles played by managers on agile teams: mentor, coordinator, negotiator, and process adapter. As a mentor, the manager guides and supports the team in agile practice; the coordinator facilitates and coordinates the teams functioning; the negotiator takes care of the budget and customer requirements; and as a process adapter, the manager customizes agile and also implements agile-waterfall hybrids. The results of this study highlight the need for in-depth research into the different management roles and functioning of the agile team and manager. Additionally, this study will help guide new and existing 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.

CCS Concepts

• Software and its engineering → Software creation and management → Software development process management → Software development methods → Agile software development
• Social and professional topics → Professional topics → Management of computing and information systems → Project and people management → Project management techniques.

Keywords

Agile project management; agile software development; manager; project management; self-organizing team; grounded theory.

1. INTRODUCTION

In traditional software development methods such as the Waterfall model, the role of the manager is perceived to be central to the project involving different facets such as leadership, team building, motivation, communication, influencing, decision

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ISEC '17, February 05-07, 2017, Jaipur, India

© 2017 ACM. ISBN 978-1-4503-4856-0/17/02...\$15.00

DOI: <http://dx.doi.org/10.1145/3021460.3021465>

making, planning, and coaching [1],[2]. However, towards the late 1990's, a different methodology known as agile software development (ASD) was gradually establishing itself in the software engineering domain. ASD's emphasis was on the concept of self-organizing teams [3]. Self-organizing teams have been described as teams displaying significant autonomy in taking decisions, managing workloads and allocating work amongst themselves [3],[4],[5],[6],[7],[8].

A more radical change was that in ASD methods such as Scrum and eXtreme Programming (XP), the job title and role of the manager simply did not exist [10]. Different agile methods introduced a raft of new roles such as the *scrum master* and *product owner* [9],[12]. A scrum master is responsible for facilitating team functioning and removal of impediments, and the product owner is the customer-representative and keeps the team aligned to the customer's product vision [10]. XP has a different set of roles altogether such as the *coach* – someone who is responsible for process and team guidance [10].

While existing ASD literature elucidates that a certain proportion of the erstwhile manager's responsibilities are carried out by the new roles such as scrum master, product manager and coach, it is still not clear how this is implemented in practice [10],[11],[12]. In particular, it is unclear as to who, if anyone, is responsible for carrying out the various aspects of the traditional manager's role and for implementing project management practices on agile projects.

At the same time, there is strong evidence that in practice the role and job title of the manager are still in existence even in organizations practicing ASD [29],[30]. This suggests that there is a gap between what is recommended by literature and what is implemented in practice. There is a paucity of guidance backed by empirical evidence on what exactly is the role of the managers in ASD. Thus, the main aim of our study was to answer the research question: *What is the role of the manager on agile projects?* We use the term *manager* or *agile manager* here interchangeably to refer to people in a variety of management roles including scrum masters, managers, product owners, project managers, etc. The term "agile manager" has previously been used in practitioner literature on APM in similar contexts [24].

To investigate this, we conducted a Grounded Theory (GT) study involving 20 participants from 18 different organizations. GT is particularly suited to study the human aspects of software engineering as it is flexible enough to accommodate both qualitative and quantitative data and focuses on discovering the main concerns of the human participants.

Our key finding was that the manager plays different roles such as the mentor, coordinator, negotiator, and process adapter on agile projects. As a mentor, the manager guides and supports the team in agile practice; as a coordinator, the manager facilitates and coordinates the teams functioning; as a negotiator, the manager takes care of the budget and customer requirements; and as a process adapter, the manager implements agile-waterfall hybrids and customizes agile practices.

The rest of the paper is structured as follows: Section 2 presents a background on traditional and agile software project management, and the role of the manager in both contexts as per the literature. Section 3 describes the research methodology, which includes an exposition of the steps used to analyze the data from the interviews. The findings of the data analysis are discussed in Section 4. Section 5 presents discussion, Section 6 presents the limitations, and Section 7 presents the conclusion which includes directions for future research.

2. BACKGROUND

2.1 Traditional Software Project Management

For many decades the leading traditional software development methodology followed in the software industry has been the Waterfall model [13],[14]. The main characteristics of the Waterfall model are a sequential arrangement of different software development steps, extensive upfront planning, upfront requirements gathering, emphasis on detailed documentation, and a focus on the process. Over the years, one of the key limitations of the Waterfall model was identified as its poor adaptability to changes in the project environment and the problems caused by incomplete requirements [15],[16].

By the early 1990's, the solution advocated to overcome the shortfalls of the Waterfall model was to adopt an iterative and evolutionary approach to software development [17]. Eventually by the late 1990's this resulted in the introduction of agile as a full-fledged software development methodology. In the following section we have used the terms agile software development (ASD) to denote the software development part of agile and agile project management (APM) to denote the project management aspects of agile methods.

2.2 Agile Project Management

In the last decade, the adoption of ASD has been extremely rapid in the software industry worldwide [18]. ASD is an incremental and iterative development methodology, with an emphasis on people and on rapid response to change. ASD is more like a broad umbrella which comprises different software development methods [4],[9],[10],[19],[20]. The two most commonly used ASD methods are Scrum and eXtreme Programming (XP), and each of them has a unique set of roles [21]. Scrum is focused on software project management with project management artifacts such as the product backlog, daily scrum, sprint review meeting, while XP mostly focuses on development activities [22],[23].

In practitioner literature agile project management (APM) has been defined as, *"the work of energizing, empowering, and enabling project teams to rapidly and reliably deliver business value by engaging customers and continuously learning and adapting to their changing needs and environments"* [24]. The concept of the self-organizing team differentiates APM from traditional software project management [3],[8],[11]. Hoda and Murugesan [8] have identified different levels of project management challenges in APM which arose due to the unique

nature of self-organizing teams. These include project level, team level, individual level and task level challenges.

2.3 Who is the Project Manager?

The earliest definition of the project manager – a formal job title – encountered in the literature comes from 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* [25].

The Project Management Institute (PMI), which is a leading professional body, through their project management book of knowledge (PMBOK) has defined the project manager *"as the person assigned by the performing organization to lead the team that is responsible for achieving the project objectives"* [1]. The project manager is envisioned as the link between the project team and the stakeholders. As per published literature, the project manager is expected to possess a variety of hard and soft skills such as leadership, team building, motivation, communication, influencing, decision making, political and cultural awareness, negotiation, trust building, conflict management, planning, effective supervision, budgeting, and coaching [1],[2].

In the software industry, the importance of the project manager's role can be gauged from a study of IT project failures, where the presence of skilled project manager is identified as a key driver for project success [26].

In traditional software development methodologies such as the Waterfall model [13],[14],[18],[27], the project manager's role is crystallized within a well-defined hierarchy in the project team and it is perceived as being a crucial pivot for project success. However, within the Waterfall approach, this same well-defined hierarchy results in a command and control response within the project team [11]. Software development nowadays is characterized by rapidly changing customer requirements [28],[31]. In this context, a rigid, top-down management approach can lead to inflexibility in responding to changes and can cause problems in meeting the project objectives [28].

In terms of the role of the project manager, Karlsen and Gottschalk mapped out six managerial roles adopted by the project manager on IT outsourcing projects [44]. These were: the leader, liaison, resource allocator, spokesperson, entrepreneur and monitor. The leader was responsible for a wide spectrum of activities including team coordination and motivation. The resource allocator ensured optimal distribution of resources, while the spokesperson role involved the project manager interacting with internal stakeholders. The liaison role is mostly identical to the spokesperson except as the liaison the project manager interacts with the environment outside the organization. The entrepreneur spots and capitalizes on nascent business opportunities and the monitor keeps a watch on the external environment. However, the study by Karlsen and Gottschalk does not focus on an agile environment.

2.4 The Agile Project Manager

As mentioned in the previous sections, different ASD methods, such as Scrum and XP, do not include the role of the manager [12]. Scrum introduced two new roles, namely that of the *product owner* and the *scrum master* [9],[12]. The product owner is essentially the customer representative and is responsible for providing guidance to the team with regards to the customer's requirements and prioritizing the product features. The scrum master is primarily tasked with facilitating the team's functioning and the removal of impediments. In other words, the product

owner is the link with the customer, whereas the scrum master is the internal facilitator [11]. XP innovated different roles such as the coach, consultant, tracker, programmer, customer, tester, and the big boss [10]. Out of these roles the role of the “coach”, shares some characteristics with the project manager’s role as laid out in PMBOK [1],[10]. A coach is responsible for the process, guiding the team and learning from other XP teams [10]. These characteristics of the coach role map with the coaching and influencing characteristics specified for the traditional project manager [1].

However, in practice, the role and job title of the project manager are still in existence on a considerable number of projects which have adopted ASD [29],[30]. This is attested to by the fact that in a leading industry survey on ASD, for the years 2014 to 2015, nearly 24% of the respondents have been project managers [29],[30]. Additionally, our own industry survey uncovered that nearly 67% of the respondents had a project manager on their ASD project [45]. Our study also discovered that there was a strong correlation between the size of the team and the presence of the project manager. In team sizes of between 5-10 and over 25 members there was a higher possibility of the project manager being present. Additionally a majority of the distributed teams had a project manager.

The role of the manager in APM has remained a relatively less explored topic in agile research literature. A recent study by Siddique and Hussein [43] addressed the aspect of conflict within agile teams from a project manager’s perspective. Their study identified the causes and the consequences of conflict within agile teams. Some of the causes identified included a lack of experience of the project manager, budgetary issues and ego conflicts within teams. The consequences of such conflicts could be a drop in productivity, lowering of motivation and poor decision making. Hence, it is important to understand the role of the manager in APM.

3. RESEARCH DESIGN

The research methodology chosen to conduct the study was Grounded Theory (GT). It is a suitable method for studying human and social aspects of software engineering [3], [22],[31],[32]. Grounded Theory (GT) has the flexibility to accommodate both qualitative and quantitative data. It 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 [33]”.

There are two popular variants of the GT method known as the Glasserian method and the Straussian method respectively [34], [35]. For purposes of this research, the Glasserian or the classic GT method has been adopted. The classic GT method has been adopted primarily due to the wide array of resources available and because Glaser’s approach has been widely used in research on software engineering [3, 8, 31, 32].

The key feature of GT is that it is a general methodology which is applicable across different disciplines. This wide appeal is mainly due to the core principle of GT, which is to avoid having a preconceived hypothesis before commencing the research. In GT, the researcher is expected to uncover the main concerns of the participants and thus uncover the problem itself. Thus, it is used in disciplines as diverse as management [36],[37], computer science

P#	TX	Role	Domain	AM	TS	CN
P1	5-10	Dev	Banking, e-commerce	S	10-15	US
P2	5-10	PM	Local Government	S	10-15	NZ
P3	5-10	PM /SM	Telecommunications	S	5-10	NZ
P4	10-20	PM	Local Government	K	>25	NZ
P5	10-20	ProM	Insurance	K	0-5	NZ
P6	10-20	SPM	Banking	K	>25	NZ
P7	10-20	PM	Insurance	SA	5-10	NZ
P8	10-20	C	Telecommunications	S	10-15	NZ
P9	10-20	SrPM	Software	S	15-20	IN
P10	10-20	SPO	Telecommunications	S	5-10	NZ
P11	10-20	ProM	Banking	S	15-20	NZ
P12	0-5	AC	Aviation	M	>25	NZ
P13	11-20	SSD	Accounting	SK	0-5	NZ
P14	11-20	SM	Telecommunications	M	6-10	NZ
P15	11-20	Dev	Finance	K	0-5	NZ
P16	11-20	PM	Telecommunications	S	21-25	NZ
P17	6-10	SM	Utilities	S	6-10	AUS
P18	11-20	SM	Finance	S	6-10	NZ
P19	6-10	SM	Entertainment	S	6-10	NZ
P20	6-10	TC	Telecommunications	S	11-15	IN

Table 1. Demographic of Participants

(P#: Participant number; TX: Total Experience in years; Dev: Developer; PM: Project Manager; SM: Scrum Master; ProM: Programme Manager; SPM: Software Product Manager; SrPM: Senior Project Manager; SPO: Scrum Product Owner; AC: Agile Consultant; C: Consultant; SSD: Senior software developer; TC: Technology Consultant; AM: Agile Method; S: Scrum; SA: Scrum Adapted; K: Kanban; SK: Scrum and Kanban mix; M: Mixed; TS: Team Size; CN: Country; IN: India; NZ: New Zealand; AUS: Australia; and US: United States of America)

and engineering [31],[38], software engineering[3], information science (IS) [39], health sciences [40] and sociology [41].The implementation of GT in our research has been explained with examples, in the following sub-sections.

3.1 Data Collection

Data was collected from 20 agile practitioners from the New Zealand, Australia, USA, and India. The maximum number of participants (N=16) were from New Zealand, while there were two from India, one participant each from the USA and Australia. Table 1 gives 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. P1, P2, etc. In terms of team sizes, there was a wide variance with team sizes going from 5 members to well over 24 members.

The participants typically belonged to the managerial level in their organization and held a variety of job titles such as project manager, scrum master, programme manager, software product manager, senior project manager, and one of the participants was a team member i.e. a developer.

The participants were contacted via LinkedIn, which is a widely used networking site for professionals. The criteria for selection included a minimum of two years of experience in agile practice as it was deemed a reasonable timeframe to reflect on relevant experiences. The participant’s demographic information was

collected via a pre-interview questionnaire, which the participants completed prior to the research interview.

The interviews were semi-structured and lasted approximately an hour. Most of the interviews were conducted face to face, except for four, which were conducted over Skype. The interview questions were open-ended and were designed to elicit comprehensive information from the participants. These were tailored depending on the participant's experience and job title. As an example some of the interview questions are given below. The questions have been divided into general and project categories.

General questions

1. Please tell me briefly about your professional background and your current role in this organization.
2. How did you get introduced to Agile software development and how long have you been practicing it?

Project specific questions

1. What was your role in the team?
2. What project management practices are utilized and who utilizes them? Which of the project management practices utilized are the most effective?
3. What are the major challenges you have faced while working in the agile project? How did you overcome those challenges?
4. Were the challenges resolved? If yes, what strategies or techniques were useful in achieving resolution?
5. Can you explain how decisions were made in your agile team?

3.2 Data Analysis

The key techniques used to identify patterns within the interview data were the GT procedures of open coding and the constant comparison method. The software used for data analysis was QSR nVivo v.10, which is a popular software tool for qualitative data analysis. The analysis was performed by the first author in consultation with the other two authors. The codes and concepts resulting from the analysis were discussed amongst all authors and any conflicts were resolved through discussion and expert advice.

The first stage of analysis involved sifting through the raw data (interview transcripts) and extracting snippets of data from the transcript. 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: *"So whatever the methodology, the purpose is still that same to me, you know, the object is to make sure the context of the project is understood and there within the context of that project your job is to sweep away the obstacles."* – P2, Project Manager.

Key point: Sweeps away obstacles

Code: Removing obstacles

Concept: Coordinator

Further analysis on the codes was done using GT's "constant comparison" method [33],[34],[35]. This involved comparing the codes within the same interview and those across all interviews and then grouping them together. In the above example, another similar code identified was *"resolving conflicts within team"*.

These codes were then grouped together to produce a higher level of abstraction called a *concept*. As has been shown above, the

concept in this case was the *"coordinator"*, which comprehensively encapsulated the code grouping. In other words, part of the agile manager's role was to act as a coordinator, removing obstacles and resolving conflicts within the team. The same process of constant comparison was repeated for each of the concepts and this led to the emergence of other concepts, i.e. the roles of the mentor, negotiator, and process adapter which formed the next higher level category of the *"roles of the agile manager"*.

Category: Roles of the agile manager

This process has been graphically demonstrated in Fig.1, which also shows all the major codes and concepts that emerged.

4. FINDINGS

In this section, we present on of the key categories that emerged, the roles of the agile manager, along with the underlying concepts as identified in this study. The category roles of agile manager includes: the mentor, coordinator, negotiator, and process adapter. We also include sample quotes drawn from the interviews.

4.1 Mentor

Participants in our study identified the characteristics of a mentor as follows: educates and trains the team in agile practices (P7, P10, P11, P13, and P16); educates stakeholders on agile practices (P3-P6, P11, and P14); makes the team aware of the larger organizational context of the project (P11, P17); ensures that the team adheres to agile practices (P2, P4, P5, P7, and P17); encourages the team to be self-organizing (P2, P3, P5-P8, P11, P13, and P17); and acts as the coach (P8, P10, P14). Each of these aspects of the mentor role are described below.

Training team members in agile practices

The manager on agile teams ensured that the team members were well versed in agile practices and roles (such as the product owner) by arranging or directly providing training and providing ongoing support to the team members (P7, P11, P13, and P16). P11, the programme manager on a banking project arranged for a large number of team members to be trained as product owners. Similarly, P16, the project manager in a utilities project, facilitated induction programs for new team members.

"I did send 11 people off to be trained as product owners, and formally certified product owners so that they understood once again from an external perspective, this is what you should be, this is what your role is, here's how you should be doing your job effectively." – P11, Programme Manager, New Zealand.

The manager on an insurance project (P7) provided basic agile training to an inexperienced test analyst. This was also seen in the case of P13, where the senior software developer would review the code of junior developers and help them in achieving quality.

"So if the code was not good quality enough, we look at the branch and help them. And we did code review, but in both sense. So we would, the senior, would code review the code of the juniors, to tell, to help them with their code." – P13, Senior Software Developer, New Zealand.

Educating and updating stakeholders

Another key aspect of the mentor role was to raise awareness and interest among stakeholders about the project and also educate them on agile practices (P3, P4, P5, P6, P11, and P14). This could involve making the client presentations entertaining by structuring them as a game show (P11), giving the client representatives a good understanding of the agile way of working (P3, P14),

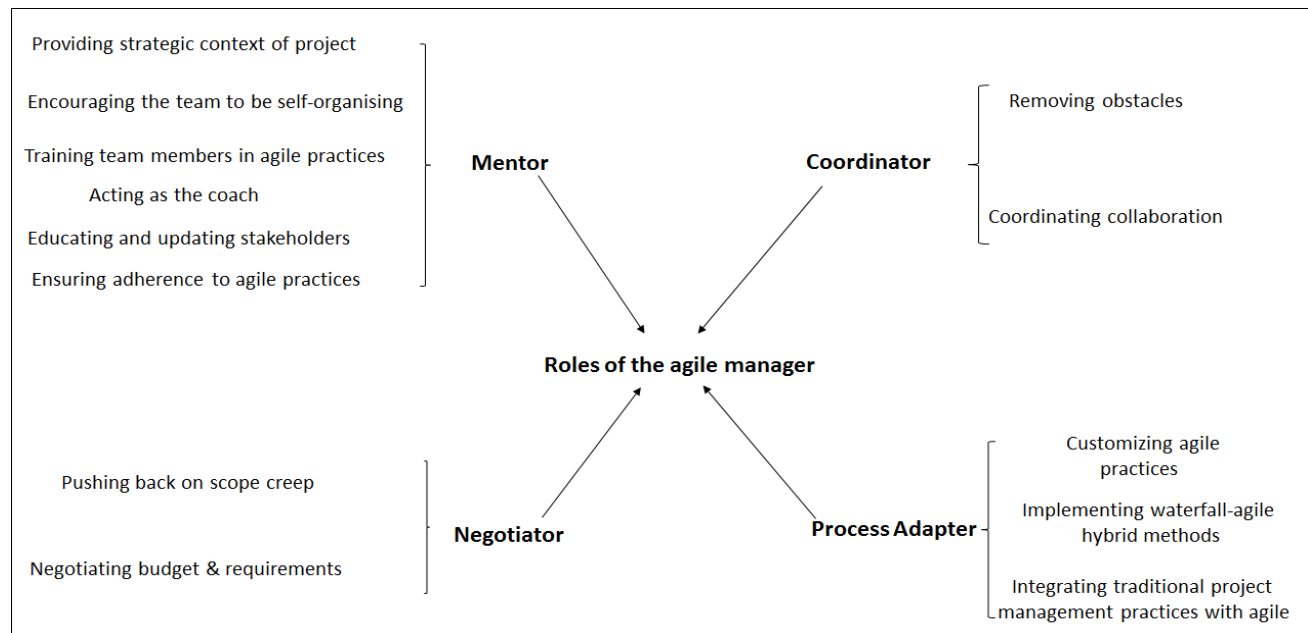


Figure 1. The emergence of the category “Roles of the Agile Manager” from underlying concepts and codes.

presenting regular product demonstrations to the customer and providing encyclopedic information packs (P5).

“So I was focusing on building the capability and helping [agile] be structurally run in the transition and helping the senior management get into this model of organization, getting the business themes and products and things like that to start thinking Agile.” – P14, Scrum Master, New Zealand.

“In terms of what I presented to the stakeholders, we kind of had set packs. We would have a project, what did we call it, it had a name, it was like a PowerPoint presentation. And it was the Bible for the project, it had everything in it. And I would just pull out the salient pieces when I was presenting to the stakeholders, but it would cover everything.” – P5, Programme Manager, New Zealand.

Ensuring Adherence to Agile practices

The project manager ensured that the team adhered to standard agile practices by personally facilitating daily standups (P2, P4, P7, and P17); by making sure product demonstrations were carried out (P2, P5); and homing in on the causes of deviation from estimates (P7).

The key driver to personally facilitate the daily standup across four participants (P2, P5, P7, and P17) was to ensure smooth flow of communication across large teams. In the case of customer demonstration, participants (P2, P5) identified that teams tended to discard the demonstrations towards the end of the sprint or during the year end.

“...the problem you get with, the bigger the team you get the lack of cross-communications, right. And people, even if they’re literally sitting in the room this size, these people don’t know what those people down the other end are doing, right. So this is why the daily stand-ups were implemented by myself.” – P2, Project Manager, New Zealand.

“So yeah so I will be looking, I will be leaving them to follow the, the defined, kind of, methods and practices and then when they would deviate look for the root causes of those deviations and

work with either the individuals or the team to address them to, you know, help correct.” – P7, Project Manager, New Zealand.

Providing strategic context of project

One of the key functions of the mentor role was to make the team aware of the user perspective and the larger strategic goals of the business (P11, P17). This was done by involving the stakeholders in the meetings and the agile manager acting as the bridge between the team and the business (P17). In the case of P11, this had the positive effect of enabling the team to identify new opportunities and made the team passionate about customer satisfaction.

“I’ll be there to assist [the] team in bringing business people together. If there were any external people they [the team] needed to talk to, I would be there to bridge those conversations.” – P17, Scrum Master, Australia.

“It also meant that the team identified opportunities for things going forwards that the business wouldn’t have even been aware of, or even thought of because they had this awareness.” – P11, Programme Manager, New Zealand.

Encouraging the team to be self-organizing

The mentor role involved encouraging the team to become self-organizing by taking ownership of risks and issues by using a task board (P11); giving the team the leeway to postpone work (P3, P7); involving team members in interactions with vendors (P2); letting the team take charge of documentation (P6); encouraging the team to interact with other stakeholders (P6, P17); encouraging team members to resolve conflicts (P5); involving the team in the planning process (P3, P8); making team aware of a different way of working (P13); and collocating the team (P17).

One of the techniques that the programme manager (P11) utilized to encourage team members to become more self-organizing was to use a ROAM (Resolved, Owned, Accepted, and Mitigated) board. The team members were encouraged to take ownership of different issues and activities listed on the ROAM board. Another method of encouraging the teams to become self-organizing was

to enable flexibility in moving items to the next sprint (P7, P11). This also involved the agile manager being perceptive enough to set realistic objectives for the team with the final deadline in view. The team was also given the freedom to choose what actually needed to be done to accomplish the project goals and additionally the method of documenting the implementation was completely left to the teams discretion (P6). The agile manager also involved the team in the planning process and made sure of the team participation in activities such as the scrum planning meeting and planning poker (P3, P8).

“So we’d sit down in our scrum planning meeting and we’d run planning poker. So we’d do, everyone, everyone involved in the sprint would participate in that.” – P3, Project Manager, New Zealand.

The agile manager also encouraged the team to have greater interaction with the vendor and stakeholders. This enabled quicker communication of relevant feedback to the vendor. Where the stakeholder was concerned, direct communication brought in clarity regarding the stakeholder requirements.

“So I had to know when to flex on things to maintain a realistic, kind of, set of goals and when to maintain a particular deadline to give them some focus on something to, to meet.” – P7, Project Manager, New Zealand.

The agile manager (P13) also acted to encourage self-organization by demonstrating a different way of doing things.

“But it’s helping them with the way they organize, the way they self-organize, and trying to show them some new things, that they may not have seen before, because some of them have worked the same [traditional] way for, like, 10 years, they don’t know [how to self-organize].” – P13, Senior Software Developer, New Zealand.

Acting as the coach

Some participants (P8, P10, and P14) identified their role as being a coach to the team. This involved providing guidance to the team in finishing the task while letting the team keep ownership of the task. Guidance to the team usually involved assisting them with fine tuning the requirements. The coach aspect of the mentor role also involved up-skilling the team members by assigning small pieces of work.

“I help them do the, clarify their requirements, break down their requirements, break down larger features into smaller ones, help them prioritize, help them advise dependencies between stories, features, consequences of changing priorities.” – P10, Scrum Product Owner, New Zealand.

One of the key approaches of a successful coaching role identified by P14 was to let the team experiment and learn from failure.

4.2 Coordinator

The coordinator in the context of our study is defined as someone who facilitates and coordinates the team’s functioning by: resolving conflicts and clearing obstacles, taking care of routine process administration, displaying a hands on approach to problem solving, and strengthening interpersonal relationships.

Removing Obstacles

Half of the participants (P2, P4-P8, P17, P19, and P20) identified that the coordinator role involved the agile manager acting as the “minesweeper” i.e. resolving intra-team conflicts and removing obstacles.

Coordinating conflict resolution: The strongest evidence came in favor of the agile manager being a conflict resolver and usually acting in scenarios where the team was unable to resolve a situation. The agile manager could act as the conflict resolver by reassigning problematic team member, having one on one discussions with the concerned team members and by encouraging the team to mutually resolve the issue.

“I would co-ordinate a lot of, you know, if you had two technical guys who couldn’t agree on an approach they’d pitch to me and then I’d make a call.” – P5, Programme Manager, New Zealand

P8 cited a scenario where there was tension in the team due to cultural differences. A talented Russian developer had recently joined the team but his attitude towards the female team members, his demanding working style and lack of empathy towards slower team members, eventually led to conflicts within the team. P8 had to step in and reassign the developer. The developer was put to work on a core component of the project where interaction with the team was minimum. Also, the agile manager made the developer aware of cultural nuances by having a one to one chat with him.

“He believed that women were, you know, not as capable as men and the balance in the team got quite disruptive. And they came and talked to me and said, look, you know, we’re really struggling with this guy. So yeah, I moved him.” – P8, Consultant, New Zealand.

A similar instance was observed in an insurance project, where P7, the project manager, had to resolve a conflict between the business analyst and the product owner, where the conflict was being accentuated by the blunt approach of the business analyst.

Another approach to conflict resolution was to encourage the team members to mutually resolve the conflict rather than seek intervention of the project manager. This was observed in the case of an insurance sector project, where P5 as the programme manager encouraged the team to mutually resolve the issue.

“So absolutely come to me if you’ve gone to that individual and you haven’t been able to resolve it, but it’s not the traditional I’m going to lob all my issues at the PM and they’re going to sort it out for me.” – P5, Programme Manager, New Zealand.

Coordinating clearing of obstacles: Another aspect of this role included the agile manager acting to clear obstacles to the team performance (P4, P5, P13, P17, P19, and P20).

“So my way of working primarily is just to let them get on with it, they come to me with problems and I get rid of the road blocks, and I’m there as a sounding board for ideas.” – P5, Programme Manager, New Zealand.

P4 used the Kanban board to highlight bottlenecks early on and this helped to resolve the bottlenecks as well.

“... and for bottlenecks and those type of things, the Kanban board just highlights it, so then I could be, I could go over and try to deal with that bottleneck. I guess that’s one of the roles that’s a little bit different than standard, say, Waterfall. The bottlenecks are highlighted pretty early on to me, and then my kind of role is to help the team succeed.” – P4, Project Manager, New Zealand.

In the project reported by P13, the project manager worked to ensure minimum interruption by customers to the project team. The project manager did this by ensuring that the customers were aware that interruptions slowed down the pace of delivery.

Another obstacle identified by P17 was the tendency of the team to rely excessively on emails and wait for the vendor to respond to their queries. This led to delays in delivering the product. In this scenario the scrum master encouraged the team to ring the vendor and get immediate clarifications.

“Let’s just get into a room, ring that person and get an answer, instead of waiting. There is a tendency for certain IT people to not want to communicate. So they would use email as a communication tool to people outside of team.” – P17, Scrum Master, Australia.

In a software project in the entertainment sector, the scrum master (P19) identified the fact that the testers were idle as the team was working on many tasks in parallel. The bottleneck was removed by limiting the number of parallel tasks the team could work upon and prioritizing closure of outstanding items.

The project manager on a telecommunications project removed obstacles by pointing the team in the right direction (P20).

“The project manager always had some solution, as he had worked a long time in the product. Any problem I tell him, [he would say] okay, maybe you should go and contact that person or you should do this. The next step guidance, that helps.” – P20, Technology Consultant, India.

Coordinating collaboration

The coordinator also helped make the team more efficient and productive by encouraging involvement of specialists, coordinating intra and inter team collaboration, and building relationships within the team.

Coordinating customer collaboration: The project manager increased the contact between the team and stakeholders by different methods (P5, P8, and P11). One way was to encourage the team to establish direct contact with the stakeholders to clarify requirements. Another method was to involve the stakeholders in product demos to enable a better understanding of the system being built. A third method was to either take team members to client meetings or invite a client representative to the site.

“I would quite often take one of the team out to the meetings at Company X with the product owner to talk around what we could and couldn’t do.” – P8, Consultant, New Zealand.

“So what I did was I changed that, and what I do is I actually brought one of the support team to Auckland. And they spent one day sitting with the team. So they’d be sitting with the developer, and the developer would be showing them these are the top 20 things that you’re probably going to run into, encounter in terms of issues.” – P11, Programme Manager, New Zealand.

Coordinating specialist collaboration: Four participants (P3, P5, P8, and P17) identified that they encouraged the involvement of specialists to facilitate the teams functioning. The active involvement of specialists, such as the solutions and security architects, had the positive effect of acting as a sounding board for the team’s ideas and also allowed the team to focus on core deliverables while the specialist worked on certain project deliverables. P16 identified the involvement of specialists as being a crucial element to being able to deliver the project on time.

“We wouldn’t have got the project across in time without validation of the approach the team was taking. Added a lot of value to the non-functional aspect [compliance] of the project.” – P17, Scrum Master, Australia.

“And when we had story pointing [estimations] we wouldn’t just have our core team, we’d also invite a few extras. So I would always invite the solutions architect, or two of them. I’d always invite another experienced developer from another team to come in, to act as a sounding board and another comparison if one developer thought it was really complex and they could bandy ideas around.” – P5, Programme Manager, New Zealand.

Coordinating intra-team & inter-team collaboration: The coordinator was also responsible for facilitating intra-team collaboration (P4, P5) and inter-team collaboration (P2, P10 and P15).

Intra-team collaboration was accomplished by holding joint meetings to root out any issues or encouraging team members to take ownership of disputes within a team.

“So from a practical standpoint I got a lot in the early days of staff coming up to me to have a moan about something, and my response to them would be well what have you done about it. Because historically I would wade in and sort it out. Under Agile you’ve all had that kick off meeting at the front where you’ve all agreed to work together, so I’m holding you to that.” – P5, Programme Manager, New Zealand.

The agile manager also coordinated between different teams working on the same project (P2, P10 and P15). This could be between different teams using agile or between teams which used waterfall and agile.

“So the project team itself weren’t overly exposed to Agile and didn’t need to be, it was really me acting as an interface to a vendor using Agile Methodologies in order to deliver a component of the overall project.” – P2, Project Manager, New Zealand.

In a software project in the financial sector the scrum master played a key role in facilitating collaboration between teams from the platform and development arenas.

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

Coordinating better team relations: The agile manager strengthened the relations of team members with each other by: rotating team members in retrospectives (P5); keeping team composition stable (P5); cross fertilizing team members (P11); and, by team bonding activities (P11). Each of these activities brought their own benefits. For example, one of the key benefits identified of keeping the team composition stable was that the expertise in estimation was not diluted with every team reshuffle.

“And again, that’s the benefit of teams that don’t shuffle all the time. They get really good at that estimating. As soon as you start shuffling the team you have to pull them back, break down a couple of stories, recalibrate that benchmark and then start again. Yeah, but it worked very well for us.” – P5, Programme Manager, New Zealand.

In the case of rotating team members in terms of responsibilities for the retrospective, one of the benefits identified by P5, on an insurance project, was that it led to stronger team cohesion and a greater buy-in towards the retrospective. Another technique was to cross-fertilize the team by including team members from the client. This led to a greater level of understanding between the two sides.

4.3 Negotiator

A negotiator is defined as a manager who negotiates with stakeholders and vendors on behalf of the team and takes care of change management. The negotiation can be for project funding, issues with the project, scope of project and to ensure vendors are meeting their deliverables.

Negotiating budget and requirements

Six participants (P1, P2, P3, P7, P8, and P10) identified that one of the functions of the agile manager's role was to negotiate with stakeholders or their representative and with vendors. The agile manager negotiated with stakeholders which included senior management, vendors and product owners. The negotiations could be around requirements, budgets and minimizing interference into the software development process.

In a telecoms project, the project manager (P3) who also doubled up as the scrum master, was in charge of conducting negotiations with the clients. This entailed reaching a middle ground between the client requirements and the perspective of the team on the project deliverables.

"Yes, it was quite challenging 'cos there was nothing quite set in stone, so it was a matter of basically taking their point of view and also explaining our point of view and trying to marry those two up." – P3, Project Manager, New Zealand.

On an insurance project, the project manager (P7) negotiated with senior management for additional budget by presenting a strong case for the increment.

"So that was, I had to basically show using evidence why I was asking for more money and what would happen if you didn't do that and what we, how it stacked up against the benefits that we'd already planned and all that kind of stuff. So yeah, so it took a bit of work to convince them, but once they could see the, the facts that the, the benefits still stacked up, there were clear reasons why we'd come up with a new number and it was more realistic than the previous number, they were happy to go ahead. They weren't happy but they did it." – P7, Project Manager, New Zealand.

In two different scenarios of insurance and telecoms projects, the project managers (P10, P7) negotiated with the product owner about the workload in the product backlog and changes to the items in the backlog.

"I would have conversations with the product owner about what was in the product backlog and how much we were likely to get done, or how much we could feasibly get done within the current budget." – P7, Project Manager, New Zealand.

Additionally, the agile manager also made sure that the stakeholders did not interfere in the software development process (P8, P15) and conducted negotiations with vendors (P2). The agile manager, in this particular scenario the scrum master, could be quite firm with even senior management and remind them that the team had already committed to certain work (P15).

"And then of course if you often have issues when vendors aren't quite up to speed, you know they said they'd deliver this, say a function specification this week, where is it, why can't we have it, you know we got people ready to go, obviously need to be negotiating with following up on that type of stuff." – P2, Project Manager, New Zealand.

Pushing back on scope creep

The agile manager also acted as the gatekeeper in terms of change management and worked to prevent scope creep. This was seen in the case of four participants (P5, P6, P7, and P8) who had worked on telecoms and insurance projects.

"Yeah, but it would more be, side distractions. So, you know, stakeholders would start talking about lobbying extra things in or kicking off a new project. And they'd start muddying the water at the stand ups, they'd start talking about other pieces of work. So then I'd have a separate session and we'd talk about here's the streams, here's the work limits, and you've have members of the team there to talk about it as well." – P5, Programme Manager, New Zealand.

In the scenario of the stakeholders trying to introduce new items of work, the agile manager would arrange a separate meeting with them and explain the importance of adhering to the current work limits. If the client was persistent regarding the addition to the scope, the agile manager had to assert the fact that the additional task could be done only on the condition of an existing task being removed. The project manager also pushed back on last minute change requests which were initiated by the client.

"No, we did it because it was really obvious that there was an acute problem and that, you know, they rang me up and said, hey, we need to go live next week, and I had to say, sorry, bad luck. You've already committed to other work. We've got a fixed capacity. You need to plan better." – P8, Consultant, New Zealand.

4.4 Process Adapter

We define the process adapter as an agile manager who implements waterfall-agile hybrid methods (P3, P6), integrates traditional project management practices with agile practices (P2, P3, P5, P6, P10), and customizes agile practices to suit business context (P13-P15, P17, P18, and P19).

Customizing agile practices

A considerable number of participants were found to customize and integrate elements of different agile methods (P13-P16, P18, and P19).

The integrations could be introducing practices from XP and Kanban into Scrum. A good example would be the inclusion of practices such as the user story format for product backlog and continuous integration (P14). On an accounting project, the team used a mix of Scrum and Kanban. The Scrum practice of iterations was not used as the team was working simultaneously on multiple projects. The Kanban board was found more useful as a tracking tool. Similarly, P15 and P19 utilized a mix of Scrum and Kanban practices, particularly the Kanban board.

"...the full backlog was visible on the board, and it was more a matter of...making sure the things are ordered, and when we see that the work starts to empty in the 'in-progress' [column], to make sure that we have talked enough with the customer that the order is correct, that we know more or less what we have to do." – P13, Senior Software Developer, New Zealand.

An adaptation to the standard demo introduced by the scrum master (P17) was that of a "snap demo". This was an informal product demonstration to the stakeholder, held every week, where the stakeholders would gather behind a developer's desk and see the product.

Time difference played an important part in the customization of agile practices. This was seen in the case of P18, the scrum master

on a finance project. As the key stakeholders were in the USA and the team was based in New Zealand, holding the conventional review was not found to be feasible. Instead the scrum master introduced two separate reviews, one with the client and another with the team.

Implementing waterfall-agile hybrids

The most typical aspect of a process adapter was to adapt the Scrum practices to fit within a more traditional waterfall wrapper. The scenario typically elucidated by participants was agile practice at the team level and traditional project management applied at the senior management level. This necessitated the development of what one participant termed as “*Waterscrum*” i.e. a hybrid of agile and waterfall (P6). The reason for the waterscrum hybrid was the lingering existence of waterfall practices as the team was transitioning from Waterfall to Scrum. Another workaround adopted was to have a dedicated team which was tasked with integrating traditional project management with agile practices.

“So there was a core team of us that looked at our traditional project practices, and then integrated it with Agile and then rolled it all out with templates and everything else and ran surveys.” – P5, Programme Manager, New Zealand.

Integrating traditional project management practices with agile

An interesting combination of traditional and agile method was to run scrum sprints within traditional project phases. In this instance, the clients plan had split the project into three phases. The manager slotted in sprints into each phase. A related approach was to explain to the clients that the sprints were analogous to traditional project milestones i.e. the completion of each sprint was essentially the achievement of a milestone according to the client’s plan.

“So each phase had its own sprint so this is where they, they followed very much on the Agile methodology, separate project plan from them, from the vendor, okay.” – P2, Project Manager, New Zealand.

5. DISCUSSION

5.1 Comparison to Related Works

Although agile project management is still a relatively less explored field, recent studies have shed light on some key aspects such as self-organizing teams [31],[42], project management challenges in agile [8] and the project manager’s view on conflict in agile projects [43].

For example, the self-organizing team is a relatively well studied area and there are studies which describe the roles that make an agile team self-organizing [42]. In Hoda et al.’s study on self-organizing teams, five new roles played by different members of the agile team were uncovered: mentor, coordinator, translator, champion, promoter, and terminator [42]. The mentor is responsible for guiding the team in agile practices; the coordinator forms the link between the customer and the team; the translator brings clarity to communications between the stakeholders and team; the champion secures buy in from senior management to agile; the promoter promotes agile to customers; and the terminator removes team members unable to thrive in a self-organizing context. Out of these, the coordinator and the mentor role are well represented in our study. While the characteristics of the mentor role uncovered in this study are identical to those put forth by Hoda et al. [42], we have expanded the definition of the

coordinator role based on our findings. Thus, our coordinator role includes activities such as resolving conflicts and clearing obstacles, taking care of routine process administration, displaying a hands-on approach to problem solving, and strengthening interpersonal relationships. Due to our small sample size, we could not find additional information regarding the other roles identified by Hoda et al. [42]. Additionally, our roles focus exclusively on the manager’s perspective as opposed to a team’s perspective [42] on project management in a self-organizing context.

Hoda and Murugesan’s study on project management challenges in agile identified eight challenges distributed across the multiple levels [8]. These levels were the project, team, individual and task levels. One of the project level challenges identified was that of delayed and changing requirements. Urgent customer requests frequently led to teams abandoning the sprint. The “negotiator” role and its emphasis on preventing scope creep can be a possible strategy when the team or the manager is confronted with intrusive customer requests.

Another project level challenge in the same study [8] was that of securing senior management support for agile methods. One of the strategies identified by Hoda and Murugesan is to present tangible benefits of agile adoption to the management in a bid to seek their support. An additional strategy that we would recommend is to encourage senior management to participate in team meetings and demonstrations from time to time. The “coordinating collaboration” code from the coordinator role in our study provides example of practical application of the strategy.

One of the causes of conflict on agile teams has been identified as the lack of stakeholder knowledge about agile methodologies [43]. A good strategy to resolve this conflict can be seen in the code “*educating and updating stakeholders*” of the mentor role. This involves using innovative methods of making the stakeholder aware of agile methods such as structuring the product demonstrations in the form of entertaining gameshows and providing comprehensive information packs.

Another cause of conflict identified in the study by Siddique and Hussein was shown to be the overlap of roles and responsibilities e.g. between the project manager and product owner [43]. Siddique and Hussein recommend that the various roles be clearly delineated to avoid confusion and conflict. However, from our study we have seen that in an APM scenario there exists a certain amount of overlap of responsibilities. Hence, we have used the term “agile manager” to encompass a range of job titles instead of the limiting it to the “project manager”. The different roles of the agile manager can be performed by one or many individuals irrespective of the job title.

5.2 Implications for Practice

The roles outlined in this study can be played by either a single person (i.e. the agile manager) or by different people in the team. The roles adopted will depend on the situation and the context of the project. The agile manager can act as the mentor, coordinator, negotiator or process adapter on different occasions or can play multiple roles on the same day.

In a scenario where the team needs to be given a good grounding in agile, the agile manager can use the strategy of training team members. Another facet of the mentor role i.e. educating and making stakeholders aware of agile, can be a useful tactic to drive stakeholder engagement in agile projects.

The agile manager can help the team become self-organizing by putting trust in the team's ability to take decisions and resolve conflicts. In terms of conflict resolution, the coordinator role provides useful tips and strategies such as removing obstacles to team performance. The agile manager can facilitate intra and inter team collaboration by holding joint meetings and making sure teams are accountable for their work commitments.

As the negotiator, the agile manager can strike a workable compromise between the team's viewpoint and that of the stakeholders. They will on occasions also need to be assertive in order to shield the team from scope creep and external interference.

All the roles call for the agile manager to have a high level of soft skills such as team building, motivation, communication, influencing, and decision making.

6. LIMITATIONS

One limitation of the study is that not all agile methodologies have been covered. In this study the majority of participants implemented Scrum and Kanban. Other agile methods such as eXtreme programming, FDD and BDD are not represented in the study. This is partially explained by the fact that Scrum and Kanban form the most popular methods used in the software industry [29]. Also, as this study focuses on APM, a majority of our participants are those in managerial roles. We will attempt to incorporate a wider variety of roles in the future rounds of data collection.

Additionally, the results of any GT study are specific to the particular contexts studied and cannot be completely generalized. The findings of the study emerged from the interview data and hence are particularly relevant to the context of the participants. An important point to note is that for the findings of a GT study to be relevant, they should be able to adapt to and fit different contexts [33]. The different roles of the agile manager can be modified and adapted to suit different contexts in the software engineering field and possibly in other domains.

7. CONCLUSION

In this paper we have presented the findings of a Grounded Theory analysis of 20 semi-structured interviews as a part of a large-scale ongoing research. The study was carried out with the motivation to uncover the role of the manager in ASD projects. We set out to answer the question "What is the role of the manager on agile software development projects?"

The key finding of this study is the identification of the four roles played by the manager on agile teams as: mentor, coordinator, negotiator, and process adapter. As a mentor, the manager guides and supports the team in agile practice; the coordinator facilitates and coordinates the teams functioning; the negotiator takes care of the budget and customer requirements; and as a process adapter, the manager customizes agile and also implements agile-waterfall hybrids. The most common role reported was that of the "mentor", with nearly fourteen participants. The least common role was the "negotiator" with eight participants reporting the role.

Our continuing empirical study on agile project management looks to answer more questions, such as: What is the reason for the project manager's continued existence on ASD projects despite practitioner literature doing away with the role? What are the project management aspects of agile practices? Future empirical studies can answer these and other questions to present a better understanding of the roles of the manager in agile projects for theory and practice. Additionally, this study will help guide

new and existing 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.

8. ACKNOWLEDGMENTS

We would like to express our gratitude to all the agile practitioners who participated in our research. This study has been funded by generous contributions from the Shastri family.

9. REFERENCES

- [1] Project Management Institute. 2013. *A Guide to the Project Management Body of Knowledge*. Newton Square, PA.
- [2] Pettersen, N. 1991. What Do We Know About The Effective Project Manager? *Int. J. Proj. Manag.* 9, 2 (May.1991), 99-104. DOI= [http://dx.doi.org/10.1016/0263-7863\(91\)90068-7](http://dx.doi.org/10.1016/0263-7863(91)90068-7).
- [3] Hoda, R., Noble, J., and Marshall, S. 2013. Self-Organizing Roles on Agile Software Development Teams. *IEEE T. Software Eng.* 39, 3 (March. 2013), 422-444. DOI= <http://dx.doi.org/10.1109/TSE.2012.30>.
- [4] Cockburn, A., and Highsmith, J. 2001. Agile Software Development: The People Factor. *Computer.* 34, 11 (November. 2001), 131-133.
- [5] Martin, R. C. 2002. *Agile Software Development: Principles, Patterns, and Practices*. Pearson Education, NJ.
- [6] Sharp, H., and Robinson, H. 2008. Collaboration and Co-Ordination in Mature eXtreme Programming Teams. *Int. J. Human-Comput. S.* 66, 7 (July. 2008), 506-518. DOI= <http://dx.doi.org/10.1016/j.ijhcs.2007.10.004>.
- [7] Chow, T., and Cao, D.B. 2008. A Survey Study of Critical Success Factors in Agile Software Projects. *J. Syst. Software.* 81, 6 (June.2008), 961-971.
- [8] Hoda, R., and Murugesan, L. 2016. Multi-Level Agile Project Management Challenges: A Self-Organizing Team Perspective. *J. Syst. Software.* 117 (July. 2016), s245-257. DOI=<http://dx.doi.org/10.1016/j.jss.2016.02.049>.
- [9] Schwaber, K., and Beedle, M. 2002. *Agile Software Development with Scrum*. Prentice-Hall, Upper Saddle River, NJ.
- [10] Beck, K., and Andres, C. 2005. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston, MA.
- [11] Highsmith, J. A. 2004. *Agile Project Management: Creating Innovative Products*. Addison-Wesley Longman Publishing Co., CA.
- [12] Drury-Grogan, M. L., and O'dwyer, O. 2013. An Investigation of the Decision-Making Process in Agile Teams. *Int. J. Inf. Tech. Decis.* 12, 6 (November. 2013), 1097-1120. DOI= <http://dx.doi.org/10.1142/S0219622013400105>.
- [13] Royce, W.W. 1987. Managing the Development of Large Software Systems: Concepts and Techniques. In *Proceedings of the 9th International Conference on Software Engineering* (Monterey, California, USA, March 30-April 2). IEEE Computer Society Press, Los Alamitos, CA, USA, 328-338.
- [14] Benington, H.D. 1987. Production of large computer programs. In *Proceedings of the 9th International Conference on Software Engineering* (Monterey, California, USA, March 30-April 2, 1987). IEEE Computer Society Press, Los Alamitos, CA, USA, 299-310.

- [15] Boehm, B.W. 2006. "A View of 20th and 21st Century Software Engineering. In *Proceedings of the 28th International Conference on Software Engineering* (Shanghai, China, May 20-28, 2006). ACM, New York, NY, 12-29. DOI=<http://dx.doi.org/10.1145/1134285.1134288>.
- [16] Robey, D., Welke, R., and Turk, D. 2001. Traditional, iterative, and component-based development: A social analysis of software development paradigms. *Int. J. Inf. Tech. Manage.* 2, 1 (January. 2001), 53-70.
- [17] Rowen, R.B. 1990. Software Project Management Under Incomplete and Ambiguous Specifications. *IEEE T. Eng. Manage.* 37, 1 (February. 1990), 10-21. DOI=<http://dx.doi.org/10.1109/17.45260>.
- [18] Larman, C., and Basili, V.R. 2003. Iterative and Incremental Development: A Brief History. *Computer*. 36, 6 (June. 2003), 47-56. DOI=<http://dx.doi.org/10.1109/MC.2003.1204375>.
- [19] Stapleton, J. 2003. *DSDM: Business Focused Development*. Addison-Wesley, Boston, MA.
- [20] Palmer, S.R., and Felsing, M. 2001. *A Practical Guide To Feature-Driven Development*. Pearson Education.
- [21] Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., and Still, J. 2008. The Impact of Agile Practices on Communication in Software Development. *Empir. Softw. Eng.* 13, 3 (June. 2008), 303-337.
- [22] Dybå, T., and Dingsøyr, T. 2008. Empirical Studies of Agile Software Development: A Systematic Review. *Inform. Software Tech.* 50, 9-10 (August. 2008), 833-859.
- [23] Abrahamsson, P., Warsta, J., Siponen, M. T., and Ronkainen, J. 2003. New Directions on Agile Methods: A Comparative Analysis. *Proceedings of the 25th International Conference on Software Engineering* (Portland, Oregon, USA, May 3-10, 2003). IEEE Computer Society Washington, DC, 244-254.
- [24] Augustine, S. 2005. *Managing Agile Projects*. Prentice Hall, Upper Saddle River, NJ.
- [25] Gaddis, P. O. 1959. The Project Manager. *Harvard Bus. Rev.* 37, 3 (May-June 1959), 89-97.
- [26] Emam, K. El, and Koru, A. G. 2008. A Replicated Survey of IT Software Project Failures. *IEEE Software*. 25, 5 (September. 2008), 84-90. DOI=<http://dx.doi.org/10.1109/MS.2008.107>.
- [27] Ruparelia, N.B. 2010. Software development lifecycle models. *ACM SIGSOFT Softw. Eng. Notes*, 35, 3 (May. 2010), 8-13.
- [28] Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. Challenges of Migrating to Agile Methodologies. *Commun. ACM*. 48, 5 (May. 2005), 72-78. DOI=<http://dx.doi.org/10.1145/1060710.1060712>.
- [29] 10th Annual State of Agile Survey. *Versionone*. 2016. <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>. Accessed on: June 20, 2016.
- [30] 9th Annual State of Agile Survey. *Versionone*. 2015. <https://versionone.com/pdf/VersionOne-9th-Annual-State-of-Agile-Report.pdf>. Accessed on: June 20, 2016.
- [31] Hoda, R., Noble, J., and Marshall, S. 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Inform. Software Tech.* 53, 5 (May. 2011), 521-534. DOI=<http://dx.doi.org/10.1016/j.infsof.2010.10.009>.
- [32] Martin, A., Biddle, R., and Noble, J. 2010. An Ideal Customer: A Grounded Theory of Requirements Elicitation, Communication and Acceptance on Agile Projects. In *Agile Software Development*, T. Dingsøyr, T. Dybå, and N. B. Moe, Eds., ed: Springer Berlin Heidelberg, 111-141.
- [33] Glaser, B.G. 1992. *Basics of grounded theory analysis*. Sociology Press, CA.
- [34] Glaser, B.G., and Holton, J. 2004. Remodeling Grounded Theory. *Hist. Soc. Res.* 5, 2 (May. 2004), 47-68.
- [35] Kathy, C. 2014. *Constructing Grounded Theory*. Sage Publications Ltd., Thousand Oaks, CA.
- [36] Gambetti, R.C., Graffigna, G., and Biraghi, S. 2012. The Grounded Theory approach to consumer-brand engagement. *Int. J. Market Res.* 54, 5 (September. 2012), 659-687.
- [37] Grover, S.L., Hasel, M.C., Manville, C., and Serrano-Archimi, C. 2014. Follower reactions to leader trust violations: A grounded theory of violation types, likelihood of recovery, and recovery process. *Eur. Manage. J.* 32, 5 (October. 2014), 689-702. DOI=<http://dx.doi.org/10.1016/j.emj.2014.01.002>.
- [38] Javdani Gandomani, T., Zulzalil, H., Abdul Ghani, A.A., Md. Sultan, A.B., and Meimandi Parizi, R. 2015. The impact of inadequate and dysfunctional training on Agile transformation process: A Grounded Theory study. *Inform. Software Tech.* 57 (January. 2015), 295-309. DOI=<http://dx.doi.org/10.1016/j.infsof.2014.05.011>.
- [39] Burford, S. 2014. A grounded theory of the practice of web information architecture in large organizations. *J. Assoc. Inf. Sci. Technol.* 65, 10 (October. 2014), 2017-2034.
- [40] Whisenhunt, J.L., Chang, C.Y., Flowers, L.R., Brack, G.L., O'Hara, C., and Raines, T.C. 2014. Working With Clients Who Self-Injure: A Grounded Theory Approach. *J. Couns. & Dev.* 92, 4 (October. 2014), 387-397.
- [41] Lee, K.-H., and Zvonkovic, A.M. 2014. Journeys to remain childless: A grounded theory examination of decision-making processes among voluntarily childless couples. *J. Soc. Pers. Rel.* 31, 4 (June. 2014), 535-553.
- [42] Hoda, R., Noble, J., and Marshall, S. 2013. Self-Organizing Roles on Agile Software Development Teams. *IEEE T. Software Eng.* 39, 3 (March. 2013), 422-444, 2013. DOI=<http://dx.doi.org/10.1109/TSE.2012.30>.
- [43] Siddique, L., and Hussein, B. 2016. Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Manager's Perspective. *J. Eng. Project and Production Manage.* 6, 2 (July. 2016), 120-135.
- [44] Karlsen, J.T., and Gottschalk, P. 2006. Project Manager Roles in IT Outsourcing. *EMJ-Eng. Manag. J.* 18, 1 (March. 2006), 3-9.
- [45] Shastri, Y., Hoda, R., and Amor, R. 2016. Does the Project Manager still exist in Agile Software Development Projects? *Proceedings of the 23rd Asia-Pacific Software Engineering Conference*, Hamilton, New Zealand, December 6-9, 2016.