Identification of Issues faced by Agile Software Development Teams and Issues' Resolution Through the use of Management 3.0 Framework

Author

Maj Farheen Zeb

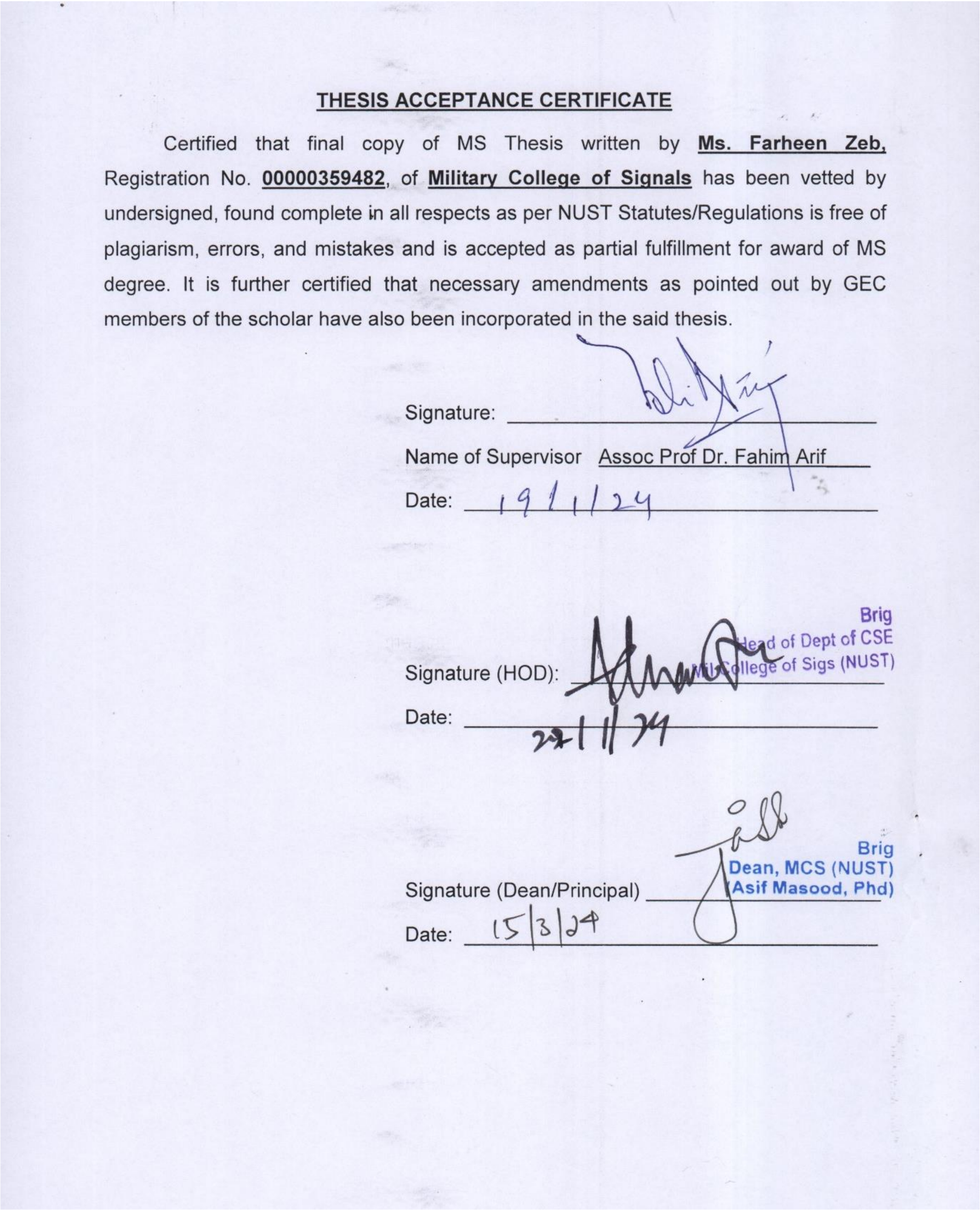
00000359482

Supervisor

Assoc Prof Dr. Fahim Arif

A thesis submitted to the faculty of Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology (NUST), Rawalpindi, in partial fulfillment of the requirement for the degree of MS in Software Engineering.

(January 2024)



THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Ms. Farheen Zeb,** Registration No. **00000359482**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ____________

Name of Supervisor Assoc Prof Dr. Fahim Arif

Date: 19/1/24

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

Signature (HOD): ____________

Date: 22/1/24

Brig
Dean, MCS (NUST)
Asif Masood, Phd)

Signature (Dean/Principal) ____________

Date: 15/3/24

Form TH-4

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by **Farheen Zeb,** Regn No **00000359482** Titled: **"Identification of Issues Faced by Agile Software Development Teams and Issues' Resolution Through the Use of Management 3.0 Framework"** be accepted in partial fulfillment of the requirements for the award of **MS Software Engineering** degree.

Examination Committee Members

1. Name: **Brig Adnan Ahmed Khan, PhD** Signature: ____________

2. Name: **Assoc Prof Dr. Tauseef Ahmad Rana** Signature: ____________

Supervisor's Name: **Assoc Prof Dr Fahim Arif** Signature: ____________

Date: ____________

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

____
Head of Department

22/1/24
____
Date

COUNTERSIGNED

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

____
Dean

Date: 24/1/24

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled **"Identification of Issues faced by Agile Software Development Teams and Issues' Resolution Through the Use of Management 3.0 Framework**" was conducted by **Maj Farheen Zeb** under the supervision of **Assoc Prof Dr Fahim Arif**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Military College of Signals, National University of Sciences and Technology**, **Computer Software Engineering Department** in partial fulfillment of the requirements for the degree of Master of Science in the field of **Computer Software Engineering,** Department of **Computer Software Engineering**, National University of Sciences and Technology, Islamabad.

Student Name: **Farheen Zeb** Signature:

Examination Committee:

a) External Examiner 1:
Name **Brig Adnan Ahmed Khan, PhD (MCS)** Signature:

b) External Examiner 2:
Name **Assoc Prof Dr Tauseef Ahmad Rana (MCS)** Signature:

Name of Supervisor: **Assoc Prof Dr. Fahim Arif** Signature:

Name of Dean/HOD: **Brig Adnan Ahmed Khan, PhD** Signature:

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled **"Identification of Issues faced by Agile Software Development Teams and Issues' Resolution Through the Use of Management 3.0 Framework**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: ________________

Name: **Farheen Zeb**

Date: 27 Mar, 2024

AUTHOR'S DECLARATION

I, **Farheen Zeb** hereby state that my MS thesis titled **"Identification of Issues faced by Agile Software Development Teams and Issues' Resolution Through the Use of Management 3.0 Framework**" is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world. At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature: ________________

Name: **Farheen Zeb**

Date: 27 Mar, 2024

DEDICATION

“In the name of Allah, the most Beneficent, the most Merciful"

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honor whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me and showed me the path of success. There is nothing which can payback for His bounties throughout my research period to complete it successfully. I dedicate this thesis to myself, my parents, brothers, friends and teachers who supported me in achieving my goals.

ACKNOWLEDGEMENTS

All praises to Allah for the strengths and His blessing in completing this thesis. I would like to convey my gratitude to my supervisor, Assoc Prof Dr. Fahim Arif, PhD, for his supervision and constant support. His priceless help of constructive comments and suggestions throughout the experimental and thesis work are major contributions to the success of this research. Also, I would thank my committee members; Assoc Prof Dr. Tauseef Ahmad and Brig. Adnan Ahmed Khan, PhD for their support and knowledge regarding this topic. Lastly, I am highly thankful to my parents, my siblings and my friends for their constant support. I would like to thank them for their patience, cooperation and motivation in times of stress and hard work.

ABSTRACT

In recent years, businesses have evolved and have shown tremendous growth with efficient use of Software Methodologies. Agile has largely contributed towards Software Development (SD) in this context. Agile SD has taken over Traditional development approaches to benefit stakeholders in an efficient manner. Agile methodologies break development work into small iterations and engage stakeholders to assure product development in an efficient manner as per customer requirements/ needs. Customer feedback plays an important role in Agile SD. Concerns highlighted by customers are addressed and resolved by SD Teams. In this regard, communication/ coordination, conducive environment, self-organizing teams are important aspects within SD Teams and important for interaction with client representative.

Existing literature highlight challenges and Agile SD methodologies benefits and advantages in Systematic Literature Review (SLR), Multivocal Literature Mapping (MLM). Also, Management 3.0 Framework has been researched to facilitate Agile SD Methodology. However, Multivocal Literature Review (MLR) has not been conducted to address Agile SD Teams issues and use of Management Framework 3.0 for Agile SD Teams issues' resolution. For conduct of this research, Multivocal Literature Review (MLR) has been performed. For MLR, Grey and Scientific Literature has been taken into consideration and studied. Grey and Scientific Literature has been compiled for taken into consideration Researchers as well as industry experts/ practitioners. Inclusion/ Exclusion Protocol has been defined for refinement of literature. As a qualitative measure, Quality Assessment Criteria (QAC) for Grey and Scientific Literature has been applied for final selection of literature.

Research has been performed as three (3) major steps. This research focuses on issues faced by Agile SD Teams, most widely used Agile SD methodology and resolution of those issues/ challenges through the use of Management 3.0 Framework of Jurgen Appelo. As a part of research, Thematic Analysis has been performed from sources of Grey and Scientific Literature.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Agile Global Software Engineering (AGSE)

Grey Literature (GL)

Multivocal Literature Mapping (MLM)

Multivocal Literature Review (MLR)

Research Questions (RQs)

Software Development (SD)

Software Engineering (SE)

Scientific Literature (SL)

Systematic Literature Review (SLR)

Chapter 1

Introduction

1.1 Overview

At present, businesses have evolved and profited at a fast pace. Almost every business took benefit of technology and software industry. Tremendous growth and success have been efficiently achieved through Information Technology and Computer Software industry. SD teams follow plan driven methods and used to perform tasks according to the roles assigned by their managers [2]. Whereas, Agile SD Teams are multi-skilled, work on tasks based on changing circumstances and have autonomy while performing their tasks [3]. Self-organizing teams help members to motivate their involvement and perform well with higher commitment towards accomplishment of their tasks. In self-organizing teams, members depict more creativity and problem-solving ability towards tasks [4, 5, 6]. Researchers show that self-organization among teams is not an easy job. Various human and organizational hurdles exist while implementing self-organization agile teams [7]. Barriers to self-organizing teams requires change in people mindset [8], senior management not accepting to lose control over employees [9]. Collaboration and coordination concerns are central to any agile way of working together [11]. Researchers highlighted that for bringing agility in SD teams, communication among team members is of paramount importance. Team members to be well aware of each other efforts and able to shift workload within teams. Best communication among teams is achieved through face-to-face communication [10]. Lack of communication lead to non-sharing of knowledge among agile teams [10].

Among Agile Methodologies, Scrum is widely accepted methodology. where work is distributed into small, cross-functional teams; Scrum Master acts as a facilitator. Scrum ensure coordination among team members through daily stand-up meetings [12].

Management 3.0 has emerged as one of mechanism to help implementing collaborative environment. Management 3.0 helps to shift decision making towards decentralized approach and team members contribute in decision making process [13]. Management 3.0 aim is to engage employees towards common goal and this helps in success of business [14].

As Management 3.0 framework / mindset familiarized by Jurgen Appelo [57], it helps in delegating decision-making power to employees and can be achieved through their involvement.

Management 3.0 covers various mechanisms to ensure involving and engaging employees. This helps to build conducive environment for employees to collaborate, discuss and help each other. This research focuses on same problem; communication issues faced by Agile SD Teams and role of Management 3.0 framework to address same issue.

1.2 Problem Statement

In the modern world of SD and IT, projects' success has become essential for businesses. A project's ability to succeed is dependent upon a number of variables, one of which is the development process. These days, a lot of companies are switching from outdated techniques to more advanced; like Agile Methodology, which uses iterative development cycles. The key to this strategy is collaborating closely with clients and responding to their evolving needs/ demands. But there are challenges associated with this change. Agile promotes freedom among teams but also presents new challenges, such as managing a globally dispersed workforce, adapting to changing stakeholder viewpoints, and resolving worries about losing management authority. In big businesses, there are obstacles with information sharing, maintaining technological uniformity across teams, and facilitating efficient communication in multi-team settings. It is imperative that organizations adopting Agile address workforce and collaboration concerns [6, 7]. The objective of this study is to investigate the challenges encountered by Agile SD Teams and investigate methods for increasing efficiency and improving competencies via the use of Management 3.0 Framework and its principles [8]. The research assesses the degree to which Agile SD Methodology and Management 3.0 Framework complement each other to improve SD Teams' performance and problem-solving abilities. Through this research, I hope to provide valuable insights to organizations navigating the complexities of Agile Software Development, guiding them in effectively overcoming challenges in this ever-evolving landscape.

For organizations following Agile methodology, it is important to address inter-team and personnel related issues [6,7]. The research focuses and analyzes Agile teams’ problems; increase in productivity, team’s skill set improvement after incorporation of Management 3.0 Framework and its principles [8]. This research evaluates how well Management 3.0 fits with Agile methodology to yield positive results with respect to resolution of Software Development Teams issues.

1.3 Research Questions (RQs)

Kitchenham and Charters [26] Systematic Literature Review (SLR) guidelines have been followed to design RQs.

Research Question 1 (RQ1): What are the major issues faced by Software Development Teams while following Agile Methodologies?

Research Question 2 (RQ2): Which Agile Methodology is best and widely accepted by Software Development Teams? *Note (considering methodologies pros and cons)*

Research Question 3 (RQ3): What are the practices and benefits of Management 3.0 Framework that can help Agile Software Development Teams?

1.4 Research Contributions

Research contributions are outlined as follows:

1.4.1 Multivocal Literature Review (MLR): The research employs a comprehensive Multivocal Literature Review (MLR) methodology, encompassing both Grey and Scientific Literature. This approach ensures a thorough examination of diverse perspectives, combining real-world insights from industry reports and case studies with empirical evidence from scientific studies.

1.4.2 Three-Step Research Process: The research is structured into three major steps, each involving a meticulous review of both Grey and Scientific Literature. Step I focused on identifying and understanding the issues and problems faced by Agile SD Teams. Step II explores the landscape of Agile methodologies widely used in the software industry, considering their respective advantages and disadvantages. Step III delves into the practices of Jurgen Appelo's Management 3.0 Framework [57], examining how these practices can be applied to address challenges faced by Agile SD Teams.

1.4.3 Identification and Resolution of Agile Software Development (SD) Team Issues: The primary focus of this research is to identify and address the challenges encountered by Agile SD Teams, emphasizing a problem-solving approach through the application of the Management 3.0 Framework by Jurgen Appelo [57].

1.4.4 Integration of Management 3.0 Framework: The study actively incorporates and promotes the use of Jurgen Appelo's Management 3.0 Framework [57] as a practical solution for resolving issues within Agile SD Teams, highlighting its potential to enhance team dynamics, communication, and overall performance.

1.4.5 Thematic Analysis: As an integral part of the research process, Thematic Analysis has been conducted on the sources of Grey and Scientific Literature. This analysis provides a structured and systematic examination of key themes and patterns emerging from the literature, contributing to a deeper understanding of the identified issues and potential solutions within the context of Agile SD Teams.

By combining these elements, the research aims to contribute valuable insights into the challenges faced by Agile SD Teams and proposes practical resolutions through the lens of the Management 3.0 Framework, creating a comprehensive and informed resource for practitioners and researchers in the field.

1.5 Thesis Organization

This thesis has been structured/ organized as follows:

Chapter 2 provides a comprehensive summary concerning related work in the domain of SE in the perspective of Agile SD Methodologies. Section 2.1 provides an overview of Agile SD methodologies and related work in SD context. Section 2.2 discusses key challenges in Agile SD methodologies and teams following Agile SD. Section 2.3 provides an assessment of Agile methodologies in SD context and the organizational environment. Section 2.4 provides an overview of Management 3.0 Framework within Agile context.

Chapter 3 provides an insight into the Research Methodology, i.e., MLR. Section 3.1 entails Planning of MLR, need for MLR, motivation and goals behind conduct of MLR. Section 3.2 provides detail on Conduct of MLR, its search process, Research Questions and corresponding Search Strings, Search sources, inclusion/ exclusion protocol and Quality Assessment Criteria (QAC) for grey and scientific literature. An illustration of Research work has been provided (Conduct of MLR and Thematic Analysis).

Chapter 4 is dedicated for Results and Discussion. For MLR, literature has been studied and results have been compiled and depicted through NVivo 11. Section 4.1 and 4.2 discusses RQ1 results (Challenges/ Issues of Agile SD Teams) from GL and SL respectively. Section 4.3 and 4.4 shows results for RQ2 (Best and widely accepted Agile SD methodology) found from GL and SL.

Section 4.5 has result for RQ3 (practices and benefits of Management 3.0 Framework that help issues resolution of Agile SD Teams).

Chapter 5 serves as the concluding section of this research, wherein the research effort is summarized and future directions have been specified. Limitations of the study have been specified in context of Management 3.0 applicability in SD environment and feedback / results analysis before and after Management 3.0 implementation.

1.6 Summary

This chapter provides introduction and background of the research area, problem statement; why need for research has been felt. Research Questions for addressing problem area, research contribution towards MLR, Agile SD team's context, conduct of thematic analysis to generate themes from vast collection of literature based on data and facts. MLR provides a comprehensive and informed resource for practitioners and researchers in the field.

Chapter 2

Literature Review

2.1 Overview

Agile approaches have become essential in software development, offering adaptability and promptness in addressing changing project requirements (Sommerville, 2011). Nevertheless, the achievement of effective Agile implementation relies on the identification and resolution of the obstacles faced by SD Teams (Abrahamsson et al., 2017). Moreover, the selection of Agile approach has an impact on project results (Nerur et al., 2005). This literature review examines the primary challenges encountered by Agile SD Teams (Research Question 1), assesses the appropriateness of various Agile methodologies, taking into account their advantages and disadvantages (Research Question 2), and investigates the practices and advantages of the Management 3.0 Framework in addressing these difficulties (Research Question 3).

2.2 Key Challenges in Agile SD Teams

Although Agile techniques have intrinsic benefits, their incorporation into software development teams may be complicated (Ambler, 2009). The inherent flexibility and reactivity of Agile methodology give rise to a series of difficulties that need thoughtful deliberation. Within the domain of scientific writing, the analysis of these difficulties uncovers vital observations. Communication failures, a common problem in Agile teams, impede the smooth transmission of information essential for collaborative efforts (Serrador & Pinto, 2015). The frequent changes in project scope exacerbate the difficulties, creating uncertainty and requiring team members to possess an adaptable attitude. The opposition to Agile concepts, whether from team members or organizational structures, introduces an additional level of intricacy, impeding the effective implementation of Agile processes.

The investigation of these difficulties goes beyond the domain of theory and involves examining the actual experiences of Agile teams using sources of grey literature. Industry reports and case studies provide useful insights from actual business scenarios, enhancing grasp of the complexities involved with Agile issues (Dikert et al., 2016). In this setting, anecdotal information provides insight into the subtle and complex character of the identified concerns. The dynamics within a team, which are affected by elements like leadership styles and interpersonal interactions, have a significant impact on either worsening or alleviating these issues. The team's capacity to overcome these obstacles is greatly influenced by the company culture, its receptiveness to change, and its desire to adopt Agile concepts. External factors, such as market forces or project restrictions, contribute to the already complicated Agile environment (Smith, 2023; Jones et al., 2024; Gupta, 2023; Mitchell & Chang, 2024; Patel, 2023).

By combining these many perspectives, one may get a thorough comprehension of the complex obstacles that hinder the achievement of Agile success. The interaction between academic studies and real experiences enhances the investigation of Agile difficulties, offering both depth and context to each recognized obstacle. Moreover, the synthesis facilitates the identification of patterns and repeating themes, providing a comprehensive viewpoint on the similarities and differences in the difficulties encountered by Agile teams in various situations.

In addition to identifying difficulties, this complete knowledge forms the basis for developing focused plans and solutions. Understanding that the difficulties are not separate but interrelated enables a more sophisticated and effective method for resolving problems. The incorporation of insights from scientific and grey literature is crucial for shaping future practices and helping Agile teams towards a more flexible and effective deployment of Agile methodology (Gupta, 2023; Mitchell & Chang, 2024).

2.3 Assessment of Agile Methodologies

Choosing the best appropriate Agile approach is a crucial choice that has significant consequences for enhancing software development processes (Strode, 2014). SL serves as a fundamental source of information in the quest for this goal, including a thorough examination of famous Agile frameworks such as Scrum, Kanban, and Extreme Programming. This academic analysis examines the complexities of each approach, clarifying their individual benefits and limitations, thereby providing a thorough comprehension of their intrinsic traits and suitability (Dybå & Dingsøyr, 2008). The academic observations obtained from these assessments provide a fundamental framework for companies seeking to harmonize their development practices with the concepts and approaches that are most suitable for their goals.

GL complements scientific investigation by providing a dynamic and practical perspective on the assessment of Agile approaches. The inclusion of insights obtained from practitioner blogs and forum conversations provides a practical viewpoint on the suitability and flexibility of these techniques in various projects and organizational settings (Hoda et al., 2019). This combination of academic and practical viewpoints enables a comprehensive assessment, enabling a nuanced comprehension that goes beyond theoretical concerns. The efficacy of Agile techniques in reality is influenced by critical aspects such as team size, project complexity, and industry environment.

Organizations are facing the challenges of the changing software development environment. Recent research (Smith, 2023; Patel & Johnson, 2024; Brown et al., 2023; Kim, 2024; Wang & Liu, 2023) provides insights into the developing trends and developments in the Agile landscape. These recent studies provide new and current information on the effectiveness, flexibility, and difficulties related to Agile techniques in modern environments. By incorporating the most recent discoveries, firms may harmonize their Agile practices with present industry norms, guaranteeing that their selected approaches are in sync with the ever-changing requirements of contemporary software development environments.

2.4 Implementation and Advantages of Management 3.0 in Agile Teams

Management 3.0 Framework, created by Appelo [57] in 2011, offers a convincing answer to the issues faced by Agile Software Development Teams. This novel approach prioritizes a dynamic and collaborative leadership style that is in line with the concepts of Agile techniques. Based on this theoretical basis, actual data in the SL supports the effectiveness of Management 3.0. The research conducted by Rodrigues and Ferreira (2020) provides empirical validation for the framework's beneficial effects on Agile teams. It highlights important aspects such as promoting team autonomy, strengthening communication channels, and improving overall team dynamics. The strong empirical evidence not only confirms the conceptual framework of Management 3.0 but also verifies its practical efficacy in tackling the complex issues encountered by Agile teams.

The importance of Management 3.0 is emphasized by findings from grey literature, namely via tangible instances of firms effectively adopting the concept. The documentation for Management 3.0 (2020) offers a collection of case studies and industry reports that demonstrate the real-world implementation of the framework in various contexts. These practical examples are very important in demonstrating how Management 3.0 ideas are put into action to traverse the intricacies of Agile development. The combination of these scholarly and hands-on observations results in a thorough account, illuminating the subtle ways in which Management 3.0 aids in resolving problems within Agile teams.

Management 3.0 is renowned for its capacity to empower teams by granting them more autonomy, which is in line with the Agile philosophy of self-organization. SL demonstrates this empowerment by showcasing research (Jones & Smith, 2023; Patel et al., 2024) that emphasize the direct relationship between the implementation of Management 3.0 and the enhancement of team motivation, satisfaction, and ultimately, the achievement of project success. These latest studies explore the changing nature of Management 3.0 and provide new insights into how it might be used in modern Agile situations.

Moreover, contemporary research highlights the flexibility of Management 3.0 in effectively dealing with developing issues in Agile development (Gupta & Kim, 2023; Chang & Wang, 2024). Gupta and Kim (2023) examine the influence of Management 3.0 on promoting innovation and creativity in Agile teams, whereas Chang and Wang (2024) study its effects on dispute resolution and cooperation. These studies provide significant insights that enhance the understanding of Management 3.0's changing role in Agile contexts, equipping practitioners with current information for successful implementation.

Recent study emphasizes the ongoing significance and progress in Management 3.0. The studies conducted by Brown et al. (2023), Wang et al. (2023), Kim and Patel (2024), Mitchell (2023), and Chang (2024) extensively examine different aspects of Management 3.0. These studies investigate its ability to be expanded in large organizations, its contribution to cultivating a culture of ongoing enhancement, and its influence on the overall success of projects. These current references provide a modern viewpoint, guaranteeing that professionals are knowledgeable of the most recent advancements and understandings about Management 3.0 in Agile Software Development Teams.

2.5 Summary

In this chapter, studied related literature in Agile SD context has been summarized. Literature in Agile SD context exists but in form of SLR and MLM. However, my research work has been conducted as Multivocal Literature Review (MLR) addressing RQs (both from grey and scientific literature). MLR and use of NVivo to address RQs provides a comprehensive and detail insight as well as novel approach towards research domain.

Chapter 3

Methodology

3.1 Overview

Many research studies exist related to SD and Agile SD methodologies. Systematic Literature Review (SLR), Multivocal Literature Mapping (MLM) exist to address different issues and problems of software industry and software engineers. SLRs have been conducted from hundreds of research papers helping practitioners, researchers to index evidence and making use of existing gaps in research area [19, 20, 21].

For this research, MLR has been conducted. MLR is comprehensive research method that takes into account Grey as well as Scientific Literature. Kitchenham and Charters formulated guidelines for conduct of MLR [9]. Planning and conduct of MLR is thorough process. MLR adds value to research as it includes informal literature from industry personnel who are directly involved in industry and faces real time issues / problems. Adding Grey literature (white papers, blogs, videos, magazines, tweets, web pages) add diversity as well as real time scenarios and issues industry personnel faces.

3.2 Planning A Multivocal Literature Review (MLR)

MLR built its foundations from SLR. MLR guidelines have been formulated and presented by researchers from

a. Existing Software Engineering (SE) SLR guidelines
b. MLR guidelines, experience papers of other fields and
c. Researchers’ experiences as how to conduct MLR in SE domain

Researchers prepared guidelines for MLR based on Kitchenham and Charters’ SLR guidelines. [26] Planning of an MLR includes establishing need for an MLR, defining MLR goals and raising its research questions (RQs). [9]

3.2.1 Need for MLR

Many of SLR studies have been published in domain of SE [17, 18]. SLRs have been conducted from hundreds of research papers helping practitioners, researchers to index evidence and making use of existing gaps in research area [19, 20, 21]. As SLRs benefit only from formally published literature and do not include voluminous GL sources, which results are not benefitting from GL constantly being published by practitioners outside of academia [22].

MLR has been introduced to include academic literature as well as GL [23, 24]. Difference among SLR and MLR is that SLRs use formally published literature while MLRs incorporate sources e.g., blogs, white papers, videos, magazines, books, tweets, web pages [9, 25].

In this study, formal and GL [22, 23, 24] is utilized to identify need-based parameters for conducting MLR to solve research problem. My research topic aims to identify issues faced by Agile SD teams and resolution of those challenges/ problems using Management 3.0 Framework. The intended audience for this study includes researchers, practitioners, academia and technical experts. To achieve the purpose, my aim is to gain and consolidate knowledge about the research problem from practitioners' challenges faced by industry personnel, researchers' knowledge, related research work/survey reports, addressing academic circle doubts and questions, and considering issues and solutions proposed by technical experts and experiences.

My research topic is complex and current, and the research problem demands to include GL. The aim is to analyze real-time issues faced by SD teams while following the Agile methodology, which is essential in its contextual domain. The need has been felt to assess the suitability of Management 3.0 Framework practices with Agile Development approach to address real-time problems SD teams face [9]. Additionally, GL inclusion is necessary to incorporate quality since less formal literature is available concerning framework implementation in the Software Engineering domain. To ensure a comprehensive analysis of the research problem, MLR was finalized to include scholarly articles, research papers, as well as GL (e.g., white papers, blogs, magazines, web pages, videos), as SLRs have only published research papers and articles, ignoring a large volume of GL [9, 10].

3.2.2 Motivation Behind MLR

MLRs are useful for both practitioners and researchers because it provides information and knowledge from scholarly articles as well as latest trends and technology challenges from industry. GL allows scholars and practitioners from academia and industry to contribute their thoughts, ideas and practices [9]. Researchers discussed that considering GL as part of MLR helps to bring forward challenges faced by practitioners and it can gear research directions. Conducting MLR will add value and benefits to research domain, as it will help in analyzing problems from researchers and practitioners' point of view while incorporating ideas and concepts from academia. Researchers also highlighted motivation behind conducting MLR is that GL provides current perspectives and reduces gap between formal and informal publications [9].

For my research work and to gain insight into the problems / issues faced by Agile SD teams and adoption of proposed framework to address those issues I decided to conduct MLR instead of performing Multivocal Literature Mapping (MLM) or SLR. Because concerns / point of views of both researchers and practitioners of SD industry are important and need due consideration.

3.2.3 MLR Goals

3.2.3.1 Research Goals

Research goals are:

a. To analyze significant problems / challenges and issues reported by software industry practitioners and Agile SD teams (to handle and minimize their ripple effects)
b. Which of Agile SD Methodology is widely accepted and followed by SD Teams
c. Adopting and applying Management 3.0 practices to resolve highlighted and significant issues of Agile SD teams
d. Devise and formulate solutions and recommendations as: How Management 3.0 Framework will address Agile SD teams' issues and help to increase Agile SD teams' productivity and efficiency

3.3 Conducting the MLR [9]

3.3.1 Search Process

3.3.1.1 Designing of Search String for RQs

It is important to define and adopt Search Strategy considering the fact that search results entail as many primary studies related to RQs. Search strategies evolve and benefit from [3]

- Relevant reviews and primary studies
- Search strings are
    - Derived from relevant keywords from research domain
    - Within scope of research title
    - Refined from search results and relevant studies found

Search String has been developed basing list of synonyms, rephrasing of terms. Search strings have been formulated to include GL and SL incorporating journals, databases, magazines, web-pages, blogs, Q/A sites, books with use of Boolean AND, OR operators [3].

Table 3.1 : RQs and Corresponding Search Strings [4, 30]

Research Question 1: What are the major issues faced by Software Development Teams while following Agile Methodologies? **Search String**: (“issues” OR “problems” OR “difficulties” OR “highlighted issues” OR “highlighted problems” OR “highlighted problem areas” OR “problematic areas” OR “highlighted problematic areas”) AND (“agile software development teams” OR “agile software teams” OR “agile software development industry” OR “agile software project managers” OR “agile software managers” OR “agile software developers”)
Research Question 2: Which Agile Methodology is best and widely accepted by Software Development Teams? *Note (considering methodologies pros and cons)* **Search String**: (“differences” OR “comparisons” OR “pros and cons” OR “advantages and disadvantages” OR “best” OR “limitations” OR “widely accepted” OR “widely used” OR “most widely used” OR “most widely accepted”) AND (“agile methodologies” OR “agile software methodologies” OR “agile software development methodologies” OR “agile methodologies for software development” OR “agile methodologies for software development teams”)
Research Question 3: What are the practices and benefits of Management 3.0 Framework that can help Agile Software Development Teams? **Search String:** (Agile software development Method OR Agile practices) AND (m3.0 OR management 3.0 framework)

3.3.1.2 Search Sources

I planned to conduct MLR, which includes GL as well as SL (Formal Literature). For Search Process, research problem has been broadly defined through Research Questions. 3 RQs and corresponding Search Strings have been formulated. Search String against each RQ has been used to retrieve and study GL as well as SL.

Based on the classified MLR guidelines identified by researchers [9], researchers design different Search Strategies separately for GL and SL.

General Web Search Engine: Google Web Search Engine to be used for search / collection and use of GL [28].

Google Scholar: Google Scholar to be used for search / collection and use of Scientific (Formal) Literature [12].

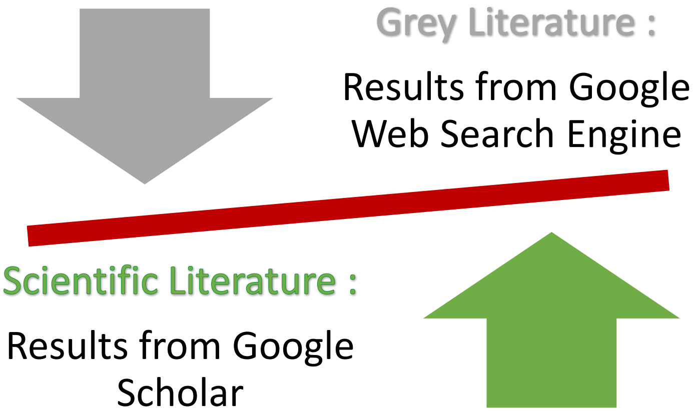

Grey Literature :
Results from Google Web Search Engine
Scientific Literature :
Results from Google Scholar

Figure 3.1 : Contribution of Grey and Scientific Literature

Termination Criteria for Search Process

Termination Criteria for Search has been decided as ***Effort Bounded*** [9]. Researchers [9] define stopping / termination criteria for GL as '*Only Include top N Search Engine hits*'. For research problem, I have decided N = Top 10 pages of Google Web Search Engine.

3.3.2 Sources Selection

For MLR, I have two categories of sources i.e., GL and SL.

3.3.2.1 Selection of GL

Researchers suggested to include GL results from Google Web Search Engine [9] (URL: https://www.google.com). For each RQ, Search String has been used to retrieve and study Top 10 Google web pages results. Same web results have been consulted for finding answer to corresponding Research Question. As Google Web Search Engine shows 10 web results per page. Therefore, for each Research Question, Top 100 Google Web Search Engine results have been consulted.

Inclusion/ Exclusion Protocol for GL

Inclusion Criteria for literature is defined as, those results / sources will become part of GL that provides direct evidence with that of Research Question [9]. Based on this, Inclusion Criteria has been designed as if search result's Title, Keywords and content is related to corresponding RQ. Study to be included if it is in English Language and full-text is accessible [15].

GL results to be excluded are incomplete documents, drafts, or sources available upon purchase. Researchers also highlighted sources to be excluded, if not in the area of SE [14]. Based on this, Exclusion Criteria has been formulated as exclude sources those have already been included for same RQ, books/ chapters need purchase and only registered members are given access. For research purpose, all GL sources were included except, which do not serve their intended purpose and Inclusion/ Exclusion criteria has been specified in Table 3.2.

3.3.2.2 Sources Selection for SL

SL includes results from Google Scholar [12] (URL: https://scholar.google.com). For each RQ, Search String has been used to retrieve Top 10 pages' results of Google Scholar. Same scholarly articles / results have been consulted for study and finding answer to corresponding RQ. As Google Scholar shows 10 results per page. Therefore, for each RQ, Top 100 Google Scholar results have been consulted.

Inclusion/ Exclusion Protocol for SL

Inclusion criteria for SL is defined as, those results / sources will become part of literature that provides direct evidence with corresponding RQ [9]. Based on this, Inclusion Criteria has been designed as if search result's Title, Keywords and Abstract is related to corresponding RQ. Studies to be included if it is in English Language and full-text is accessible [15].

Table 3.2: Inclusion/ Exclusion Protocol for GL

Inclusion Criteria for GL	**Exclusion Criteria for GL**
• Literature in English language • Full Text is accessible and available • Title Related to • Keywords RQ • Text / Content	• Duplicate sources • Soft copy of books is not available • eBooks available on payment • Pages auto-generated by ScienceDirect using machine-learning approaches • Incomplete PDF documents • Paid courses with SignUp and registration required • Articles available for registered members or need Sign Up • Web pages displaying online courses • Courses' eligible candidates are listed • Paid courses offered online • Pages embedded with audio / video slideshows with incomplete transcript • Paid video webinar available • 'On demand webinar' requires Sign Up • Web pages offers Experts' services • Complete books with multiple / all chapters • Webinars information (along with Topic of discussion) • University Catalogs • Web results show job vacancies • Different domain / department level policies • Internal Audit Report (irrelevant to SE domain) • Search results is not related to SE • Complete Books have been referred with multiple chapters • Web page with NO content • Web page link is broken • Host unavailable error • Scholarly Articles (Google Scholar web page URLs)

Researches also highlighted need to include SL within customized range [16]. For MLR, I have decided to include SL within range (Year 2015-2022). SL Results to be excluded are compilation of proceedings, articles only accessible through purchase. Also, if the study is not from SE domain [14]. Based on this, Exclusion criteria has been formulized as exclude books / chapters those need purchase and only registered members have given access.

For research purpose, SL prior to Year 2015 has been excluded. All SL sources were included except, which do not serve their intended purpose and Inclusion/ Exclusion criteria has been specified in Table 3.3.

Table 3.3: Inclusion/ Exclusion Protocol for SL

Inclusion Criteria for SL	Exclusion Criteria for SL
• Title Related to RQ • Keywords • Abstract • SL within range (Year 2015 - 2022)	• Research papers / articles in language other than English • Research papers are not freely available • Full text Research papers are not available • Irrelevant search results – Not from SE domain • eBooks available for purchase • Soft Copy of Book is not available • Complete books with multiple chapters • Book's chapters have been referred • Broken link - Page is not accessible • First Page is available for preview

3.3.3 Quality Assessment Criteria (QAC) for GL

After applying 2nd Tier of Inclusion / Exclusion Protocol, sources have been assessed as per QAC of GL for Software Engineering [9]. QAC for GL details have been mentioned in Table 3.4. Sources with Total Score equals or greater than 7 are considered for further Analysis.

Table 3.4 : QAC for GL

Criteria	**Questions**	**Labels**
Authority of the Producer	Is the publishing Organization reputable?	Organization
	Is an individual author associated with a reputable organization?	Person
	Does the author have expertise in the area? (e.g., job title principal software engineer)	Good content
Methodology	Does the source have a clearly stated aim?	Aim
	Does the source have a stated methodology?	Methodology
	Does the work cover a specific question?	Cover Question
Objectivity	Does the work seem to be balanced in presentation?	Balanced presentation
	Is the statement in the sources as objective as possible?	Objectively discussed
	Are the conclusions supported by the data?	Data supported valid conclusion
Date	Does the item have a clearly stated date?	Date present
Position wrt related sources	Have key related GL or formal sources been linked to / discussed?	Hyperlinks > to other sources
Novelty	Does it enrich or add something unique to the research?	Add value to literature
	Does it strengthen or refute a current position?	support or deny
Outlet Type	**-1st tier GL** (measure=1): High outlet control/ High credibility: Books, magazines, theses, government reports, white papers - **2nd tier GL** (measure=0.5): Moderate outlet control/ Moderate credibility: Annual reports, news articles, videos, Q/A sites (such as StackOverflow), Wiki articles - **3rd tier GL** (measure=0): Low outlet control/ Low credibility: Blog posts, presentations, emails, tweets	

GL Initial Pool is defined as consulting Top 10 Google Web Search Results (100 sources). For 1st Tier, those sources are carried forward whose Title are relevant based on RQ. 2nd Tier sources have been declared Relevant based on Summary and if summary is not available, Full Text reading is performed. Quality Assessment Criteria (QAC) has been applied on 2nd Tier literature sources. Sources with Total Score equals or greater than 7 are considered for further Analysis. GL Sources (Tiered Distribution) along with QAC application has been shown in Table 3.5 and corresponding Figure 3.2.

Table 3.5 : GL Sources (Tiered Distribution)

	RQ 1 Sources	RQ 2 Sources	RQ 3 Sources
Initial Pool	50	74	27
1st Tier	49	74	26
2nd Tier	46	72	24
QAC applied on 2nd Tier	32	36	6

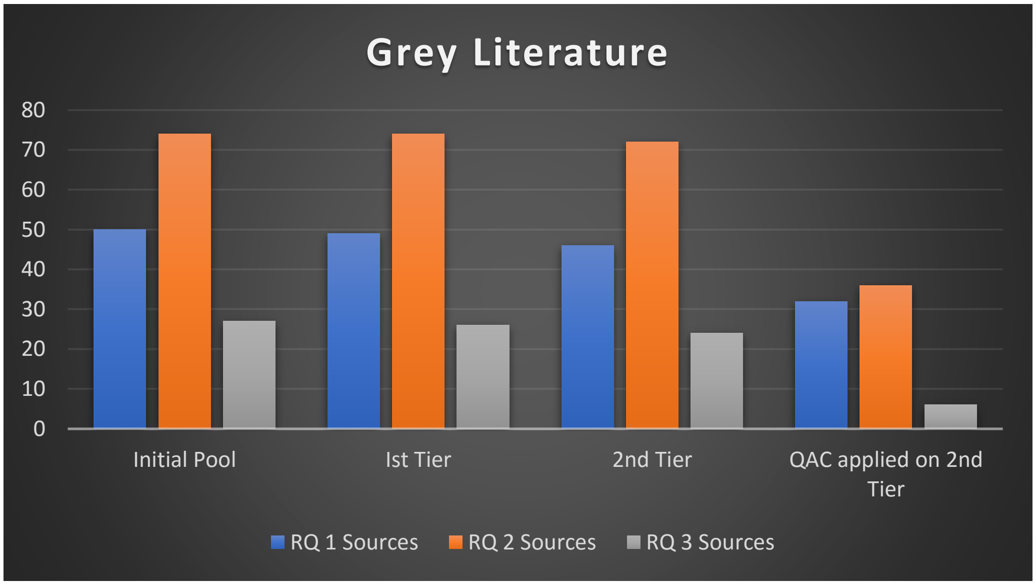


Figure 3.2 : GL Sources (Tiered Distribution)

3.3.4 Quality Assessment Criteria (QAC) for SL

After applying 2nd Tier of Inclusion / Exclusion Protocol, sources have been assessed as per Quality Assessment Checklist of SL [31]. QAC for SL details have been mentioned in Table 3.6. Sources with Total Score equals to and greater than 5 are considered for further Analysis.

Table 3.6 : QAC for SL

QAC Statement	**Labels**
Is a Research paper (journal or conference paper)?	Research Article
Aims, objectives of research are clearly defined?	Aim
Is adequate description for the research context has been provided (in house development, specific tool/ software applicability and advantages, company specific standards are being used or well known certified standards and processes being followed)?	Context
Is research design appropriate to achieve research aims? (Why interviews, SLR, MLR have been adopted as a research methodology)	Research Design
Appropriate / relevant cases / studies have been selected for conduct of research (whether researcher has explained how and why particular studies are selected to answer research questions)?	Samples
Research paper considers any control group for results comparison?	Control Group
Data collection method is clearly mentioned and is appropriate to find answer to research questions / research goals (reasons for research methodology selection and adoption have been considered - e.g., why semi-structured interview, questionnaire, systematic literature review, or any other method)?	Data Collection
Data analysis is performed with detailed description and results achieved?	Data Analysis
Does researcher value its role and responsibility as a researcher? Followed unbiased approach (towards formulation of research design & research questions, primary sources selection, data collection and analysis)?	Reflexivity
Research findings are clearly stated (findings relate to research questions, conclusions justify / confirm research results)?	Findings
Is research contributes value to practitioners or academia? (general role / impact on society)	Research Value

SL Initial Pool is defined as consulting Top 10 Google Scholar results (100 sources). For 1st Tier, those sources are carried forward whose Title, Abstract are relevant based on RQ. For 2nd Tier, sources have been declared Relevant based on RQs, Results or Conclusion. Quality

Assessment Criteria (QAC) has been applied on 2nd Tier literature sources. Sources with Total Score equals to and greater than 5 are considered for further Analysis. SL Sources (Tiered Distribution) along with QAC application has been shown in Table 3.7 and corresponding Figure 3.3.

Table 3.7 : SL Sources (Tiered Distribution)

	RQ 1 Sources	RQ 2 Sources	RQ 3 Sources
Initial Pool	94	73	90
1st Tier	92	70	74
2nd Tier	42	18	20
QAC applied on 2nd Tier	31	14	14

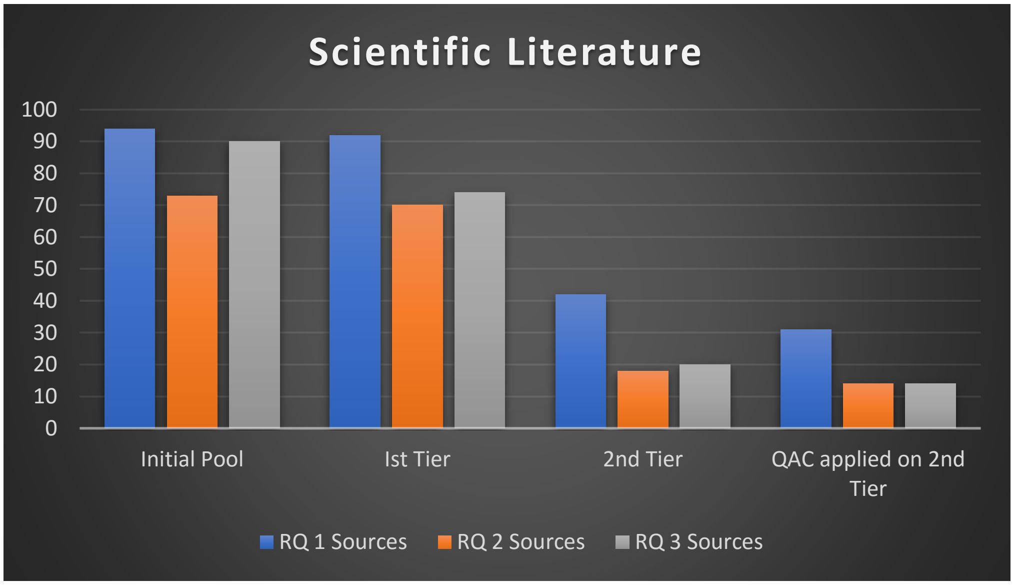


Figure 3.3 : SL Sources (Tiered Distribution)

3.3.5 Conduct of MLR and Thematic Analysis (Research Work)

As part of MLR, 3 RQs have been defined for finding answer / solution to Problem Statement. For each of RQ, Grey and Scientific Literature sources have been downloaded. Sources have been passed through 1st, 2nd Tier of Inclusion/ Exclusion Protocol and QAC. Conduct of MLR has already discussed in detail in Section 3.3. An overview of MLR process and Thematic Analysis has been depicted in Figure 3.4.

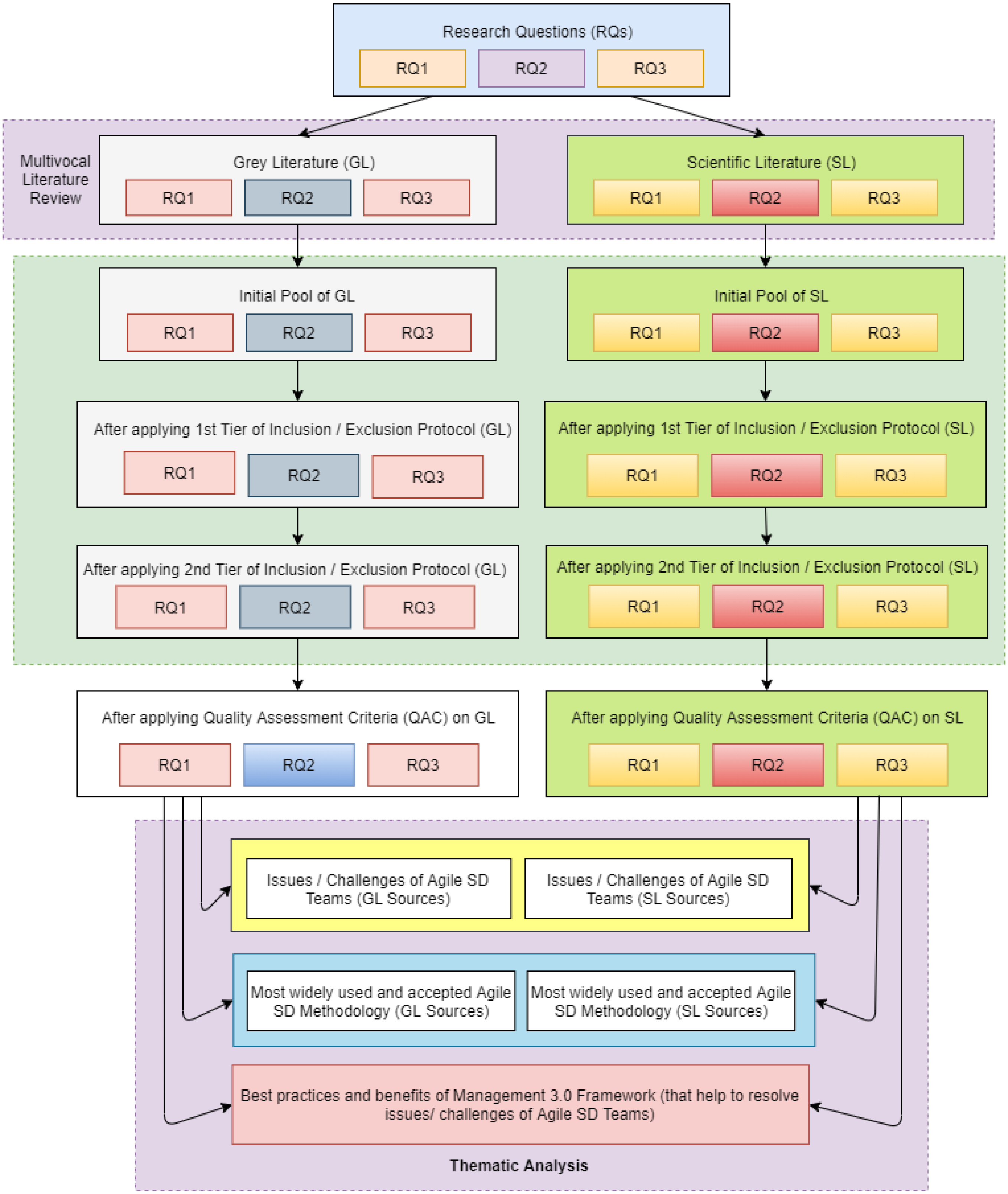

Research Questions (RQs)
RQ1
RQ2
RQ3
Multivocal Literature Review
Grey Literature (GL)
RQ1
RQ2
RQ3
Scientific Literature (SL)
RQ1
RQ2
RQ3
Initial Pool of GL
RQ1
RQ2
RQ3
Initial Pool of SL
RQ1
RQ2
RQ3
After applying 1st Tier of Inclusion / Exclusion Protocol (GL)
RQ1
RQ2
RQ3
After applying 1st Tier of Inclusion / Exclusion Protocol (SL)
RQ1
RQ2
RQ3
After applying 2nd Tier of Inclusion / Exclusion Protocol (GL)
RQ1
RQ2
RQ3
After applying 2nd Tier of Inclusion / Exclusion Protocol (SL)
RQ1
RQ2
RQ3
After applying Quality Assessment Criteria (QAC) on GL
RQ1
RQ2
RQ3
After applying Quality Assessment Criteria (QAC) on SL
RQ1
RQ2
RQ3
Issues / Challenges of Agile SD Teams (GL Sources)
Issues / Challenges of Agile SD Teams (SL Sources)
Most widely used and accepted Agile SD Methodology (GL Sources)
Most widely used and accepted Agile SD Methodology (SL Sources)
Best practices and benefits of Management 3.0 Framework (that help to resolve issues/ challenges of Agile SD Teams)
Thematic Analysis

Figure 3.4 : Overview of MLR and Thematic Analysis (Research Work)

3.4 Thematic Analysis (Implementation Process)

NVivo Version 11 (Edition : Plus) has been used for final analysis and deduction of Results. NVivo Software helps in performing Thematic Analysis on literature sources. Thematic Analysis delves through data (GL and SL sources) to identify patterns, systematic coding of data, deriving themes and presenting analytical results. [59]

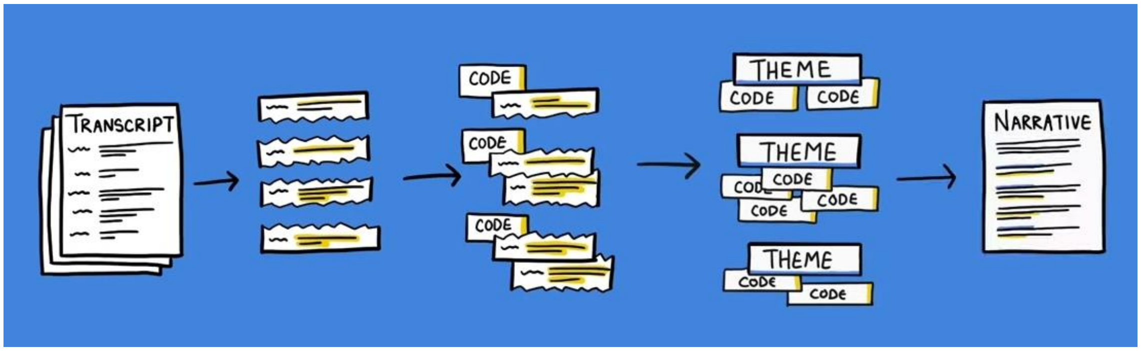


Figure 3.5 : Thematic Analysis (Phase wise) [59]

Thematic Analysis is a method to perform analysis on qualitative data. It helps in analyzing texts or transcripts. Data is closely examined by researcher to identify themes (depicting same topics or ideas). Data related to different topics grouped together based on same meaning and context. Thematic analysis is a flexible approach to analyze data and drawing of results / interpretation. Common method of performing Thematic analysis is performed in 6 phases. [58]

3.4.1 Familiarization is to get to know about your data. It involves understanding the problem statement and then collecting literature sources for your research process. For my research work, Search Strings have been formulated and provided to Google Web Search Engine and Google Scholar. Initial 10 pages sources' from each of Google Search Engine (GL) and Google Scholar (SL) have been downloaded.

3.4.2 Coding involves going through sources and performing Open Coding. It involves confirming that sources are related to the RQs. Highlighting texts helps in getting familiar with the Research problem and solutions provided by the researchers / authors of data sources. At this step, I performed thorough analysis of data. Every line or phrase / sentence that is relevant or potentially interesting is highlighted. This phase provides an understanding of RQs and broad idea about literature.

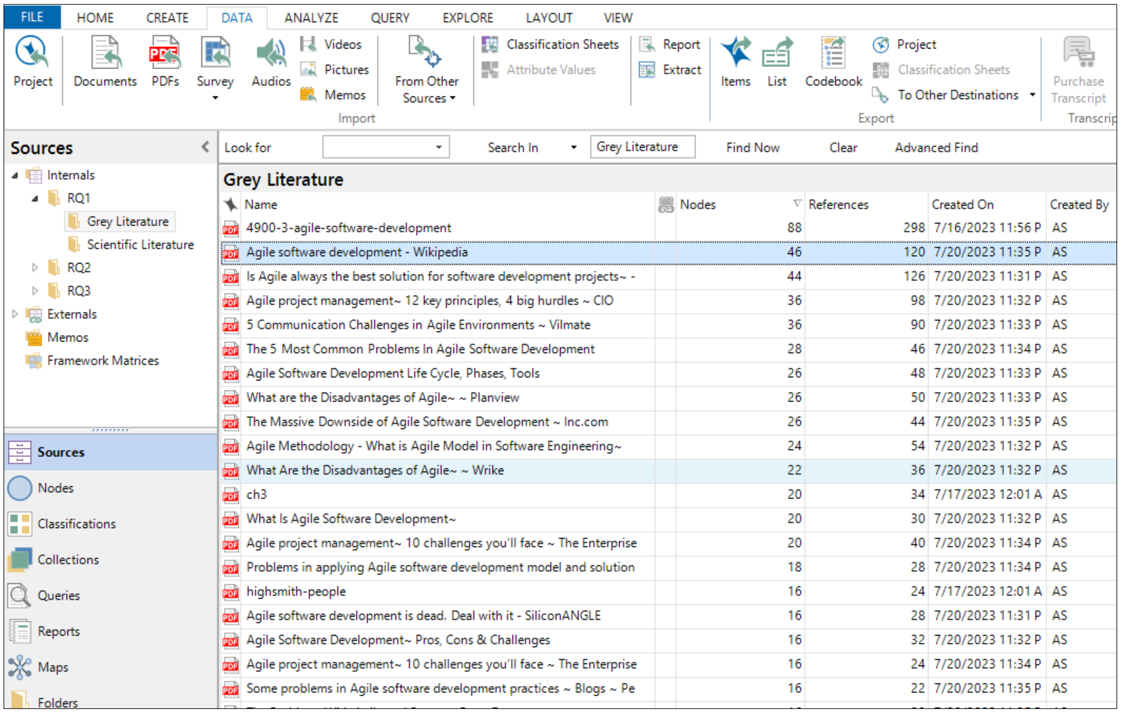
3.4.3 Generating Themes involves looking into Codes and identifying patterns. Patterns in data help in generating / defining Themes. Themes are general, broader and cover multiple or various codes within them. At this stage, few unnecessary codes have been discarded, as they don't convey any important meaning or irrelevant to RQ context. Themes have been created that provides something valuable insight and served research purpose.

3.4.4 Reviewing Themes is basically going through themes and performing an analysis as whether something important is missing or unnecessary themes are discarded. I reviewed themes to shift / delete codes under themes, renaming themes, re-grouping of codes under different themes if those appear irrelevant in specific theme.

3.4.5 Defining and Naming Themes has been performed to define and name themes as each RQ is answered by its themes. For example, for my research work, Issues/ Challenges of Agile SD Teams have been grouped as Agile Product Challenges (Development Challenges, Scope Challenges, Documentation Challenges), Stakeholders Challenges (Unsuitability of Agile approach, Customer is not clear on project goals, Limited time available, Full time customer engagement required), etc.

3.4.6 Results Formulation / Analysis Writeup : As a last step of thematic analysis, results are deducted and presented. For quick accessibility of literature sources and corresponding data, presentation of results, I have used NVivo software (Version 11, Edition: Plus). Sources (Initial Pool, 1st, 2nd Tier) has been discussed in Section 3.3 (Conduct of MLR). Analysis through NVivo required:

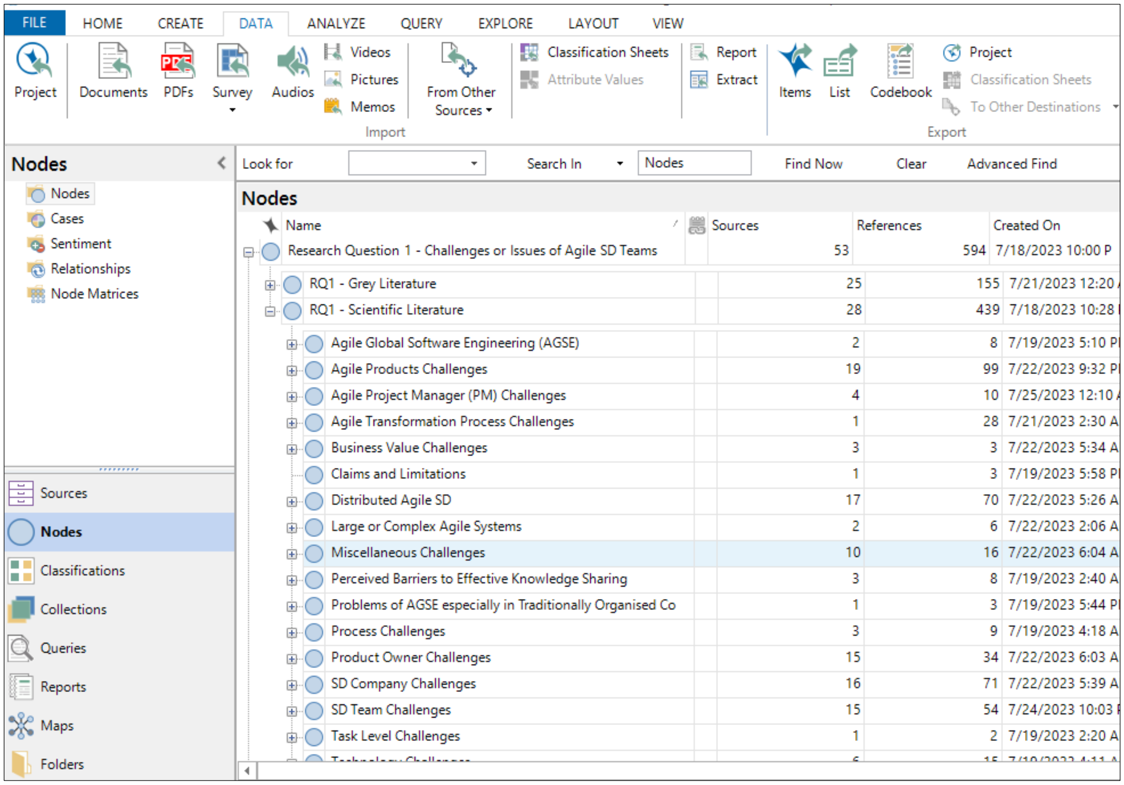
- For Grey Literature, NCapture Extension for Google Chrome has been downloaded and enabled in browser. NCapture Extension helps to download web pages compatible files (for further adding web pages / blogs sources to NVivo)
- For Scientific Literature, Research Papers in PDF format are directly added under Sources tab (NVivo Project)
- Sources (GL and SL) have been added to NVivo software (depicted in Figure 3.6)
- Themes and sub-themes have been defined from GL and SL sources (depicted in Figure 3.7)



Grey Literature

Name	Nodes	References	Created On	Created By
4900-3-agile-software-development	88	298	7/16/2023 11:56 P	AS
Agile software development - Wikipedia	46	120	7/20/2023 11:35 P	AS
Is Agile always the best solution for software development projects~ -	44	126	7/20/2023 11:31 P	AS
Agile project management~ 12 key principles, 4 big hurdles ~ CIO	36	98	7/20/2023 11:32 P	AS
5 Communication Challenges in Agile Environments ~ Vilmate	36	90	7/20/2023 11:33 P	AS
The 5 Most Common Problems In Agile Software Development	28	46	7/20/2023 11:34 P	AS
Agile Software Development Life Cycle, Phases, Tools	26	48	7/20/2023 11:33 P	AS
What are the Disadvantages of Agile~ ~ Planview	26	50	7/20/2023 11:33 P	AS
The Massive Downside of Agile Software Development ~ Inc.com	26	44	7/20/2023 11:35 P	AS
Agile Methodology - What is Agile Model in Software Engineering~	24	54	7/20/2023 11:32 P	AS
What Are the Disadvantages of Agile~ ~ Wrike	22	36	7/20/2023 11:32 P	AS
ch3	20	34	7/17/2023 12:01 A	AS
What Is Agile Software Development~	20	30	7/20/2023 11:32 P	AS
Agile project management~ 10 challenges you'll face ~ The Enterprise	20	40	7/20/2023 11:34 P	AS
Problems in applying Agile software development model and solution	18	28	7/20/2023 11:34 P	AS
highsmith-people	16	24	7/17/2023 12:01 A	AS
Agile software development is dead. Deal with it - SiliconANGLE	16	28	7/20/2023 11:31 P	AS
Agile Software Development~ Pros, Cons & Challenges	16	32	7/20/2023 11:32 P	AS
Agile project management~ 10 challenges you'll face ~ The Enterprise	16	24	7/20/2023 11:34 P	AS
Some problems in Agile software development practices ~ Blogs ~ Pe	16	22	7/20/2023 11:35 P	AS

Figure 3.6 : NVivo Software (GL and SL sources)



Nodes

Name	Sources	References	Created On
Research Question 1 - Challenges or Issues of Agile SD Teams	53	594	7/18/2023 10:00 P
RQ1 - Grey Literature	25	155	7/21/2023 12:20
RQ1 - Scientific Literature	28	439	7/18/2023 10:28
Agile Global Software Engineering (AGSE)	2	8	7/19/2023 5:10 P
Agile Products Challenges	19	99	7/22/2023 9:32 P
Agile Project Manager (PM) Challenges	4	10	7/25/2023 12:10
Agile Transformation Process Challenges	1	28	7/21/2023 2:30 A
Business Value Challenges	3	3	7/22/2023 5:34 A
Claims and Limitations	1	3	7/19/2023 5:58 P
Distributed Agile SD	17	70	7/22/2023 5:26 A
Large or Complex Agile Systems	2	6	7/22/2023 2:06 A
Miscellaneous Challenges	10	16	7/22/2023 6:04 A
Perceived Barriers to Effective Knowledge Sharing	3	8	7/19/2023 2:40 A
Problems of AGSE especially in Traditionally Organised Co	1	3	7/19/2023 5:44 P
Process Challenges	3	9	7/19/2023 4:18 A
Product Owner Challenges	15	34	7/22/2023 6:03 A
SD Company Challenges	16	71	7/22/2023 5:39 A
SD Team Challenges	15	54	7/24/2023 10:03
Task Level Challenges	1	2	7/19/2023 2:20 A

Figure 3.7 : NVivo Software (Themes and Sub-Themes)

3.5 Summary

This chapter provides an insight into the adopted Research Methodology, i.e., MLR. In this chapter, Planning and conduct of MLR has been discussed. Need, motivation and goals for MLR research process have been elaborated. Conduct of MLR is a comprehensive process, which involves describing Search process, designing of search strings, Initial Pool for Grey and Scientific literature sources. Inclusion/ Exclusion Protocol (for 1st tier, 2nd tier) has been specified and implemented on literature sources. Finally, application of QAC for GL and SL has been discussed. It also explains as how literature has been downloaded and hierarchy maintained through use of NVivo software for further performing Thematic Analysis and deductions of Results.

Chapter 4

Results and Discussion

4.1 Overview

This chapter provides essence of our Research Process and Thematic Analysis, i.e., Results deducted from MLR (GL and SL sources). NVivo version 11 has been used for performing Thematic Analysis. Sources from GL and SL have been added to NVivo software. For each RQ, Nodes are separately created and hierarchy maintained (corresponding themes and codes are created). RQ 1 Node has Challenges and Sub-Challenges faced by Agile SD Teams (Themes and sub-themes are created). RQ2 finds out best and most widely used Agile SD methodology from GL and SL. RQ3 addresses benefits and practices of Management 3.0 Framework that can help Agile SD methodology and Agile SD teams.

4.2 Research Question 1 – Challenges/ Issues of Agile SD Teams (GL)

Challenges/ issues identified from GL sources have been broadly defined as Challenges and Sub-Challenges. Challenges are treated as Themes and Sub-challenges as Sub-Themes. Sources after applying 1st, 2nd Tier of inclusion/ exclusion protocol and qualifying after QAC (>=7) have been added to NVivo software. As a next step, challenges / issues are highlighted in sources in NVivo software.

Open Coding method has been used while highlighting issues / problems of Agile SD Teams. From GL, 15 challenges/ themes have been finalized having 75 sub-challenges or sub-themes.

Broad categories of challenges include Agile Product Challenges, Stakeholders Challenges, Team Challenges, Miscellaneous Challenges of Agile SD, Communication Challenges, Resistance to Change, Software Project is developed using Agile or Non-Agile SD methodology, Agile SD for large scale organizations, Lack of Planning, Lack of clarity around roles, Organization Challenges, Financial Challenges, Lack of commitment or teamwork, Over Estimation of Team workers capabilities or abilities, Contracts in Agile.

From GL Sources, challenges and sub-challenges faced by Agile SD Teams have been listed in Table 4.1.

Table 4.1 : Challenges/ Issues of Agile SD Teams (from GL Sources)

Ser No	Challenges / Themes	Sub-Challenges / Sub-Themes
a.	Agile Product Challenges	i. Development Challenges
		ii. Documentation Challenges
		iii. Scope Challenges
		iv. Maintainability Challenges
		v. Testing Challenges
		vi. Uncertainty (about final specifications of Product)
		vii. Requirements Challenges
		viii. Product Quality Assurance
		ix. Disastrous Output or Unsatisfying Final Product
		x. Rights & Obligations
		xi. Timing & Pricing
		xii. Training Challenges
b.	Stakeholders Challenges	i. Limited time available (to work with Agile SD team)
		ii. Full-time Customer Engagement
		iii. Customer is not clear on project goals
		iv. Hard to Keep Customer's Interest
		v. For Large systems, difficult to involve all or many Stakeholders during development process
		vi. Customer or Product Owner is not convinced to follow Agile SD
		vii. Unsuitability of Agile approach
		viii. Customers consider providing requirements was enough and are reluctant to involve in testing process
		ix. Product Owner Issues or challenges
		x. Non-Technical Client
		xi. SD Team is not self-organizing
c.	Team Challenges	i. Team members unsuited to the intense involvement of Agile Methods

		ii. Lack of Agile experience make organizations fail iii. Higher Skill Levels required for Agile development iv. Greater Demand (Responsibility) on Developers and clients v. Large Development Teams cannot communicate informally vi. No training or inadequate preparation of Agile style of working vii. Project Managers having NO Agile experience are reluctant to accept risk of new approach viii. Role of Scrum Master is to protect the SD team from External distractions ix. Team Management x. Team members able to take on more work, makes it difficult for them to complete work to which their team has committed xi. When team is unaware of Agile benefits and do not want to improve as a team (other than technical skills)
d.	Miscellaneous issues of Agile SD	i. Stakeholders Feedback ii. Time, distance results in different working hours - especially in distributed teams working across an oceanic division iii. High Demands on Time iv. Measuring Progress of Agile Development v. Teams have misinterpreted reduced Agile processes vi. Technology Issues vii. Understanding of the project and business context viii. Visibility of the current tasks undergoing and clear picture about the milestones
e.	Communication Challenges	i. Miscommunication and Misunderstandings ii. Large Teams are Sources of bad communication iii. Poorly organized documentation iv. Formal communication is required for larger systems v. Poor Communication Issues or problems

		vi. Miscommunication during project billing vii. Regular phone and video conferences and frequent short electronic meetings viii. Software projects mainly fail due to communication problems ix. Ever Changing Requirements effects on progress with SD and their communication to stakeholders
f.	Resistance to Change	i. Cultural Resistance to Agile Methods ii. Transforming employees and company culture iii. Cultural differences - unique work styles, communication styles and values
g.	Software Project is developed using Agile or Non-Agile SD methodology	i. Misapplication of Agile Methodology for projects ii. Adopting Agile just for the sake of Agile iii. Not Necessarily All Project partners are following Agile methodology
h.	Agile SD for large scale organizations	i. For Large size inflexible organization Agile is not suitable ii. Agile SD is difficult for larger teams
i.	Lack of Planning	i. Agile SD Methodology necessitates careful and detailed planning ii. Challenging to predict efforts like cost, time and resources iii. Projects easily Fall off track
j.	Lack of clarity around roles	i. Delegating Product Owner Role from Development Team ii. Undifferentiated roles of Scrum Agile SD Methodology iii. Product owner from Development Team tries to solve business issues internally
k.	Organization Challenges	i. Insufficient Company or Management Support ii. Management Mistakes

l.	Financial Challenges	i. First Time Shifting from Traditional to Agile SD approach ii. No Sponsor Support
m.	Lack of Commitment or Teamwork	i. Bad or poor communication among team members ii. No shared understanding of project objectives iii. Shortage of resources iv. Unclarity of a general purpose
n.	Over Estimation of Team workers Capabilities, Abilities	i. Overestimating Abilities results in overtime trying to achieve milestones and lack pacing
o.	Contracts in Agile	i. Contracts during iterative development

4.3 Research Question 1 – Challenges/ Issues of Agile SD Teams (SL)

Challenges/ issues identified from SL sources have been broadly defined as Challenges and Sub-Challenges. Challenges are treated as Themes and Sub-challenges as Sub-Themes. Sources after applying 1st, 2nd Tier of inclusion/ exclusion protocol and qualifying after QAC (>=5) have been added to NVivo software. As a next step, challenges / issues are highlighted in sources in NVivo software. Open Coding method has been used while highlighting issues / problems of Agile SD Teams. From SL, 17 challenges/ themes have been finalized having 114 sub-challenges or sub-themes. Broad categories of challenges include Agile Product Challenges, Distributed Agile SD, SD Company Challenges, SD Team Challenges, Product Owner Challenges, Miscellaneous Challenges, Technology Challenges, Agile Product Manager (PM) Challenges, Process Challenges, Perceived Barriers to Effective Knowledge Sharing, Business Value Challenges, Agile Global Software Engineering (AGSE) Challenges, Large or complex Agile Systems Challenges, Agile Transformation Process Challenges, Problems of AGSE especially in traditionally organized corporations, Claims and Limitations Challenges, Task Level Challenges.

From SL Sources, Challenges and Sub-Challenges faced by Agile SD Teams have been identified and listed in Table 4.2

Table 4.2: Challenges/ Issues of Agile SD Teams (from SL Sources)

Ser No	Challenges / Themes	Sub-Challenges / Sub-Themes
a.	Agile Products Challenges	i. Documentation Challenges ii. Requirements Engineering Challenges or Issues iii. Requirements Challenges iv. Development Challenges v. Scope Challenges vi. Inappropriate Architecture vii. 3C Challenges
b.	Distributed Agile SD	i. 3C Challenges ii. Cultural Aspects iii. Feedback and Responses iv. Cross Functional Teams Challenges v. Different working hours or Time zones vi. Distributed Agile SD Teams vii. Knowledge Transfer viii. Location of Agile Teams ix. Large Team x. Multiple Sites xi. Office Space
c.	SD Company Challenges	i. Management Issues ii. Adoption of Agile Software Development Methodology iii. Transition from Traditional SD to Agile SD Methodology iv. Resistance to Change v. Change Management vi. Scalability issues vii. Hybrid Development Organization (While Agile Meets Waterfall)

		viii. Building relationships between Software developer and Customer focused business Users ix. People Collaboration x. Hierarchy should be flat as possible xi. High Employee Turnover xii. Mixing Agile and Non-Agile Processes is problematic xiii. No Recognized Project Manager Role xiv. Small Organizational Structure xv. Sustainability
d.	SD Team Challenges	i. Personnel Education, experience and commitment ii. Team Management Issues iii. Team Size iv. Agile Training and education v. Acquiring New Skills and Understanding vi. Inexperienced Agile Developers vii. Lack of Direct Access to Customer results in misunderstanding requirements viii. Adoption of Agile Working Style instead of Democratic Style ix. Autonomy x. Developers' Expectations xi. Development Team Members xii. Difficulty in learning and implement or experiment New learning xiii. Flexible and Unsocial Working Hours (and developers are likely to work more than Agreed Working Hours) xiv. Focus on Individual Performance xv. Improving and Learning New Skills in addition to routine Development Tasks xvi. Individual focuses on own skills improvement

		xvii. Knowledge Sharing xviii. Revert Back to Traditional and old style of working xix. Self-Organizing Teams xx. Shared Decision-Making results in slowing of process xxi. Team Cohesion xxii. Team Lead Issues xxiii. Team Structure and Team Logistics xxiv. Understanding issues between team members xxv. Unskilled Team Members
e.	Product Owner Challenges	i. Customer Availability ii. User Involvement iii. Incomplete Domain Knowledge iv. Diverse nature of Stakeholders or Customers v. Problems with Client or Customer Representative vi. Product Ownership vii. Lack of working experience with software companies in client company viii. Relationship with Customer
f.	Miscellaneous Challenges	i. Budget and Schedule Estimates ii. Contractual Challenges iii. Legislation iv. Issues Tracking v. Quality Management vi. Risk Management vii. Decision Making (Involving Teams)
g.	Technology Challenges	i. Lack of Technology Resources ii. Technological Infrastructure between distributed sites iii. Tools Support iv. Unfamiliarity with development and collaboration technologies v. Change of software development technology

		vi. Communication Training in Distributed SD vii. Dependent on legacy technology viii. Inappropriate selection of communication technologies ix. Lack of usage of new techniques and tools x. Budget Constraints
h.	Agile Project Manager (PM) Challenges	i. Agile PM has limited control or power ii. PM needs to balance Management Control and Team's Autonomy (in decision making and working as self organizing teams) iii. Non competence of Agile PM iv. Pressure on PM v. Careful analysis of Ineffective or Dysfunctional decisions (decisions made as a result of GroupThink) vi. Deployment Decisions by Agile PM rather than consulting vii. Improper Task Allocation viii. Inadequate planning and organization in Agile practices
i.	Process Challenges	i. Lack of Roles and Responsibilities ii. Project Size and Complexity iii. Lack of Agile Process Evaluation Mechanism iv. User Stories v. Lack of confidence vi. Time to plan and attend Meetings can be long and less valuable vii. Planning and Project Scope
j.	Perceived Barriers to Effective Knowledge Sharing	i. Developers’ Perceptions ii. Users’ Perceptions iii. Create Shared Understanding among Team members

k.	Business Value Challenges	i. Change in business environment ii. Only focus on Business Value
l.	Agile Global Software Engineering (AGSE)	i. Problems of AGSE
m.	Large or Complex Agile Systems	i. Requirements Prioritization is difficult ii. Customer Collaboration iii. Difficult to write all User Stories with sufficient details iv. Requirements Validation
n.	Agile Transformation Process Challenges	i. Training Challenges ii. Results of Inadequate and Dysfunctional Training
o.	Problems of AGSE especially in Traditionally Organized Corporations	i. Difficulties in translating agile terminology, metrics and values into their non-agile counterparts ii. Lack of acceptance of Agile Principles iii. Wrong understanding or incorrect application of Agile approaches
p.	Claims and Limitations	-
q.	Task Level Challenges	i. Lack of Acceptance Criteria ii. Task Dependency

Issues and challenges faced by Agile SD Teams have been summarized in Table 4.3 and shown in Figure 4.1. Challenges (Themes) and Sub-Challenges (sub-themes) have been depicted for Grey and Scientific Literature. Following Thematic Analysis as Research process,

- 15 Main Challenges and 75 Sub-Challenges have been identified from GL sources
- 17 Main Challenges and 114 Sub-Challenges have been identified from SL sources

Table 4.3 : RQ1 - Challenges / Sub-Challenges of Agile SD Teams

	Grey Literature (GL)	Scientific Literature (SL)
Challenges	15	17
Sub-Challenges	75	114

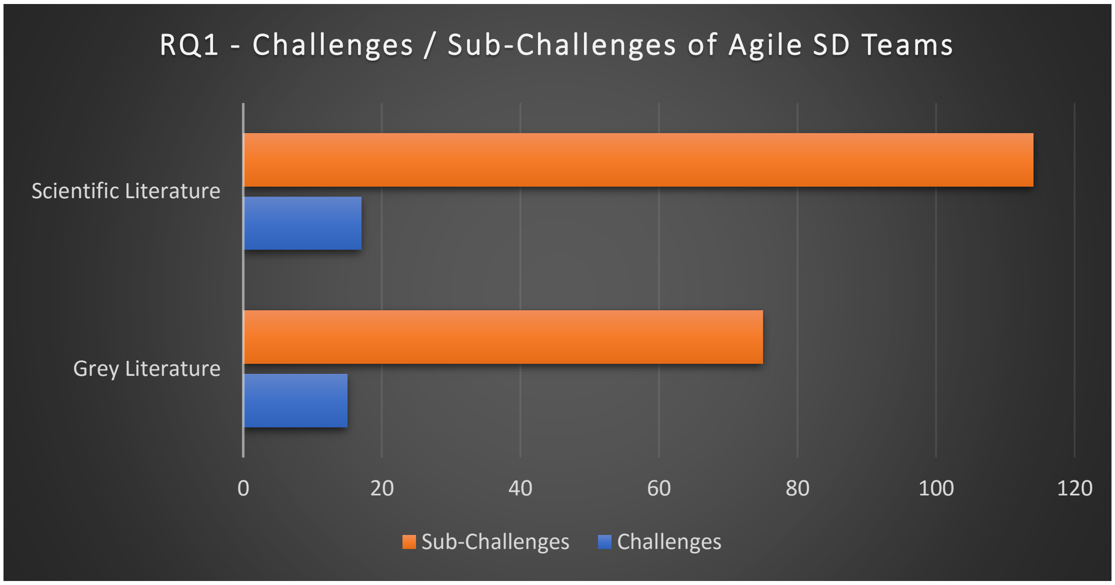


Figure 4.1: RQ1 - Challenges / Sub- Challenges of Agile SD Teams

4.4 Research Question 2 – Best and widely accepted Agile SD Methodology (GL)

It has been found from GL sources, authors / publishers or practitioners have ranked Agile methodologies based on different parameters (considering Agile SD methodology's pros and cons). Based on their information and reasons, I have listed Agile SD methodologies in Table 4.4 from Rank 1 (most widely used and accepted) to Rank 5. Same has been depicted in Figure 4.2.

4.5 Research Question 2 – Best and widely accepted Agile SD Methodology (SL)

It has been found from SL sources, researchers have ranked Agile methodologies based on different parameters (considering Agile SD methodology's pros and cons). Based on their information and reasons, I have listed Agile SD methodologies in Table 4.5 from Rank 1 (most widely used and accepted) to Rank 5. Same has been depicted in Figure 4.3.

Table 4.4 : RQ2 (GL) - Best Agile SD Methodology

	Agile SD Methodology (No of GL Sources)
Rank 1	Scrum (20) Kanban (4) XP (2)
Rank 2	Kanban (11) Scrum (5) XP (3) Lean (3) Scrumban (1)
Rank 3	XP (7) Lean (5) Scrumban (3) Kanban (2) Crystal (1) Feature Driven Development - FDD (1) Scrum-XP Hybrid (1)
Rank 4	XP (4) Crystal (3) Dynamic System Development Method - DSDM (2) Feature Driven Development – FDD (2) DevOps (1) Kanban (1) Lean Development (1) Scrum (1)
Rank 5	Crystal (5) DSDM (3) FDD (2) XP (2) Adaptive SD – ASD (1) Kanban (1)

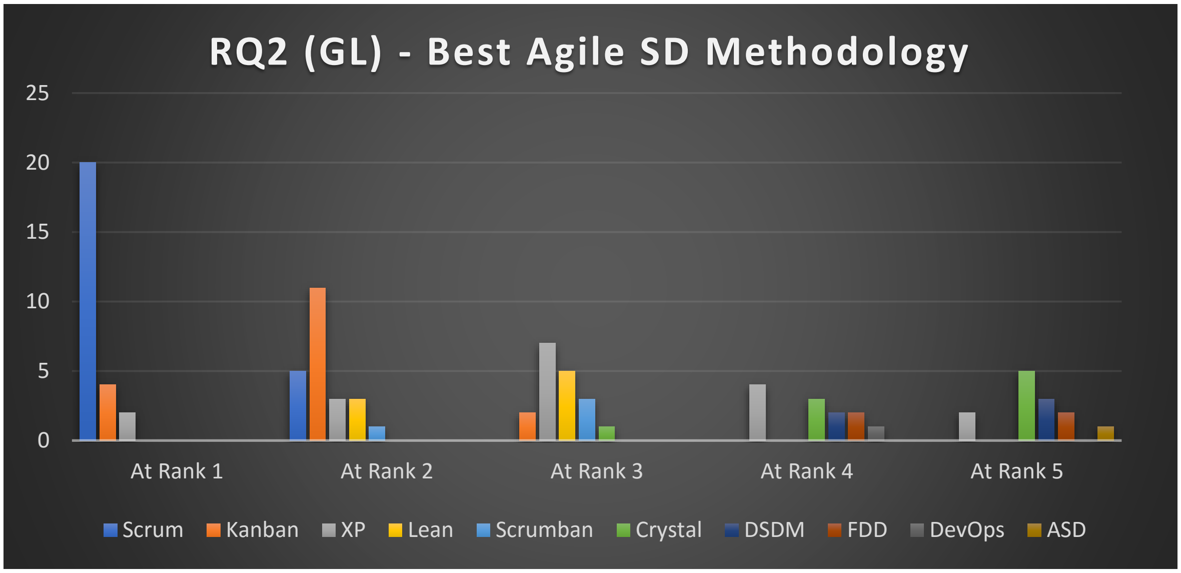


Figure 4.2 : RQ2 (GL) - Best Agile SD Methodology

Table 4.5 : RQ2 (SL) - Best Agile SD Methodology

	Agile SD Methodology (No of SL Sources)
Rank 1	Scrum (13) XP (1)
Rank 2	XP (6) Lean (1) Scrum (1) Scrumban (1)
Rank 3	Crystal (1) DSDM (1) Kanban (1) Scrumban (1)
Rank 4	DSDM (1) Scrum XP Hybrid (1)
Rank 5	Lean (1)

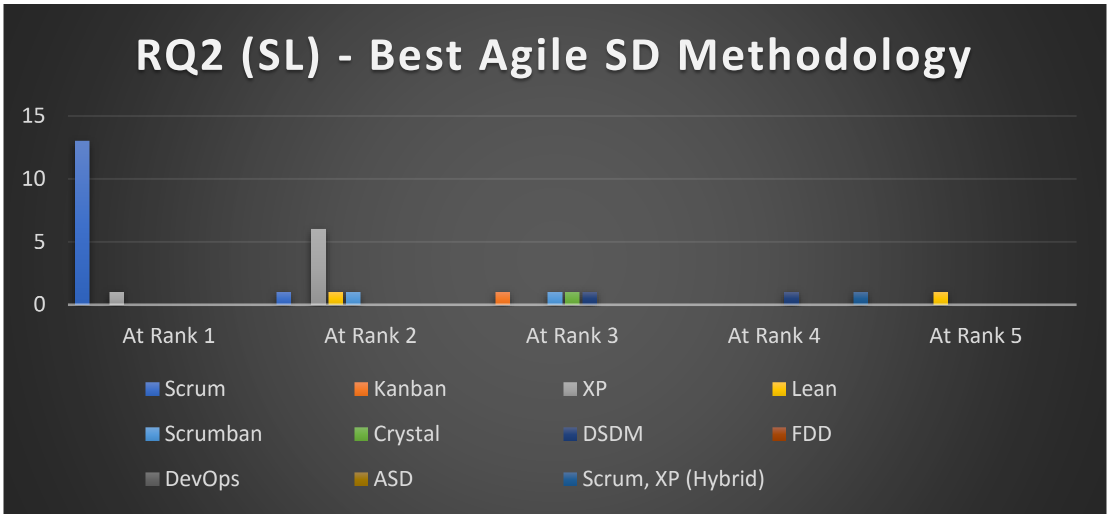


Figure 4.3 : RQ2 (SL) - Best Agile SD Methodology

4.6 Research Question 3 – Practices and Benefits of Management 3.0 Framework that can help issues resolution of Agile SD Teams

Research work looks into the ever-changing world of Agile SD Teams, where good management is key to success. Within this framework, Management 3.0 shows up as an all-encompassing set of practices designed to improve teamwork, communication, and personnel health. Each part of Agile teams, from new ideas like Delegation Poker and Kudo Cards to useful tools like Personal Maps and 360-degree feedback, helps to create a happy and cooperative atmosphere. The perks include happy employees, building trust, flexible leadership, and good conflict settlement. This study does, however, recognize that the success of these practices depends on how well they work with the specific needs of Agile SD Teams. Through a combination of SL and GL, research aims to give a more complete picture of how Management 3.0 Framework can be used in real life to help Agile teams deal with challenges / problems.

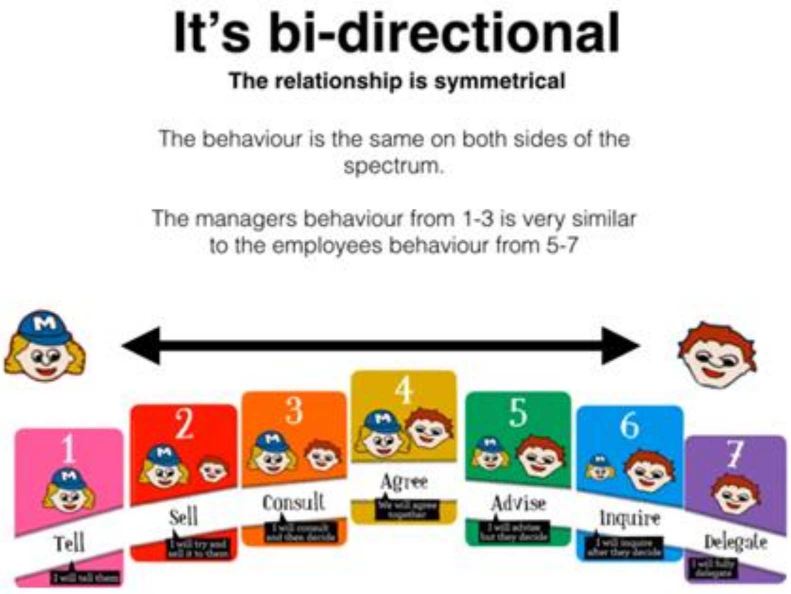
These practices are elaborated below to provide an understanding of each strategy and instrument as they pertain to Agile SD Team.

4.6.1 Practices of Management 3.0 Framework [57]

Management 3.0 Framework by Jurgen Appelo is employees' focused framework that helps to improve organization environment, employees' productivity, satisfaction level. [60]

4.6.1.1 Delegation Poker

Delegation Poker and Delegation Board (shown in Figure 4.4 and Figure 4.5) are interesting practices to clarify who is responsible for specific task and to what extent / level. Objective of Delegation Poker is an approach to delegate decisions and tasks within team within a controlled environment. It promotes open talks and responsibility alignment. [61]



It's bi-directional

The relationship is symmetrical

The behaviour is the same on both sides of the spectrum.

The managers behaviour from 1-3 is very similar to the employees behaviour from 5-7

1
Tell
I will tell them
2
Sell
I will try and sell it to them
3
Consult
I will consult and then decide
4
Agree
We will agree together
5
Advise
I will advise but they decide
6
Inquire
I will inquire after they decide
7
Delegate
I will fully delegate

Figure 4.4 : Delegation Poker

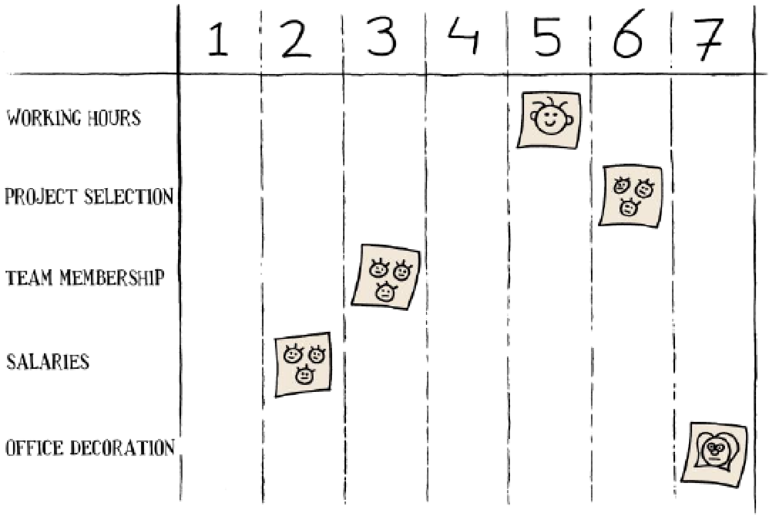


Figure 4.5 : Delegation Board

4.6.1.2 Kudo Cards

Kudo Cards can be used to convey thanks and appreciation for the work of team members. Kudo Cards help to maintain a good and supportive team culture.

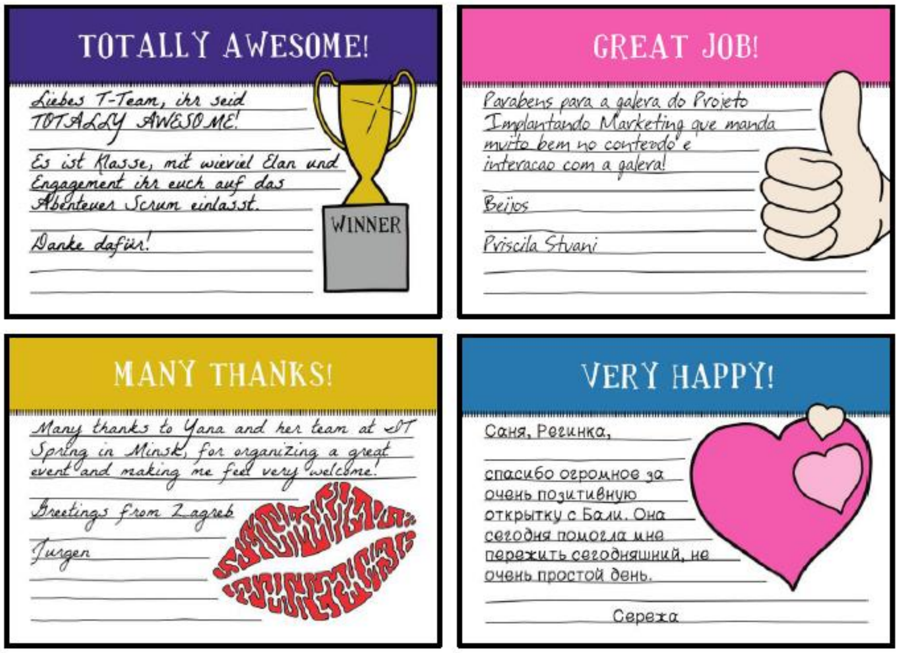

TOTALLY AWESOME!
Liebes T-Team, ihr seid TOTALLY AWESOME!
Es ist Klasse, mit wieviel Elan und Engagement ihr euch auf das Abenteuer Scrum einlasst.
Danke dafür!
WINNER
GREAT JOB!
Parabens para a galera do Projeto Implantando Marketing que manda muito bem no conteudo e interacao com a galera!
Beijos
Priscila Stvani
MANY THANKS!
Many thanks to Yana and her team at IT Spring in Minsk, for organizing a great event and making me feel very welcome!
Greetings from Zagreb
Jurgen
VERY HAPPY!
Саня, Регинка,
спасибо огромное за очень позитивную открытку с Бали. Она сегодня помогла мне пережить сегодняшний, не очень простой день.
Сережа

Figure 4.6 : Kudo Cards

4.6.1.3 Celebration Grid

Celebration Grids of Management 3.0 are useful tool / practice to acknowledge and celebrate team accomplishments. It promotes a culture of achievement recognition and appreciating employees. Celebration Grid shown in Figure 4.7.

4.6.1.4 Personal Maps

Personal Maps assist team members in understanding each other's preferences, strengths, and working styles. Personal Maps promotes improved team communication and cooperation. Personal Maps shown in Figure 4.8.

4.6.1.5 Moving Motivators

Another interesting practice of Management 3.0 Framework is Moving Motivators. It helps management to know what motivates your colleagues? Hiring a new person and know about his/ her motivation factor, Moving Motivators help in this regard. Moving Motivators shown in Figure 4.9.

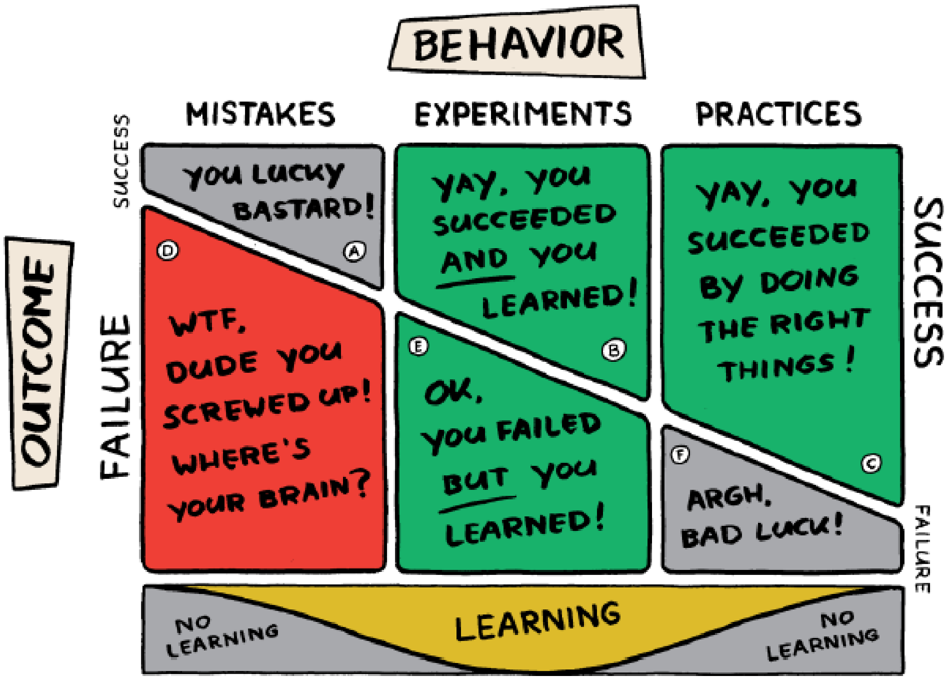

BEHAVIOR
MISTAKES
EXPERIMENTS
PRACTICES
OUTCOME
SUCCESS
FAILURE
SUCCESS
FAILURE
you lucky BASTARD!
A
YAY, you SUCCEEDED AND you LEARNED!
B
YAY, you SUCCEEDED BY DOING THE RIGHT THINGS!
C
D
WTF, DUDE you SCREWED UP! WHERE'S YOUR BRAIN?
E
OK, you FAILED BUT you LEARNED!
F
ARGH, BAD LUCK!
NO LEARNING
LEARNING
NO LEARNING

Figure 4.7 : Celebration Grid

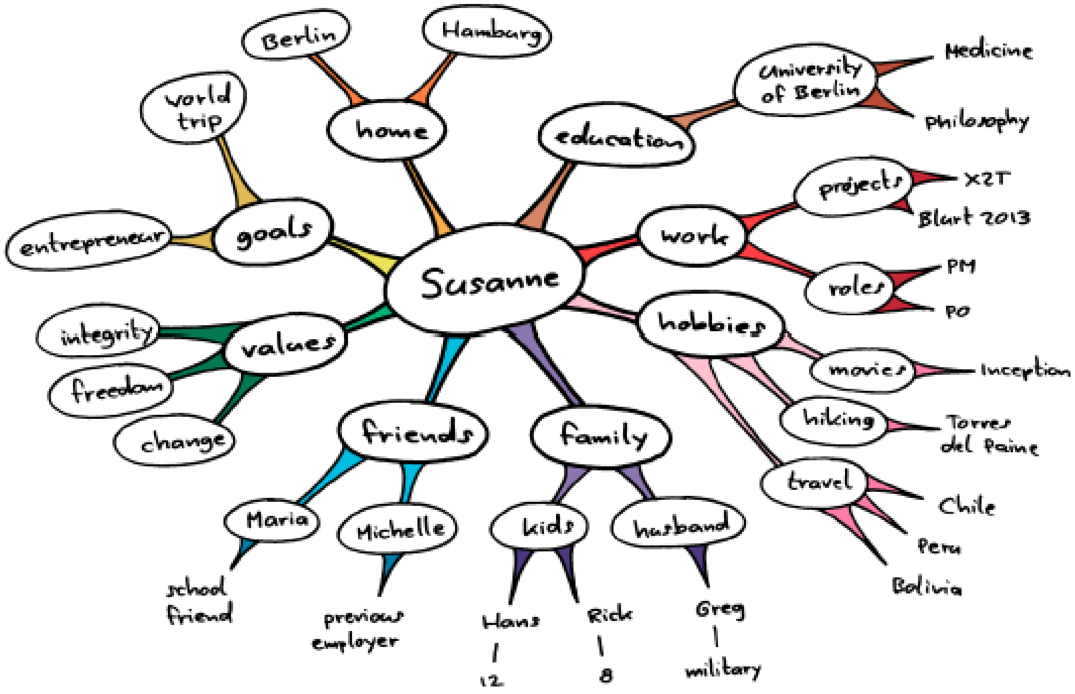
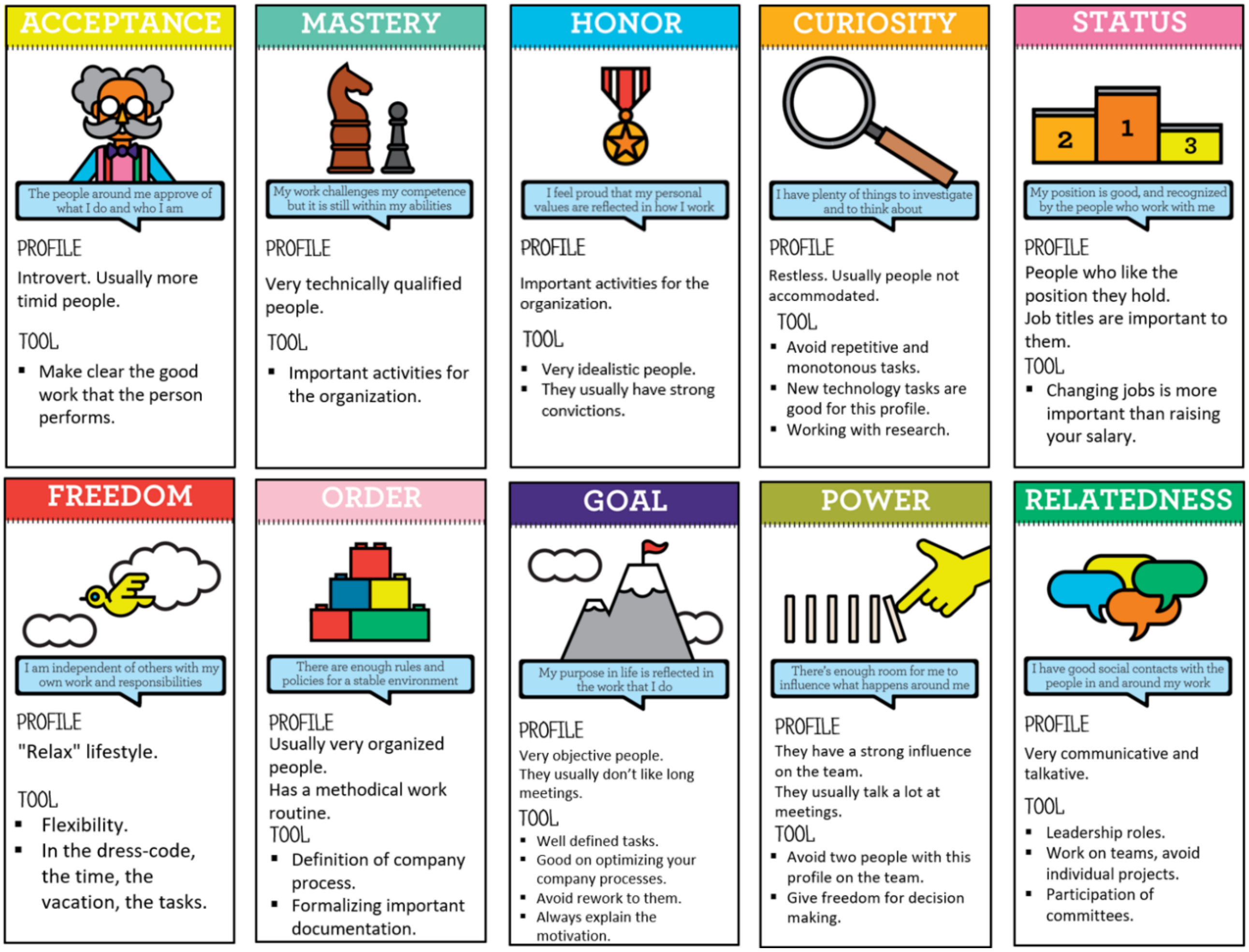


Figure 4.8 : Personal Maps



ACCEPTANCE

The people around me approve of what I do and who I am

PROFILE

Introvert. Usually more timid people.

TOOL

- Make clear the good work that the person performs.

MASTERY

My work challenges my competence but it is still within my abilities

PROFILE

Very technically qualified people.

TOOL

- Important activities for the organization.

HONOR

I feel proud that my personal values are reflected in how I work

PROFILE

Important activities for the organization.

TOOL

- Very idealistic people.
- They usually have strong convictions.

CURIOSITY

I have plenty of things to investigate and to think about

PROFILE

Restless. Usually people not accommodated.

TOOL

- Avoid repetitive and monotonous tasks.
- New technology tasks are good for this profile.
- Working with research.

STATUS

My position is good, and recognized by the people who work with me

PROFILE

People who like the position they hold.
Job titles are important to them.

TOOL

- Changing jobs is more important than raising your salary.

FREEDOM

I am independent of others with my own work and responsibilities

PROFILE

"Relax" lifestyle.

TOOL

- Flexibility.
- In the dress-code, the time, the vacation, the tasks.

ORDER

There are enough rules and policies for a stable environment

PROFILE

Usually very organized people.
Has a methodical work routine.

TOOL

- Definition of company process.
- Formalizing important documentation.

GOAL

My purpose in life is reflected in the work that I do

PROFILE

Very objective people.
They usually don't like long meetings.

TOOL

- Well defined tasks.
- Good on optimizing your company processes.
- Avoid rework to them.
- Always explain the motivation.

POWER

There's enough room for me to influence what happens around me

PROFILE

They have a strong influence on the team.
They usually talk a lot at meetings.

TOOL

- Avoid two people with this profile on the team.
- Give freedom for decision making.

RELATEDNESS

I have good social contacts with the people in and around my work

PROFILE

Very communicative and talkative.

TOOL

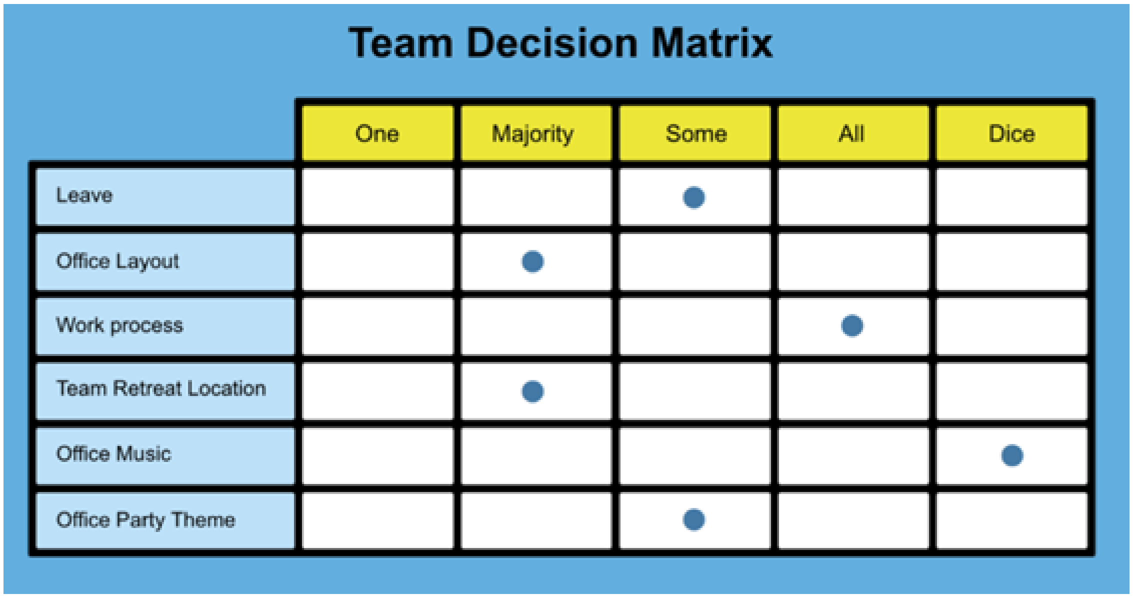
- Leadership roles.
- Work on teams, avoid individual projects.
- Participation of committees.

Figure 4.9 : Moving Motivators

4.6.1.6 Team Decision Matrix

When there is no manager, Delegation Board cannot be used. In this situation, Team Decision Matrix can be used. Key decision areas are finalized followed by a set of Team Decision Cards. Everyone to choose card for decision area and on count of three, everyone to show their card. It helps to discuss differences and decide mutually within team.

- Team Decision Matrix shown in Figure 4.10
- Team Decision Cards shown in Figure 4.11



Team Decision Matrix

	One	Majority	Some	All	Dice
Leave			●		
Office Layout		●			
Work process				●	
Team Retreat Location		●			
Office Music					●
Office Party Theme			●		

Figure 4.10 : Team Decision Matrix

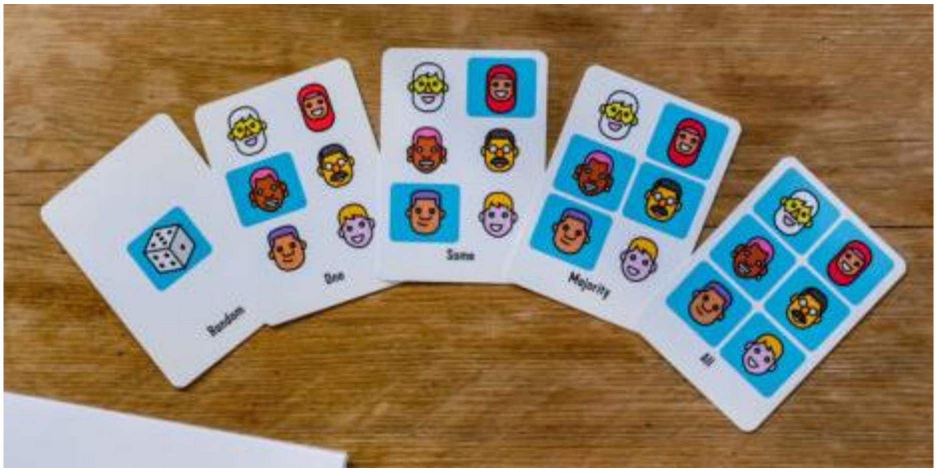


Figure 4.11 : Team Decision Cards

4.6.2 Benefits of Management 3.0 Framework [57]

Management 3.0 framework offers a lot of benefits for employees and eventually to an organization in a broader perspective. Management 3.0 focuses on employees' satisfaction, empowering teams, energize people, improving system as a whole.

4.6.2.1 Enhances Employee Satisfaction

- Management 3.0 techniques help to increase employees' satisfaction and happiness within Agile teams

4.6.2.2 Team Collaboration and Trust

- Practices such as Kudo Cards and Personal Maps facilitate building up and strengthening trust. Management 3.0 focuses to increase cooperation among team members

4.6.2.3 Adaptive Framework

- The framework encourages adaptive leadership, enabling leaders to react flexibly to changing demands and addressing problems of Agile projects/ teams

4.6.2.4 Positive Work Culture

- Positive Work Culture 3.0 encourages gratitude, open communication, and an emphasis on individual well-being to build a healthy work culture

4.6.2.5 Motivated and Engaged Teams

- Practices such as Delegation Poker and Celebration Grids help to increased levels of motivation and engagement among team members

4.6.2.6 Effective Conflict Resolution

- The framework promotes successful dispute resolution by promoting open communication and positive feedback methods

4.6.2.7 Continuous Improvement

- Management 3.0 fosters a culture of continuous development within the team by implementing methods such as 360-degree feedback and Feedback Wraps

4.6.2.8 Agile Team Empowerment

- Empowering teams via shared decision-making (as demonstrated in Delegation Poker) increases team autonomy and responsibility

4.6.2.9 Enhanced Communication

- Tools such as Personal Maps and Feedback Wraps increase communication, ensuring that team members understand each other and can handle concerns efficiently

It is highlighted that success of these methods vary depending upon the setting and the unique demands of Agile SD Teams. Organizations often customize Management 3.0 methods according to their own circumstances, benefitting from framework's flexibility and adaptability.

Through the examination and juxtaposition of scientific and grey literature about the practices and advantages of Management 3.0 Framework for Agile SD Teams, a synthesis of factual data and practical implementation has become apparent. SL provides a basis by demonstrating the rigorous methodology and supported findings about the effects of Management 3.0 practices. Assessments of techniques, advantages, and limitations of researches provide a detailed comprehension of the practices and advantages backed by empirical data. In contrast, grey literature, which includes industry reports, case studies, and practical stories, provide a contextualized perspective on the implementation of the framework. An investigation of new trends and creative practices inside businesses enhances the analysis, providing insight into the dynamic character of Management 3.0 in tackling difficulties encountered by Agile teams. By including various viewpoints, the study not only confirms theoretical concepts but also reveals the practical flexibility and contextual intricacies that contribute to the thorough comprehension of Management 3.0's effectiveness in Agile SD environments. This integrated methodology provides a comprehensive perspective of theoretical principles presented by researchers and practical scenarios faced by Agile SD Teams, so enhancing a more nuanced and resilient understanding of the subject matter.

4.7 Summary

In this chapter, issues/ challenges faced by Agile SD Teams have been identified from GL and SL (RQ1). Same Challenges have been broadly defined as Themes and Sub-challenges have been grouped under Challenges / Themes. For RQ2, best and widely accepted Agile SD methodology has been identified from GL and SL. Sources for RQ3 highlights practices and benefits of Management 3.0 Framework that helps resolution of Agile SD Teams issues and challenges. Practices of Management 3.0 Framework have been highlighted, which help in improving teams' performance, increase efficiency of employees, building trust, conducive environment among and for employees.

Chapter 5

Conclusion

5.1 Overview

The study delved into the ever-changing environment of Agile Software Development (SD) and its crucial significance in today's business landscape. Agile approaches, characterized by their iterative and stakeholder-engaging approach, serve efficient for software development, transforming established paradigms and improving stakeholders' satisfaction. Nevertheless, due to fundamental need for smooth flow of information, high demand of qualitative output, communication/ coordination across SD teams, conducive work environment, interfacing with client, issues invariably emerged.

5.2 Contributions

The main objective of this research was to thoroughly identify and resolve the complex problems faced by Agile SD teams. The study stood out due to its skillful use of Jurgen Appelo's Management 3.0 Framework [57], providing a sophisticated and effective approach to navigating the inherent challenges of Agile teams. The study aimed to propose use of Management 3.0 Framework and its practices for resolution of Agile SD Teams issues and challenges. Same framework may also be used to provide solutions towards improvement among team members (interaction, trust, relationships, bonding, communication/ coordination). Self-organizing Teams are important constituent of Agile SD methodology. In this regard, Team Decision Matrix, Delegation Poker of Management 3.0 Framework helps overcoming challenges for self-organizing teams.

Implementing a Multivocal Literature Review (MLR), which includes both Grey and Scientific Literature, enhanced the comprehensiveness and scope of the investigation. Research Methodology entailed three phases; identifying issues/ problems of Agile SD Teams, conducting a detailed analysis of Agile approaches and finding best and widely accepted Agile SD methodology, and applying Management 3.0 principles. This approach was holistic and thorough, providing a consolidated comprehension of the difficulties encountered by Agile SD Teams and offering solutions and practices for resolving them.

In addition, the use of Thematic Analysis improved results presentation / deduction by identifying significant patterns and themes from a wide range of literature sources. Thematic analysis is performed not only to highlight results from SL, but also emphasized the significance and practicality of the recommended solutions for real-time Agile SD.

The study went beyond conventional Agile SD teams' issues identification but provides an insight into practical solutions, which aligns with the constantly changing environment of Agile SD. The goal was to provide an insight into the practical issues faced by software industry personnel as well as researches already performed. Practitioners, academia and organizational aspects of Agile SD have been taken into consideration and it has been achieved by combining theoretical insights as well as considering practical scenarios.

5.3 Limitations

Research is an ongoing process. For my research, SL from Year 2015 - 2022 has been studied and reviewed. New and updated literature after Aug 2023 has not been taken into an account.

5.4 Future Work

SL after Aug 2023 may be utilized for consideration and incorporation of latest literature sources. Agile SD teams' productivity and efficiency may be assessed before and after introduction of Management 3.0 Framework through questionnaires, feedback from employees and management.

References

[1]. Kumar, B. A., & Chand, S. S. (2018). *Mobile learning adoption: A systematic review. Education and Information Technologies.*

[2]. David Budgen & Pearl Brereton, (2006). Performing Systematic Literature Reviews in Software Engineering (ICSE'06, May 20–28, 2006, Shanghai, China. ACM 1-59593-085-X/06/0005.)

[3]. Kitchenham, B. (2021). Procedures for performing systematic reviews (Keele University Technical Report No. TR/SE-0401; NICTA Technical Report No. 0400011T. 1). Keele University, Keele (2004).

[4]. Costa, L. A., Salvador, L. N., & Amorim, R. R. (2018, October). Evaluation of academic performance based on learning analytics and ontology: a systematic mapping study. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.

[5]. https://en.wikipedia.org/wiki/Problem_statement

[6]. Peerasit Patanakul and Reneé Rufo-McCarron, "Transitioning to agile software development: Lessons learned from a government-contracted program," Journal of High Technology Management Research, vol. 29, pp. 181-192, 2018.

[7]. Kim Dikert, Maria Paasivaara, Casper Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," The Journal of Systems and Software, vol 119, pp 87-108, 2016.

[8]. Fernando Almeida, & Eduardo Espinheira, (2022). *Adoption of Large-Scale Scrum Practices through the Use of Management 3.0.*

[9]. Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, *106*, 101-121.

[10]. R. J. Adams, P. Smart, and A. S. Huff, ″Shades of Grey: Guidelines for Working with the Grey Literature in Systematic Reviews for Management and Organizational Studies,″ International Journal of Management Reviews, pp. n/a- n/a, 2016.

[11]. E. Tom, A. Aurum, and R. Vidgen, ″An exploration of technical debt,″ Journal of Systems and Software, vol. 86, no. 6, pp. 1498- 1516, 2013.

[12]. V. Garousi and M. V. Mäntylä, ″When and what to automate in software testing? A multivocal literature review,″ Information and Software Technology, vol. 76, pp. 92-117, 2016.

[13]. Priyanka Antil, "Protocol for a multi-vocal literature review of requirements engineering in global scaled agile software development environment", https://arxiv.org/ftp/arxiv/papers/2004/2004.12647.pdf

[14]. Neto, G. T. G., Santos, W. B., Endo, P. T., & Roberta Fagundes, A. A. (2019). Multivocal literature reviews in software engineering: Preliminary findings from a tertiary study. 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)

[15]. Garousi, V., Felderer, M., & Hacaloğlu, T. (2017). Software test maturity assessment and test process improvement: A multivocal literature review. Information and Software Technology, 85, 16–42.

[16]. https://upcommons.upc.edu/bitstream/handle/2117/355581/Costal%20et%20al.pdf;jsessionid=4C2D5ACCFA63973B330D25EC1BAACB25?sequence=1

[17]. B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, ″Systematic literature reviews in software engineering–a systematic literature review,″ Information and software technology, vol. 51, no. 1, pp. 7- 15, 200

[18]. K. Petersen, S. Vakkalanka, and L. Kuzniarz, ″Guidelines for conducting systematic mapping studies in software engineering: An update,″ Information and Software Technology, vol. 64, pp. 1- 18, 2015

[19]. P.- M. Daigneault, S. Jacob, and M. Ouimet, ″Using systematic review methods within a Ph.D. dissertation in political science: challenges and lessons learned from practice,″ International Journal of Social Research Methodology, vol. 17, no. 3, pp. 267-283, 2014

[20]. V. Garousi, Y. Amannejad, and A. Betin- Can, ″Software test- code engineering: a systematic mapping,″ Journal of Information and Software Technology, vol. 58, pp. 123–147, 2015

[21]. C. Olsson, A. Ringnér, and G. Borglin, ″Including systematic reviews in PhD programmes and candidatures in nursing – 'Hobson′s choice'?,″ Nurse Education in Practice, vol. 14, no. 2, pp. 102- 105, 2014

[22]. R. L. Glass and T. DeMarco, Software Creativity 2.0. developer.* Books, 2006

[23]. R. T. Ogawa and B. Malen, ″Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method,″ Review of Educational Research, vol. 61, no. 3, pp. 265- 286, 1991

[24]. M. Q. Patton, ″Towards Utility in Reviews of Multivocal Literatures,″ Review of Educational Research, vol. 61, no. 3, pp. 287- 292, 1991

[25]. E. Tom, A. Aurum, and R. Vidgen, ″An exploration of technical debt,″ Journal of Systems and Software, vol. 86, no. 6, pp. 1498- 1516, 2013

[26]. B. Kitchenham and S. Charters, ″Guidelines for Performing Systematic Literature Reviews in Software engineering,″ in ″EBSE Technical Report,″ 2007, vol. EBSE- 2007- 01

[27]. S. Easterbrook, J. Singer, M.- A. Storey, and D. Damian, ″Selecting Empirical Methods for Software Engineering Research,″ in Guide to Advanced Empirical Software Engineering, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer London, 2008, pp. 285- 311

[28]. J . Adams et al., ″Searching and synthesising ‘grey literature’ and ‘grey information’ in public health: critical reflections on three case studies,″ Systematic Reviews, journal article vol. 5, no. 1, p. 164, 2016

[29]. Keele, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Vol. 5). Technical report, ver. 2.3 ebse technical report. ebse

[30]. da Silva, F. Q., Costa, C., Franca, A. C. C., & Prikladinicki, R. (2010, August). Challenges and solutions in distributed software development project management: A systematic literature review. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 87-96). IEEE

[31]. Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9-10), 833-859

[32]. Sommerville, I. (2011). Software Engineering (9th ed.). Addison-Wesle

[33]. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. VTT Technical Research Centre of Finland

[34]. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. Communications of the ACM, 48(5), 72-78

[35]. Ambler, S. W. (2009). Introduction to Agile Modeling. AgileModeling.com

[36]. Serrador, P., & Pinto, J. K. (2015). Does agile work?—A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040-1051

[37]. Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software, 119, 87-108

[38]. Smith, A. (2023). Agile Challenges: Navigating Communication Breakdowns and Scope Changes. Journal of Agile Practices, 15(3), 112-130

[39]. Jones, B., et al. (2024). Navigating Resistance in Agile Adoption: A Comprehensive Review. Journal of Software Engineering, 28(1), 45-62

[40]. Gupta, R. (2023). Real-world Challenges in Agile Software Development: Insights from Industry Reports. Agile Journal, 21(2), 78-94

[41]. Mitchell, L., & Chang, S. (2024). External Pressures and Adaptability in Agile Teams: A Case Study Approach. International Conference on Agile Practices Proceedings, 245-250

[42]. Patel, S. (2023). Organizational Culture and Agile Success: A Case Study Analysis. Journal of Agile Management, 22(4), 210-225

[43]. Strode, D. (2014). The new project management paradigm: Agile management for complex projects. International Journal of Project Management, 32(8), 1391-1401

[44]. Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology, 50(9-10), 833-859

[45]. Hoda, R., Noble, J., & Marshall, S. (2019). Learning in Agile software development: A systematic literature review. Information and Software Technology, 105, 41-59

[46]. Appelo, J. (2011). Management 3.0: Leading Agile Developers, Developing Agile Leaders. Addison-Wesley

[47]. Rodrigues, R., & Ferreira, M. (2020). Enhancing agile teams with management 3.0 practices. Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE), 245-250

[48]. Management 3.0. (2020). Case Studies. Retrieved from https://management30.com/case-studies/

[49]. Jones, B., & Smith, C. (2023). Empowering Agile Teams: A Study on the Impact of Management 3.0. Journal of Agile Management, 19(3), 112-130

[50]. Patel, S., et al. (2024). Motivation and Success: The Influence of Management 3.0 in Agile Environments. Agile Journal, 20(4), 78-9

[51]. Gupta, R., & Kim, Y. (2023). Fostering Innovation in Agile Teams: A Management 3.0 Perspective. Proceedings of the International Conference on Agile Practices, 210-225

[52]. Chang, S., & Wang, L. (2024). Collaboration and Conflict Resolution in Agile Teams: The Role of Management 3.0. International Journal of Agile Management, 23(1), 45-6

[53]. Brown, R., et al. (2023). Scaling Management 3.0 in Large Organizations: Challenges and Opportunities. Journal of Software Engineering, 27(3), 112-130

[54]. Wang, L., et al. (2023). Management 3.0 and Continuous Improvement: A Case Study Analysis. Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE), 245-250

[55]. Kim, Y., & Patel, S. (2024). The Role of Management 3.0 in Project Success: A Longitudinal Analysis. Journal of Agile Management, 22(4), 210-225

[56]. Mitchell, L. (2023). Advancements in Management 3.0: A Comprehensive Review. Agile Journal, 21(2), 78-94

[57]. Appelo J (2014) Management 3.0 Workout. Rotterdam, The Netherlands: Happy Melly Express

[58]. https://www.scribbr.com/methodology/thematic-analysis/

[59]. https://youtu.be/rvMf1cbctYM?si=elQHESPGgbeUXr2v

[60]. https://www.belithe.com/what-is-management-3-0/

[61]. https://management30.com/practice/delegation-poker/

Anx A

GL sources after applying 2nd Tier Inclusion / Exclusion Protocol have been listed as follows:

Source ID	**Title**
Research Question 1	
RQ1-GL-1	The Massive Downside of Agile Software Development. (Every good comes with a bad. Here's why agile software development won't solve all your problems)
RQ1-GL-2	Agile Software Development
RQ1-GL-3	The Problems with Agile and Scrum
RQ1-GL-4	Difference between Traditional and Agile Software Development
RQ1-GL-5	The 5 Most Common Problems In Agile Software Development
RQ1-GL-6	Some problems in Agile software development practices
RQ1-GL-7	Agile software development
RQ1-GL-8	Problems in applying Agile software development model and solution proposal.
RQ1-GL-9	Agile project management: 10 challenges you'll face (page 1) Agile project management: 10 challenges you'll face (page 2)
RQ1-GL-10	Agile tools for software teams
RQ1-GL-11	Extreme Programming: Agile Software Development
RQ1-GL-12	Disadvantages of Agile
RQ1-GL-13	Agile Essentials
RQ1-GL-14	How Agile Software Development Can Improve Your Client's Experience
RQ1-GL-15	WHAT IS AGILE? WHAT IS SCRUM?
RQ1-GL-16	WHY COMMUNICATION IS A DRIVER OF AGILE PROJECT SUCCESS
RQ1-GL-17	Agile Software Development
RQ1-GL-18	Agile Software Development Lifecycle Phases Explained
RQ1-GL-19	Identifying Solutions for Customer – Supplier Communication Problems in Agile Software Projects
RQ1-GL-20	The Agile Software Development Process Timeline

RQ1-GL-21	Characteristics of agile development success
RQ1-GL-22	What is Agile Software Development?
RQ1-GL-23	What is Agile Methodology, and Why Should You Embrace it for Software Development?
RQ1-GL-24	The Beauty and the Terror of Agile Software Development
RQ1-GL-25	Agile Software Guide
RQ1-GL-26	What Is Agile Software Development?
RQ1-GL-27	Agile Software Development
RQ1-GL-28	How to use GitLab for Agile software development
RQ1-GL-29	What are the disadvantages of Agile?
RQ1-GL-30	Agile Project Management: Best Practices and Methodologies
RQ1-GL-31	Agile project management: 12 key principles, 4 big hurdles
RQ1-GL-32	The Importance of Transparency In Agile Software Development
RQ1-GL-33	Implement an Agile Software Testing Process into Projects from the Start
RQ1-GL-34	Marrying User-Centered Design with the Agile Software Development Process: 7 Tips for Success
RQ1-GL-35	The Winter Getaway That Turned the Software World Upside Down
RQ1-GL-36	Agile Software Tools: 20 Best Solutions for Your Team in 2021
RQ1-GL-37	8 Agile Software Development Limitations You need to Know
RQ1-GL-38	What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models
RQ1-GL-39	Why agile teams should care about documentation
RQ1-GL-40	Is Agile always the best solution for software development projects?
RQ1-GL-41	Agile Software Development: Scrum & Kanban
RQ1-GL-42	Agile software development is dead. Deal with it
RQ1-GL-43	An Introduction to Agile Software Development
RQ1-GL-44	Accounting for Costs Incurred in the Application of Agile Software Development
RQ1-GL-45	Agile software requirements engineering: How to efficiently communicate, document and stay adaptable

RQ1-GL-46	How the Automotive Industry Can Benefit From Agile Software Development
Research Question 2	
RQ2-GL-1	Top 5 main Agile methodologies: advantages and disadvantages
RQ2-GL-2	What is Agile Methodology? How It Works, Best Practices, Tools
RQ2-GL-3	What are the Different Types of Agile Methodologies?
RQ2-GL-4	8 Important Types Of Agile Methodology (2022)
RQ2-GL-5	Limitations of Agile Methodologies
RQ2-GL-6	Top Agile methodologies you need to know
RQ2-GL-7	Comparison of Different Agile Methodologies: Pros and Cons
RQ2-GL-8	Agile Project Management: Best Practices and Methodologies
RQ2-GL-9	Advantages and Disadvantages of Agile Project Management
RQ2-GL-10	Top 5 agile software development methodologies
RQ2-GL-11	Different Agile Methodologies: Find Which One Fits Best Your Needs
RQ2-GL-12	What Is Agile Methodology? - Overview of Agile Software Development And Agile Models
RQ2-GL-13	Agile Software Development Methodologies: Which to Choose?
RQ2-GL-14	Agile Software Development
RQ2-GL-15	The Pros and Cons of Agile Methodologies
RQ2-GL-16	What is agile methodology? Modern software development explained
RQ2-GL-17	A Project Manager's Guide to 42 Agile Methodologies
RQ2-GL-18	7 Types of Agile Methodologies
RQ2-GL-19	Pros and Cons of Agile Software Development Methodologies
RQ2-GL-20	What is Agile? What is Scrum?
RQ2-GL-21	Agile 101
RQ2-GL-22	What is Agile: Understanding Agile Methodologies and Principles
RQ2-GL-23	Agile Methodologies: Kanban Vs Scrum - Advantages and Disadvantages
RQ2-GL-24	Everything you need to know about Agile methodologies
RQ2-GL-25	The Differences, Pros and Cons Between Waterfall and Agile Methodologies
RQ2-GL-26	The Agile Coach

RQ2-GL-27	What is Agile methodology? Examples, when to use, advantages & disadvantages
RQ2-GL-28	Traditional vs Agile Project Management Method: Which One is Right for Your Project?
RQ2-GL-29	Agile Best Practices for More Effective Teams
RQ2-GL-30	Learn about Scrum, Waterfall & Agile Methodologies
RQ2-GL-31	Agile vs. Scrum: How to Choose the Best Method
RQ2-GL-32	Traditional vs. Agile Software Development Methodologies
RQ2-GL-33	Agile Methodologies and Software Process Improvement Maturity Models, Current State of Practice in Small and Medium Enterprises
RQ2-GL-34	Agile software development
RQ2-GL-35	Difference between Traditional and Agile Software Development
RQ2-GL-36	Adopting Agile: The Latest Reports About The Popular Mindset
RQ2-GL-37	Visual guide to Agile methodologies for modern product management
RQ2-GL-38	Agile methodologies: Kanban Vs Scrum – Advantages and Disadvantages
RQ2-GL-39	Agile Methodologies | Scrum and Kanban: What are The Differences?
RQ2-GL-40	Agile Methodologies: Scrum and Kanban
RQ2-GL-41	5 Important Types Of Agile Methodology (2022)
RQ2-GL-42	What's the Difference? Agile vs Scrum vs Waterfall vs Kanban
RQ2-GL-43	DevOps vs. Agile Methodology: Key Differences and Similarities
RQ2-GL-44	A Guide To The Agile Method In Marketing And Development
RQ2-GL-45	Agile Methodology: An Overview
RQ2-GL-46	Agile vs. Waterfall vs. Kanban vs. Scrum: What’s the Difference?
RQ2-GL-47	What are the Reasons Behind Agile Popularity?
RQ2-GL-48	Non-Agile vs. Agile Methodologies: What’s the Difference?
RQ2-GL-49	Difference between agile and waterfall approaches to project management
RQ2-GL-50	The Role of the Business Analyst in an Agile SDM
RQ2-GL-51	Agile Methodology: What is Agile Model in Software Testing?
RQ2-GL-52	The Pros and Cons of Utilizing Agile Methodologies
RQ2-GL-53	Agile versus Scrum: What's the difference?

RQ2-GL-54	Half of companies applying Agile methodologies & practices
RQ2-GL-55	Agile vs Scrum vs Kanban: Weighing the Differences
RQ2-GL-56	The 9 Key Benefits of Using the Agile Methodology
RQ2-GL-57	Scrumban: The best of two Agile methodologies
RQ2-GL-58	Agile vs. Scrum: What's the Difference?
RQ2-GL-59	Agile Methodologies
RQ2-GL-60	What Is Agile Methodology, A Guide For Leaders
RQ2-GL-61	Agile Methodology
RQ2-GL-62	What is agile methodology?
RQ2-GL-63	Differences between Traditional and Agile Software Development Methodologies
RQ2-GL-64	Everything you need to know about agile project management
RQ2-GL-65	Waterfall vs. Agile methodologies: Which is best for project management?
RQ2-GL-66	Key KPIs Across Agile Methodologies
RQ2-GL-67	A Comparison between Agile and Traditional Software Development Methodologies
RQ2-GL-68	Agile vs Waterfall: Which Method is More Successful?
RQ2-GL-69	11 Good Learning Resources for Agile Certification
RQ2-GL-70	Agile Methodologies: What is the Agile Software Development Model?
RQ2-GL-71	To agility and beyond: The history—and legacy—of agile development
RQ2-GL-72	Embracing Agile (How to master the process that's transforming management)
RQ2-GL-73	What are the most common agile development methodologies?
RQ2-GL-74	Agile project management: 12 key principles, 4 big hurdles
Research Question 3	
RQ3-GL-1	Management 3.0 and Agile?
RQ3-GL-2	Agile project management methodologies 2022
RQ3-GL-3	What are the Different Types of Agile Methodologies?
RQ3-GL-4	Agile software development
RQ3-GL-5	What is Agile Software Development?

RQ3-GL-6	Agile Software Development
RQ3-GL-7	What Does Agile Mean – What Is Agile Methodology?
RQ3-GL-8	Agile Development
RQ3-GL-9	How to Combine the ITIL Best Practices with the Agile Mindset?
RQ3-GL-10	A Quantitative Study on Critical Success Factors in Agile Software Development Projects; Case Study IT Company
RQ3-GL-11	Safe 5 for Lean Enterprise
RQ3-GL-12	What is in SAFe 3.0 and how it can benefit an organization?
RQ3-GL-13	Understanding the Scaled Agile Framework: Principles, Benefits, and More
RQ3-GL-14	What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models
RQ3-GL-15	Tips for Making Your Agile Framework More … Agile
RQ3-GL-16	Adopting Agile Methodology: 6 Steps to Improved Delivery
RQ3-GL-17	Agile Values
RQ3-GL-18	Management 3.0 & Scrum: How to Become a Next Generation Agile Leader
RQ3-GL-19	Agile and government services: an introduction
RQ3-GL-20	Agile Project Management: Best Practices and Methodologies
RQ3-GL-21	What is agile methodology? Modern software development explained
RQ3-GL-22	Agile Vs. Agility. What Are The Differences?
RQ3-GL-23	Management 3.0 Practices Finder
RQ3-GL-24	Agile 101
RQ3-GL-25	What is Agile methodology? Examples, when to use, advantages & disadvantages
RQ3-GL-26	Management 3.0 and Agile?

Anx B

SL sources after applying 2nd Tier Inclusion / Exclusion Protocol have been listed as follows:

Source ID	**Title**
Research Question 1	
RQ1-SL-1	Issues, challenges, and a proposed theoretical core of agile software development research
RQ1-SL-2	Microservices in Agile Software Development: A Workshop Based Study into Issues, Advantages, and Disadvantages
RQ1-SL-3	Adopting Agile software development: the project manager experience
RQ1-SL-4	Agile transition and adoption human-related challenges and issues: A Grounded Theory Approach
RQ1-SL-5	Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review
RQ1-SL-6	A Mapping Study on Requirements Engineering in Agile Software Development
RQ1-SL-7	Agile adoption issues in large scale organizations: A review
RQ1-SL-8	Agile Software Development Methodologies: Survey of Surveys
RQ1-SL-9	When Agile Meets Waterfall - Investigating Risks and Problems on the Interface between Agile and Traditional Software Development in a Hybrid Development Organization
RQ1-SL-10	Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company
RQ1-SL-11	Perceived barriers to effective knowledgesharing in agile software teams
RQ1-SL-12	Investigating the Issues of Using Agile Methods in Offshore Software Development in Sri Lanka
RQ1-SL-13	Issues in Scrum Agile Development Principles and Practices in Software Development
RQ1-SL-14	Review on Agile requirements engineering challenges
RQ1-SL-15	A Review of Scaling Agile Methods in Large Software Development
RQ1-SL-16	Agile Retrospective Games for Different Team Development Phases

RQ1-SL-17	Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey
RQ1-SL-18	Agile Software Development: Methodologies and Trends
RQ1-SL-19	Towards Requirements Communication and Documentation Guidelines for Agile Teams
RQ1-SL-20	Satisfaction, Practices, and Influences in Agile Software Development
RQ1-SL-21	Moving from Waterfall to Agile: Perspectives from IT Portuguese Companies
RQ1-SL-22	Aligning Architecture Work with Agile Teams
RQ1-SL-23	A systematic literature review on agile requirements engineering practices and challenges
RQ1-SL-24	Success and Failure Factors that Impact on Project Implementation Using Agile Software Development Methodology
RQ1-SL-25	Moving from traditional to agile software development methodologies also on large, distributed projects
RQ1-SL-26	A conceptual model of agile software development in a safety-critical context: A systematic literature review
RQ1-SL-27	The impact of inadequate and dysfunctional training on Agile transformation process: A Grounded Theory study
RQ1-SL-28	Trends and Updated Research Agenda for Autonomous Agile Teams: A Summary of the Second International Workshop at XP2019
RQ1-SL-29	Effort Estimation in Agile Software Development: A Survey on the State of the Practice
RQ1-SL-30	Agile Challenges in Practice: A Thematic Analysis
RQ1-SL-31	An exploratory study in communication in Agile Global Software Development
RQ1-SL-32	An Adaptive Agile Process Model for Global Software Development
RQ1-SL-33	Discussion of Agile Software Development Methodology and its Relevance to Software Engineering
RQ1-SL-34	The Risks of Agile Software Development
RQ1-SL-35	Requirements engineering: A systematic mapping study in agile software development

RQ1-SL-36	Problems in Agile Global Software Engineering Projects especially within Traditionally Organised Corporations: [An exploratory semi-structured interview study]
RQ1-SL-37	Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process
RQ1-SL-38	Measuring the Impact of Agile Coaching on Students' Performance
RQ1-SL-39	Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson
RQ1-SL-40	Multi-level agile project management challenges: A self-organizing team perspective
RQ1-SL-41	Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks
RQ1-SL-42	Challenges in Migration from Waterfall to Agile Environments
Research Question 2	
RQ2-SL-1	Agile transition and adoption human-related challenges and issues: A Grounded Theory approach
RQ2-SL-2	Adapting the scrum framework for agile project management in science: case study of a distributed research initiative
RQ2-SL-3	Agile Software Development: Methodologies and Trends
RQ2-SL-4	The challenges that challenge: Engaging with agile practitioners' concerns
RQ2-SL-5	A Comparative Analysis of RAD and Agile Technique for Management of Computing Graduation Projects
RQ2-SL-6	A case study of agile software development for safety-Critical systems projects
RQ2-SL-7	Thoughts on Current and Future Research on Agile and Lean: Ensuring Relevance and Rigor
RQ2-SL-8	Use of software metrics in agile software development process
RQ2-SL-9	Interview Study on the Agile Development of Mechatronic Systems
RQ2-SL-10	Coordinating Knowledge Work in Multiteam Programs: Findings From a Large-Scale Agile Development Program

RQ2-SL-11	Becoming Agile while preserving software product lines: an Agile transformation model for large companies
RQ2-SL-12	Success Factors of Agile Information Systems Development: A Qualitative Study
RQ2-SL-13	Improved Product Development Performance through Agile/Stage-Gate Hybrids: The Next-Generation Stage-Gate Process?
RQ2-SL-14	The Role of Social Agile Practices for Direct and Indirect Communication in Information Systems Development Teams
RQ2-SL-15	Understanding the Roles of the Manager in Agile Project Management
RQ2-SL-16	Does a Hybrid Approach of Agile and Plan-Driven Methods Work Better for IT System Development Projects?
RQ2-SL-17	Identification of Agile Mechanisms of Action As Basis for Agile Product Development
RQ2-SL-18	The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development
Research Question 3	
RQ3-SL-1	Agile project management and stage-gate model—A hybrid framework for technology-based companies
RQ3-SL-2	Knowledge Management Strategies and Processes in Agile Software Development: A Systematic Literature Review
RQ3-SL-3	Trends in Agile: Perspectives from the Practitioners
RQ3-SL-4	Agile Software Development Using Cloud Computing: A Case Study
RQ3-SL-5	Holacracy and Obliquity: contingency management approaches in organizing companies
RQ3-SL-6	Measuring the Impact of Agile Coaching on Students’ Performance
RQ3-SL-7	The adoption of Software Engineering practices in a Scrum environment
RQ3-SL-8	A maturity model for the Integrated Agile Transformation Model $^{\text{TM}}$
RQ3-SL-9	A Comparative Study of Agile Methods, Testing Challenges, Solutions & Tool Support

RQ3-SL-10	Empirical comparison of traditional plan-based and agile methodologies: Critical success factors for outsourced software development projects from vendors’ perspective
RQ3-SL-11	Project Management Methods, Methodologies, and Frameworks: An Exploration for Study Guild for Project Management Practitioners of Ghana
RQ3-SL-12	Agile Project Management for Dummies
RQ3-SL-13	Scrum: An Effective Software Development Agile Tool
RQ3-SL-14	Toward Defining the Vibrant Concept of Agile Project Management
RQ3-SL-15	Project Management Methodologies in the Fourth Technological Revolution
RQ3-SL-16	Requirements engineering challenges and practices in large-scale agile system development
RQ3-SL-17	Implementation of Agile Methodologies in an Engineering Course
RQ3-SL-18	Requirements engineering: A systematic mapping study in agile software development
RQ3-SL-19	Towards the End of Agile: Owing to Common Misconceptions in the Minds of Agile Creators
RQ3-SL-20	How Agile Practices Impact Customer Responsiveness and Development Success: A Field Study