An Agile solution to address communication challenges during software development requirements engineering

A VENTER

A thesis submitted in partial fulfilment of the requirements for the degree
MASTERS (INDUSTRIAL ENGINEERING)
at the
FACULTY OF ENGINEERING, BUILT ENVIRONMENT, AND INFORMATION TECHNOLOGY
UNIVERSITY OF PRETORIA

SUPERVISOR: Prof. M de Vries

December 2021

© University of Pretoria

ABSTRACT

Action Design Research (ADR) continues to evolve to meet the demands of new and challenging environments due to ever-expanding applications. In this study, we gradually assemble multiple artefacts by using ADR. The *diagnosis* stage was the initial point of entry at a medium-sized enterprise in South Africa in the geographical information system (GIS) industry, referred to as Company-GIS or CGIS in this study. The researcher in collaboration with CGIS practitioners determined that *inadequate communication* while executing software development projects causes project tasks to take longer than expected, negatively impacting on-time delivery, quality of delivery, and delivery within budget. After conducting a systematic literature review (SLR) to determine whether *communication problems* exist as a class-of-problems in software development projects, the researcher determined that the root-cause identified, i.e., inadequate communication, was too vague and thus a *rudimentary taxonomy of perceived communication problems at CGIS* was created. The main communication problems identified *in the requirement elicitation/analysis phase* at CGIS are requirements that can't be defined and/or translated, and misalignment between stakeholders.

In the 15th State of Agile report, the most important reasons for adopting Agile within a team or organisation was to enhance the ability to manage changing priorities, accelerate software delivery, increase team productivity, and improve business and IT alignment. According to a 2020 survey conducted by McKinsey & Company, the COVID-19 crisis is a tipping point for technology adoption or digital disruption. Understanding which technologies to apply and how to manage change at a pace that far exceeds that of prior experiences, is critical going forward since the pace of change is not likely to slow down anytime soon. Considering that CGIS does not adhere to an Agile framework, and only incorporates some Agile practices, an experimentation opportunity existed.

This study answers the following primary research question: *What adaption of an Agile RE solution (or a subset of its associated mechanisms) could address a subset of classified communication challenges in software development companies (such as CGIS) to improve the information flow/communication between stakeholders during RE in order to assist management in reducing the misalignment between project stakeholders and/or the negative impact on project delivery?*

In collaboration with CGIS practitioners, the ADR *design* stage was initiated. The researcher created a proposed Agile solution which included Scrum to incrementally and frequently deliver software and therefore validate customer requirements, as well as the Requirements Specification for Developer (RSD) approach to document the requirements. The proposed solution was presented to CGIS practitioners and collaboratively modified.

The ADR *implementation* stage followed, during which the adapted Agile RE solution (ARES) was implemented and evaluated. Two sprints of one week each were evaluated and included in this study and it was determined that the ARES increased effective communication between stakeholders during RE at CGIS. Challenges were identified, including documentation and testing concerns as well as the idea of scaling the solution within CGIS. Recommendations for future research include validating the taxonomy presented, documenting the learnings obtained from the ADR methodology to showcase its usefulness, and incorporating DevOps into the ARES as part of the *evolution* stage of the ARES.

Keywords: Agile, communication, project management, requirements engineering, software development

© University of Pretoria

Table of contents

© University of Pretoria

© University of Pretoria

© University of Pretoria

List of tables

Table of figures

© University of Pretoria

Acronyms and abbreviations

Agile will be portrayed with a capital 'A' in this study because it is an abbreviation for 'Manifesto for Agile Software Development'

Acronym/abbreviation	Definition
.net	Managed computer software framework
AC	Acceptance criteria
ACM	Association for computing machinery
ADR	Action design research
AR	Action research
ARES	Agile requirement engineering solution
ASD	Agile software development
AT	Acceptance testing
ATDD	Acceptance test-driven development
Atlas.ti	The qualitative data analysis and research software
BDD	Behaviour driven development
BIE	Building, intervention, evaluation
CD	Cause-and-effect diagram
CGIS	A pseudonym for the real-world enterprise referred to in this study
CI	Continues integration
COVID	'CO' stands for corona, 'VI' for virus, and 'D' for disease
CRT	Current reality tree
DSD	Distributed software development
DSR	Design science research
GIS	Geographical information system
GSD	Global software development
ID	Interrelationship diagram
IEEE	Institute of electrical and electronic engineers
IET	Institute of engineering and technology
IGI	International academic publisher
IT	Informational technology
ITC/ICT	Information and communication technology
LeSS	Large-scale scrum

© University of Pretoria

Acronym/abbreviation	Definition
MIS	Managing information system
n.d.	No date
NFR	Non-functional requirement
OCLC	Online computer library centre
PB	Product backlog
PBIs	Product backlog items
PIECES	Framework classification divided into six categories: Performance, Information and Data, Economics, Control and Security, Efficiency, and Service
PLOS	Peer-reviewed open access scientific journal
PR	Primary researcher
QA	Quality assurance/assurer
RAC	Repository of acceptance criteria
RCA	Roof cause analyses
RE	Requirement engineering
REFSQ	Requirement engineering: Foundation for software quality
RSD	Requirements specification for developer
SAFe	The Scaled Agile Framework
SDLC	Software development life cycle
SIGSOFT	Special interest group of software engineering
SLR	Systematic literature review
SoS	Scrum of Scrums
SRS	Software requirements specification
TDD	Test driven development
UI	User interface design
UX	User experience design
XP	Extreme programming

© University of Pretoria

Preface

Firstly, this thesis applies active voice, rather than passive voice, as advised by Hofstee (2006) in his book, titled: Constructing a good dissertation.

Secondly, I would like to thank my supervisor, Prof. M de Vries, for her excellent guidance and support during this process. I would also like to thank all the participants without whose involvement I would not have been able to conduct this study.

Lastly, a part of this study has already been sent to a prestigious high-impact-factor journal. We are grateful for the insightful feedback provided by the anonymous reviewers that were useful in providing additional guidance in this study.

© University of Pretoria

Chapter 1– Introduction

Although efficient communication is paramount in software development projects due to the many stages and stakeholders within a knowledge-intensive environment, communication challenges are still evident in literature and real-life software development projects. Ensuring that appropriate information or knowledge is shared timely with the suitable stakeholder(s) is a challenge and could cause misalignment between the project stakeholders and/or have an impact on project delivery.

The objective of this study is to address a subset of classified communication challenges in software development companies (such as the real-life enterprise introduced in section 1.4) in order to reduce the level of misunderstanding and misalignment between project stakeholders during the requirements analysis phase of software development projects. The study should assist management in reducing the impact of the problems on project delivery. The study is presented in multiple chapters:

Chapter 1: Providing background on the research project, stating the research questions.
Chapter 2: Selecting an appropriate research methodology for the study.
Chapter 3: Identifying and validating a problem instance that also exists as a class-of-problems.
Chapter 4: Presenting a systematic literature review (SLR) about the problem.
Chapter 5: Presenting literature on Agile requirements engineering (RE) solutions and selecting an appropriate solution for experimentation at a real-world enterprise to address their communication-related problems.
Chapter 6: Demonstrating and evaluating the implementation of the Agile RE solution (ARES).
Chapter 7: Discussing the main contributions of the study.
Chapter 8: Concluding and presenting ideas for future research.

Three inter-linked knowledge areas relevant to this study are introduced in section 1.1, 1.2 and 1.3, i.e., *communication*, *requirement engineering (RE)*, and *Agile software development (ASD)*. Section 1.4 provides information on the particular enterprise in this study, i.e., CGIS. The problem statement is defined in section 1.5. The research questions are presented in section 1.6, whereas the thesis statement is presented in section 1.7. The delineations and limitations of the study are included in section 0 and section 1.9 argues the significance of this study.

1.1 Communication

Communication is the verbal or non-verbal, formal or informal act of exchanging knowledge or information (IGI Global, 2021). Effective communication consists of achieving a fully reciprocal understanding between people, which is a complicated process (Gode, 2012). Communication can take place horizontally, vertically, or diagonally, and can occur between stakeholders within the same department, from different departments, within an enterprise or between different enterprises. The communication between the project team and the customer, as well as the communication within the project team should be correct, relevant, and understood, and must reach the target audience in time (Muszyńska, 2018).

Several models have been suggested for describing and analysing the communication process, originating from the traditional sender/receiver model of communication and associated concepts (Pernstål, Feldt, Gorschek, & Florén, 2019). Although most models reference the elements of communication, sender, receiver, medium and message, there is no model that considers all aspects of communication as it would be too complex and detailed (Pernstål et al., 2019). Elements that are excluded, for example, are disturbances changing the sender's original message, i.e., noise, and the social situation in which the communication is taking place, i.e., context.

In the area of system or software development projects, communication means that different people agree to a common definition of what they are building, share information and mesh their activities (Kraut & Streeter, 1995). Software development projects are particularly susceptible to failure (Cerpa & Verner, 2009; Lehtinen, Mäntylä, Vanhanen, Itkonen, & Lassenius, 2014) and communication contributes to projects succeeding or failing (Cervone, 2014). Ineffective project and stakeholder communication could affect project expectations, estimations, schedules, and *requirements* (DeFranco & Laplante, 2017).

© University of Pretoria

1.2 Requirement Engineering (RE)

Software development consists of several interrelated processes. The process of collecting information from stakeholders and defining their needs and expectations in an understandable manner is referred to as requirement elicitation/analysis (Tiwari, Rathore, & Gupta, 2012). Requirement elicitation/analysis is a critical activity in software development projects (Davis, Dieste, Hickey, Juristo, & Moreno, 2006) as the information gathered during this process is used to construct the system, solution, or product to address the needs identified.

Requirement engineering (RE) is concerned with identifying, modelling, communicating, and documenting the requirements of a system and the context in which the system will be used (Paetsch, Eberlein, & Maurer, 2003). The use of the term "engineering" implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant (Curcio, Navarro, Malucelli, & Reinehr, 2018).

Software development requirements remain a challenge for software development projects specifically due to the complexity and involvement of multiple stakeholders (Coughlan, Lycett, & Macredie, 2003). Incomplete or ambiguous requirements are often provided (Aldave, Vara, Granada, & Marcos, 2019; Kerzner, 2014), articulation and/or language (whether native or business terminology) is often not understood by all stakeholders making it difficult to comprehensively understand the requirements (Aldave et al., 2019). Perspectives are different so there isn't consensus on what needs to be addressed, i.e. conflicting requirements, requirements evolve or are discarded as the project progresses (Aldave et al., 2019; Kerzner, 2014), the relevant stakeholders are often not available or involved (Aldave et al., 2019; Kerzner, 2014), and expectations are often unrealistic (Aldave et al., 2019; Kerzner, 2014).

1.3 Agile Software Development (ASD)

Agile emerged as a software methodology during the 1990s (Abbas, Gravell, & Wills, 2008) and the concept was formally defined in 2001 in The Agile Manifesto (Beedle et al., 2001). Based on the 12 values and 4 principles in the manifesto, Agile software development (ASD) is a set of frameworks, e.g. Scrum (Schwaber & Beedle, 2002) and Extreme Programming (Stott, 2003), and practices (e.g. stand-ups and sprints) that is focused on people and how they work together (Agile Alliance, n.d.)

Agile principles are often suggested to increase communication as there is a strong focus on different stakeholders, e.g. business and developer, working together daily and the sharing of project information through informal conversation rather than through documentation (Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008). ASD frameworks emphasise maintaining open communication channels and close communications between team members, including customers (Baham & Hirschheim, 2021), as this captures the Agile Manifesto's value of placing *individuals and interactions over processes and tools* (Beedle et al., 2001).

In the context of Agile methodologies, RE is carried out iteratively during the whole development process instead of during a closed phase in the beginning of the project (Schön, Thomaschewski, & JoEscalona, 2017) and relevant stakeholders work together in a collaborative manner. To this end, a just-in-time model is often used to refine high level requirements into low level tasks that can be implemented by developers.

The development of software at Company-GIS (CGIS), a medium-sized enterprise in South Africa that is the enterprise relevant to this study, currently follows an approach somewhere between waterfall and Agile (Grech, 2015) i.e., the company traditionally applies the waterfall method, and applies Agile practices where applicable, e.g., post-it boards. The development manager at CGIS confirmed that no specific Agile framework or methodology is currently implemented, although the tenets and principles of the Manifesto for Agile Software Development (see Beedle et al. (2001)) are considered important. CGIS and the company's context is discussed in the following section.

1.4 Enterprise context

Company-GIS (CGIS), a geographical information company, currently employs ±90 employees across five cities of which three are in South Africa. Departments at CGIS collaborate on software development projects that should be delivered within cost, on time, and meet the agreed scope, while also adhering to customer quality requirements. According to Chatzoglou, Theriou, Dimitriadis, and Aggelides (2007), the goal for all software project managers is to bring a project to completion on time, within the budgeted costs, and to meet the planned performance or end-product goals by orchestrating all resources assigned to the project effectively and efficiently. Although sophisticated tools are available, projects are still delivered behind schedule, cost more than initially estimated, and fail to meet user requirements.

Multiple stakeholders could be involved in the projects at CGIS, for example the sales representative, the client, the project manager, the development team member(s), the tester, as well as the IT team member(s) that assist with deployment(s) and support. In addition to this, there is a call centre that assists with product support, product owners, and, in some projects, data team member(s).

© University of Pretoria

The problem at CGIS, presented in more detail in the following section, was identified in September 2019 when all team members were predominantly based at the same office space in a South African city. According to Wiersinga and Prescott (2020), enterprises were forced into a distributed software development (DSD) context due to COVID-19 restrictions, mandating online coordination. According to Jiménez, Piattini, and Vizcaíno (2009), DSD allows team members from the same organisation or different organisations to be located in various remote sites during the software lifecycle, thus making up a network of distant sub-teams. Traditional face-to-face meetings are not common in DSD due to the distance, and interactions require the use of technology to facilitate communication and coordination (Jiménez et al., 2009). Although most CGIS employees reside in the same country and are not impacted by factors such as different time zones, remote working and virtual correspondence i.e., distributed teams now exist.

1.5 Problem statement

We validate a problem instance at CGIS in Chapter 3 and indicate in Chapter 4 that the problem instance also exists as a class-of-problems. A problem at CGIS, as well as other enterprises within the broader software development sector, is inadequate communication between software development project stakeholders. According to DeFranco and Laplante (2017), ineffective communication is confirmed by the number of papers that cite project failure points, and emphasized by the large number of research papers included in their study that focus on improving communication in software development.

General terms are used for inadequate communication in both the study conducted at CGIS (refer to Chapter 3) as well as the principle studies included in the SLR (refer to Chapter 4). The nature of the communication challenges is often not investigated/included in the study, and by not comprehensively defining and/or classifying the inadequate communication, the nature of the challenge could be misunderstood. A lack of systematic understanding of the nature and underlying dimensions of collaboration in ASD was identified by Batra, Xia, and Zhang (2017), and they state that the absence of a well-defined conceptualisation for and underlying dimensions of collaboration hinders researchers' ability to develop a formal measurement of collaboration. This limits practitioners' ability to create effective mechanisms to manage the specific facets of collaboration to improve ASD.

At CGIS, inadequate communication results in negative impacts on stakeholder alignment and project performance variables, such as project delivery. In DSD teams, according to Jiménez et al. (2009), lack of continuity in communications causes stakeholders to struggle to find the right person and/or timely information. This hinders stakeholders from working together efficiently, which results in misalignment, re-planning, redesign, and rework. In literature it is stated that communication gaps result in increased implementation costs and a higher test effort (Abelein & Paech, 2011), and can have serious and expensive consequences, such as wasted effort, quality issues, not meeting client expectations, and other RE-related challenges such as over-scoping (Bjarnason, Wnuk, & Regnell, 2011).

The requirement elicitation/analysis phase is especially prone to inadequate communication at CGIS. The SLR included in Chapter 4 indicates that of the principle studies that referred to inadequate communication during a specific phase, 93% identified the requirement elicitation/analyses phase as problematic, emphasising that other enterprises also experience communication challenges in this phase of the software development project. In a study conducted by Iden, Tessem, and Päivärinta (2011), in which they explore the problems encountered between system development and IT operations in system development projects, it was determined that the most serious problems are: (1) Stakeholders not being involved in the requirements specification, and (2) Poor communication and information flow. Communication is a challenging part of RE and inadequate communication could result in requirements being misinterpreted or overlooked (Bjarnason et al., 2011).

CGIS is already moving away from the waterfall approach towards Agile, but without a systematic approach in adopting a standard Agile framework. The focus of this study is to explore Agile RE solutions that could address the communication problems between stakeholders during requirement elicitation at CGIS. Using action design research (ADR), the objective is therefore to select, adapt, implement, and evaluate a solution artefact that addresses a subset of specific communication challenges in software development companies (such as CGIS) during the requirement analysis phase, to reduce the level of misunderstanding that takes place between stakeholders that affect project delivery.

To address the communication problem(s) at CGIS, 9 primary research questions which should be addressed systematically during the study were identified and are presented in the following section.

1.6 Research questions

In adherence to the problem statement (see section 1.5) and according to the main research question template provided by Wieringa (2014), the primary research question of this study is: *What adaption of an Agile RE solution (or a sub-set of its associated mechanisms) could address a subset of classified communication challenges in*

© University of Pretoria

software development companies (such as CGIS) to improve the information flow/communication between stakeholders during RE in order to assist management in reducing the misalignment between project stakeholders and/or the negative impact on project delivery?

The following questions decompose the main research question and will be answered throughout this study:

- *RQ1:* What are the problems experienced at CGIS that hampers/impacts the delivery of projects within the project management triple constraint? Answered in Chapter 3. *Rationale:* To determine the main areas of concern in project delivery at CGIS as well as the causes thereof.
- *RQ2:* Do the communication challenges identified at CGIS exist within the broader software development sector? Answered in Chapter 4. *Rationale:* To validate that the problem(s) experienced at CGIS exist as a class-of-problems within software development projects, i.e., this study will not only address a problem that exists at CGIS alone.
- *RQ3:* What are the specific communication challenges that exist during the requirements elicitation/analysis phase at CGIS that should be addressed? Answered in Chapter 4 (refer to section 4.4). *Rationale:* To comprehensively define and/or classify the inadequate communication so that the nature of the challenge is understood, and that the solution's impact can be compared/measured.
- *RQ4:* What are existing Agile methodologies, frameworks, and Agile RE practices? Answered in Chapter 5 (section 5.1). *Rationale:* To understand existing Agile RE solutions before the most suitable solution for CGIS is selected (see *RQ5*).
- *RQ5:* Which Agile RE solution would be the most suitable for CGIS to address the communication challenges identified? Answered in Chapter 5 (section 5.2). *Rationale:* To determine which solution (within scope) should be implemented at CGIS to address the communication challenges experienced.
- *RQ6:* How (and why) should the selected Agile RE solution be adjusted/adapted for CGIS? Answered in Chapter 5 (section 5.3). *Rationale:* To determine how the solution could be implemented to (1) Ensure feasibility of the solution within CGIS, and (2) Ensure that the stakeholders involved feel included in the decision, which could result in acceptance and willingness to actively support and participate in the study.
- *RQ7*: How were the Agile RE components implemented? Answered in Chapter 6 (section 6.1). *Rationale:* To demonstrate a real-world implementation of the adapted Agile RE solution.
- *RQ8:* Based on the implementation and evaluation results, how well are the communication challenges addressed by the implemented solution? Answered in Chapter 6 (section 6.2.2). *Rationale:* To determine whether the communication problem was addressed/ by the solution developed and implemented at CGIS.
- *RQ9:* What were the deficiencies of the implemented solution? Answered in Chapter 6 (section 6.2.3). *Rationale:* To determine improvement opportunities and suggestions for future research.

The thesis statement, aligned to the research questions presented above, is included in the following section.

1.7 Thesis statement

The thesis statement of this study is as follows:

An Agile RE solution can be designed to address a subset of classified communication challenges in software development companies (such as CGIS) to improve the information flow/communication between stakeholders during RE, that will assist management in reducing the misalignment between project stakeholders and/or the negative impact on project delivery.

To conduct this study, certain conditions and boundaries have been set and there are therefore limitations which might have influenced the results. The delineations and limitations identified are included in the following section.

1.8 Delineations and limitations

The following delineations and limitations were defined for which the thesis statement will hold true:

- The study was done in South Africa, and it can therefore not be stated that the thesis will hold true for other countries.
- The study was done within the context of a medium sized enterprise, and it can therefore not be stated that the thesis will hold true for other sized enterprises.
- The study was done within the context of a geographical information centric enterprise that executes software development projects and it can therefore not be stated that the thesis statement will hold true for other contexts.
- The study was done in an environment where multiple languages are spoken and multiple cultures are present in the work environment, and the impact of this on the thesis will not be measured i.e., it can therefore not be stated that the thesis will hold true in monolingual contexts.

© University of Pretoria

1.9 Significance

Inadequate communication between software development project stakeholders is a problem, as identified in a specific organisational setting (refer to Chapter 3) and supported by literature (refer to Chapter 4).

The following sections synthesize the value that this study will be adding to the existing knowledge base, motivating the significance of the study from five perspectives, detailed in the sub-sections.

1.9.1 Agile RE – practical demonstrations needed

Curcio et al. (2018, p. 32) state that it is difficult to describe or characterise Agile requirements engineering (RE) because "it is still cloudy, not only for software developers but for the research community too". Agile RE applied to real-world projects remains scarce and therefore requires empirical evaluation of practices in industry cases (Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015). This is supported by Schön et al. (2017), in which it is stated that there is a need for more empirical studies on Agile RE due to the lack of appropriate guidelines in practice, as well as Mendez-Fernandez et al. (2017) that state that empirical evidence in RE is particularly weak with case studies being isolated and small-scale studies that can't be generalised.

1.9.2 Effective communication in software development teams – insight required

Communication, collaboration, and coordination are significant components of software development in general and specifically to the Agile methodology of software development (Mishra & Mishra, 2009). When multiple and integrated stakeholder teams are involved, communication effectiveness is particularly critical to project success (Mishra & Mishra, 2009). In a study conducted by Mtsweni and Mavetera (2019) in South Africa, a need is identified to validate soft issues that limit the sharing of tacit knowledge within software development teams and to conduct research on these issues. Pikkarainen et al. (2008) state that the communication in projects using Agile practices should be examined, and that there is a need to obtain a systematic and insightful understanding of communication in ASD teams. This view is supported by Inayat, Salim, and Marczak (2017) that emphasises the importance of evaluating the incorporation of social aspects in Agile teams.

1.9.3 Stakeholder communication during RE – more insight required

Schön et al. (2017) have found that appropriate methodologies should be identified for building a shared understanding concerning the user perspective among project members and stakeholders. According to Buchan (2014), there is a need for empirical research considering that the shared understanding of requirements is an enduring challenge and error-prone, due to its inherent complexity. Bridging the communication gaps between RE and other development roles and activities, in particular for distributed development, has been identified as an important area for future RE research by Cheng and Atlee (2007). The need for increased insight in this area is further highlighted by Inayat et al. (2015) who found that development teams may adhere to very different communication structures than those prescribed in a formal process.

1.9.4 Need to reduce misalignment and the negative impact on project delivery

Many RE challenges facing large-scale software development are of an organisational and social character, rather than technical character (Bjarnason et al., 2011). In the explanatory case study performed by Bjarnason et al. (2011), a number of communication gaps that affect requirements were identified, which results in misalignment. RE-related problems could have a significant impact on IT projects and their final outcomes (Jarzębowicz & Poniatowska, 2020), including project failures or challenges. According to Jarzębowicz and Poniatowska (2020), it is difficult for researchers and industry practitioners to investigate and measure how decisions about RE influence other project areas. Evaluation of RE practices could however provide valuable information about the effectiveness of RE practices among the IT industry as well as improvement opportunities regarding the RE process for future projects at a particular company (Jarzębowicz & Poniatowska, 2020).

1.9.5 Applying ASD in Africa – more studies needed

In Chapter 4, in which we validate that the problem exists as a class-of-problems in literature, 4% of the principle studies were conducted in the South African context, and only one other African study was conducted in Nigeria.

© University of Pretoria

A lack of peer-reviewed literature was also identified while searching for sources on ASD in Africa. Peer-reviewed literature found included a study by Kunda, Mulenga, Sinyinda, and Chama (2018) in which challenges of the Agile methodology in Zambia were reported, as well as a study by Sebega and Mnkandla (2017) in which it was identified that elicitation, documentation, and non-functional requirements integration need special attention in Agile RE in the South African software industry. According to Ferreira and Cohen (2008), that conducted an empirical study of 59 South African development projects, case studies and longitudinal research into Agile methodologies would benefit South African developers in understanding the contexts in which Agile development is most appropriate, as well as long-term implications for system quality and maintainability.

© University of Pretoria

Chapter 2– Research methodology

This study aims to adapt an Agile framework and practices to address a subset of classified communication challenges in software development companies, such as CGIS, in order to reduce the level of misalignment between project stakeholders during the requirements analysis phase of software development projects in order to assist management in reducing the impact of the problems caused on project delivery.

In section 2.1 we provide an overview of research methodologies as well as qualitative research methods. The selected research methodology for this study is elaborated on in section 2.2. Validity, reliability, and generalisability of the study is included in section 2.3, limitations are included in section 2.4, and ethical considerations are included in section 2.5.

2.1 Literature on research methodologies

Case studies have the objective of increasing knowledge, e.g. about individuals, groups, and social phenomena, and bringing about change in the phenomenon being studied (Runeson, 2012). It is an empirical inquiry that "investigates a contemporary phenomenon in depth and within its real-world context, especially when the boundaries between phenomenon and context may not be clearly evident" (Yin, 2014, p. 16) and "relies on multiple sources of evidence, with data needing to converge in a triangulating fashion" (Yin, 2014, p. 17).

Action research (AR) is the "systematic inquiry that is collective, collaborative, self-reflective, critical, and undertaken by the participants of the inquiry" (McCutcheon & Jung, 1990, p. 148). AR is designed to establish a close association between actions and solving problems, and it involves researchers and participants of a research situation in a cooperative and participatory way (Collatto, Dresch, Lacerda, & Bentz, 2018). AR studies identify a problem, conduct data analyses, plan actions, implement actions and finally present an evaluation of a problem (Collatto et al., 2018). By repeating this cycle, i.e., using an incremental approach, the researcher's knowledge of the original question is enhanced and could therefor lead to an appropriate solution.

Design science research (DSR) involves the design of a new artefact. According to Hevner, March, Park, and Ram (2004), design is both a process and an artefact. The aim of DSR is to generate knowledge of how things can and should be constructed or arranged, i.e. designed, to achieve a desired set of goals (vom Brocke, Hevner, & Maedche, 2020). The DSR cycle's explicit stages according to Peffers, Tuunanen, Rothenberger, and Chatterjee (2007) are to identify a problem, suggest a solution, develop the artefact to satisfy intentions, demonstrate and evaluate the artefact against intentions, concerns, and/or criteria, and communicate the results. In the cycle presented by Peffers et al. (2007), evaluation only takes place after design, development, and demonstration. This process has been further developed, specifically with regards to the evaluation activities and allowing for a more concurrent evaluation of intermediate steps in the design process (vom Brocke et al., 2020), therefore mitigating risk and incorporating feedback sooner. An artefact is produced from the design process and by iteratively evaluating the artefact, the problem, the artefact, and the design process can be better understood until it is satisfactory (vom Brocke et al., 2020).

Action design research (ADR) combines the activities of AR and DSR, constructing an artefact to address the problem and addressing a problem in a specific organisational setting (Sein, Henfridsson, Purao, Rossi, & Lindgren, 2011). ADR therefor addresses both technological rigour and organisational relevance. The original ADR process model suggested by Sein et al. (2011) identifies four stages, namely (1) Problem formulation, (2) Building, intervention, and evaluation (BIE), (3) Reflection and learning, and (4) Formulation of learning. Researchers in collaboration with participants identify a problem and suggest an artefact or course of action, design/demonstrate/intervene/evaluate, reflect, and communicate results. Mullarkey, Hevner, and Ågerfalk (2019) found that the presence of multiple entry points with clear definitions allows the team to investigate their project goals and identify the entry point that best aligns with the project's purpose. They re-defined the ADR stages, defending four *intervention stages*: (1) Diagnosing, (2) Design, (3) Implementation, and (4) Evolution. In addition to the new ADR stages, they show that each stage consists of a cycle of (1) Problem-formulation, (2) Artefact-development, (3) Evaluation, (4) Reflection, and (5) Learning. Therefore, Mullarkey et al. (2019) propose that an artefact should be built and evaluated in every ADR stage, and that reflection and learning take place during each cycle of an ADR stage. The artefact built in each cycle is built and evaluated in order to address the problem formulated in that ADR cycle and each iteration learns from prior cycle(s) and modifies the problem formulation. This process is depicted in Fig. 1.

© University of Pretoria

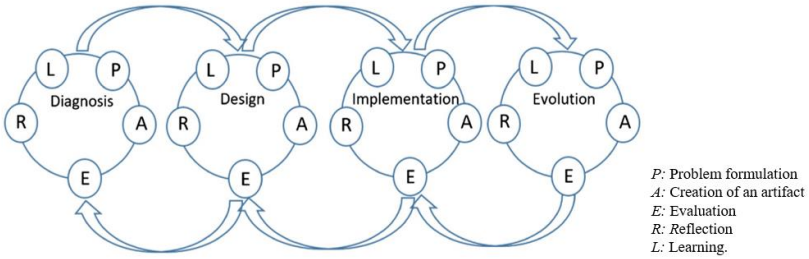


Fig. 1. The ADR process model according to Mullarkey et al. (2019)

Qualitative research uses text as empirical material and is interested in the perspectives of participants in everyday practices and everyday knowledge regarding the issue under study (Flick, 2018). Data-gathering is an important part of conducting research. According to Marshall and Rossman (2006), qualitative researchers typically rely on four *primary methods for data-gathering*, namely (1) Participating in the setting, (2) Observing directly, (3) Interviewing in depth, and (4) Analysing documents and material culture.

Participating in the setting i.e., *participant observation* is both an overall approach to inquiry and a data-gathering method as first-hand involvement in the study context is required and it allows the researcher to experience reality as the participants do. Observing directly includes the systematic noting and recording of events, behaviours, and artefacts in the social setting chosen for study. *Field notes* i.e., detailed and concrete notes on what is observed, are then used to discover and investigate complex interactions in natural social settings including examples such as an interviewee's body language during an interview. According to Phillippi and Lauderdale (2018), qualitative field notes are an essential component of rigorous qualitative research which enhances data and provides rich context for analysis.

In-depth *interviewing* plays an important role in qualitative research and this data-gathering method could be used with other methods or as the overall strategy (Marshall & Rossman, 2006). Qualitative, in-depth interviews are typically like conversations with the researcher exploring topics to assess the view(s) of the participant and how the participant frames and structures the responses. Cooperation is essential during interviews and the most important aspect of the interviewer's approach is conveying the attitude that the participant's views are valuable and useful. Kvale and Brinkmann (2015) describe seven stages that should be followed during interviews, namely (1) Thematising the interview, (2) Designing the interview, (3) Conducting the interviews, (4) Transcribing the interviews, (5) Analysing the interviews, (6) Determining validity, reliability, and generalisability of the interview results, and (7) Communicating the findings. It is important to understand the implications and requirements of transcription and translation of in-depth interviews. Lastly, *document analysis* is a systematic procedure for reviewing documents and requires that data be examined and interpreted to gain understanding and develop empirical knowledge (Bowen, 2009).

Several *secondary methods for gathering data* can be used such as focus groups, questionnaires/surveys, life histories and narrative inquiry, historical analysis, and interaction analysis (Marshall & Rossman, 2006). In a focus group, participants are interviewed. Usually, 7 to 10 unfamiliar people are selected based on certain shared characteristics relevant to the study's questions. The purpose is for the researcher to identify trends in the perceptions and opinions expressed. Questionnaires are used to gather information regarding the distribution of characteristics, attitudes, or beliefs on a certain topic. Questionnaires typically entail several questions with structured response categories, and the questions are assessed to address/consider factors such as bias and clarity.

According to Strauss and Corbin (1998), qualitative data analysis is a search for relationships and underlying themes. For *text to be analysed as a proxy for experience*, collecting and analysing words or phrases (i.e. techniques for systematic elicitation) was explored by Bernard, Wutich, and Ryan (2017) as well as methods that require the reduction of text to codes (i.e. free-flowing text such as narratives and responses to open-ended interview questions). For analysing chunks of text, coding is paramount in whole-text analysis (Bernard et al., 2017) and allows the researcher to make judgements regarding the meanings of contiguous blocks of text. Once the researcher has identified themes or concepts, a conceptual model is built to determine how the themes are linked, e.g., grounded theory, schema analysis, and analytic induction.

Grounded theory is "a set of inductive and iterative techniques designed to identify categories and concepts within text that are then linked into formal theoretical models" (Guest, MacQueen, & Namey, 2012a, p. 12). Thematic analysis is similar to grounded theory, with the only difference being method (i.e., there is no need to build a theoretical model). According to Guest et al. (2012a), thematic analysis makes use of an iterative approach to analyse text and identifies and describes both implicit and explicit ideas within the data, i.e. themes. A codebook is used for analysis with the purpose of identifying commonalities, differences, and relationships amongst instances of meaning in text.

Considering the literature presented, the research methodology relevant to this study and its objectives are discussed in the following section.

© University of Pretoria

2.2 Research methodology for the study

This study aims to identify and develop a solution artefact that addresses the communication challenges within CGIS during RE in software development projects. The solution artefact will be implemented and evaluated at CGIS to determine whether the problem is addressed and/or how the problem is impacted by the solution.

Addressing a problem in a specific organisational setting and constructing an artefact to address this problem is required, as well as collaboration between the researcher and CGIS employees. *Action design research* (ADR) was selected as the most applicable research methodology for this purpose. Considering that CGIS practitioners were involved during the diagnosis, design, and implementation stages of this study, the study relied *primarily* on qualitative data-gathering and analysis. In-depth interviews, focus group sessions, thematic analysis, and note-taking was used for data-gathering and qualitative data analysis in the study, as depicted in Fig. 2 and detailed below.

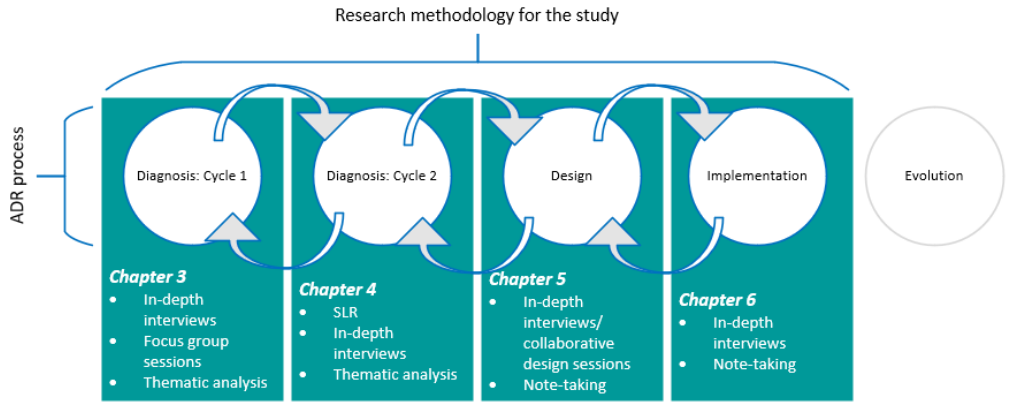


Fig. 2. Research methodology

Mullarkey et al. (2019) defined a sequence of iterative cycles of four different main ADR stages (diagnosis, design, implementation, and evolution) in ADR, gradually assembling multiple artefacts through these stages instead of combining activities into one single BIE stage (Sein et al., 2011). Mullarkey et al. (2019) suggest that an abstracted artefact should be built and evaluated in every cycle, and that reflection and learning should also be performed in every cycle.

This study covers three of the four main ADR stages to develop a *main artefact*, i.e., an *Agile RE solution (ARES)*, in accordance with the thesis statement (refer to section 1.7). The *diagnosis* stage was the initial point of entry and during this stage, the study also follows two cycles, each cycle consisting of problem-identification, artefact-development, evaluation, reflection, and learning. The *diagnosis* stage therefore produced two secondary artefacts, explained in the following two paragraphs.

The researchers in collaboration with CGIS participants identified a problem with how software development projects are being executed at CGIS. The first problem identified, by conducting in-depth interviews with CGIS practitioners as detailed in Chapter 3, is that project tasks take longer than expected, negatively impacting on-time delivery, quality of delivery and delivery within budget. The artefact created was synthesized *root cause analysis results*, identifying *inadequate communication* as one of the main concerns. The artefact was evaluated with CGIS participants in focus group sessions.

An SLR (refer to Chapter 4) was then conducted as a second cycle of the diagnosis stage. The aim was to determine whether the challenges identified at CGIS exist within the broader software development sector. The researcher determined that the problem identified was too vague and that an additional intervention was required. To identify and understand the *specific communication problems* at CGIS, an additional round of interviews was conducted with CGIS employees (refer to section 4.4). The artefact created, using thematic analysis, was a *rudimentary taxonomy of perceived communication problems at CGIS* (refer to section 4.4.3).

Once the problem at CGIS was comprehensively diagnosed, in collaboration with participants, the ADR *design* stage was initiated. According to Mullarkey et al. (2019), the design stage of the ADR process consists of one or more iterative cycles in which design principles emerge that address the problem-class identified via diagnosis and move towards the implementation of a solution. The main artefact, the ARES, that would address the specific problem experienced at CGIS was preliminarily developed by the researcher, based on literature (refer to section 5.1 and section 5.2). The adapted artefact, collaboratively modified between the researcher and CGIS practitioners, is elaborated in section 5.3.

The ADR *implementation* stage followed, during which the adapted artefact/intervention was implemented and evaluated (refer to Chapter 6). Additional feedback was obtained from the CGIS employees after the intervention to identify improvement opportunities. Refer to Chapter 6 (section 6.2) for the results of the implementation.

© University of Pretoria

According to Lawrence (2015), the *quality* of qualitative research can be assessed by considering the validity, reliability and generalisability of the research. According to Guest, MacQueen, and Namey (2012b, p. 3), validity is the "the notion that one is assessing what one is intending to assess". There are six types of validity, namely validity, content validity, construct validity, criterion validity, external validity, and internal validity (Guest et al., 2012b). External validity is also known as *generalisability*, which is "the degree to which study findings are relevant to other populations and contexts" (Guest et al., 2012b, p. 3) and *reliability* is "consistency when repeating or comparing assessments within a study" (Guest et al., 2012b, p. 4). This study's validity, reliability, and generalisability is discussed in the following section.

2.3 Validity, reliability and generalisability

According to Guest et al. (2012b) there are techniques that could decrease the likelihood of making critical mistakes and increase the degree of transparency within a study, which therefore enhances validity and reliability. Table I depicts the techniques relevant to and used by the researcher in each ADR stage of this study.

Table I. Techniques to enhance validity and reliability per ADR stage

ADR Stage	Chapter	Techniques used by the researcher to enhance validity and reliability, based on the techniques suggested by Guest et al. (2012b)
Diagnosis: Cycle 1	Chapter 3	• Using multiple methods and/or data sources e.g., individual interviews, focus group session. • Having participants review their feedback in the focus group session. • Using transcription protocols and developing and using a precise codebook. • Using an intercoder agreement and creating an audit trail i.e., documenting the analysis and steps. • Triangulating data sources.
Diagnosis: Cycle 2	Chapter 4	• Using transcription protocols and developing and using a precise codebook. • Using an intercoder agreement and creating an audit trail i.e., documenting the analysis and steps. • Supporting themes and interpretations with quotes.
Design	Chapter 5	• Using multiple data sources i.e., interviewing multiple participants and comparing their feedback. • Supporting themes and interpretations with quotes.
Implementation	Chapter 6	• Using multiple data sources i.e., interviewing multiple participants and comparing their feedback, collecting data via interviews, observations, and the retrospective events. • Supporting themes and interpretations with quotes. • Having participants review their feedback e.g., the participants could provide feedback and make suggestions during the implementation.

According to Kitto, Chesters, and Grbich (2008), guidelines for assessing and presenting qualitative research exist which can be used to enhance the rigour of qualitative studies. This includes clarification, justification, procedural rigour, representativeness, interpretation, reflexivity and evaluative rigour, and transferability. Table II indicates how this study addressed the criteria presented by Kitto et al. (2008).

Table II. Criteria for assessing and presenting qualitative research and relevance to this study

Criteria (Kitto et al., 2008)	Description
Clarification & justification	• The aim of the research is included in Chapter 1. • The research questions are clear and outlined in Chapter 1.
Procedural rigour	• The study describes how the research was conducted in each ADR stage and detail how data are collected and analysed, e.g., in section 4.1. • The study provides detail about interviews/focus group sessions conducted and how participants/sources were accessed, who was interviewed and observed (participant demographics), how often and how long participants were interviewed as well as the questions that were asked, e.g., section 3.1.1, section 5.3.2, and section 6.2.1.
Representativeness	• The study included different sources from different years, publishers, and authors in the SLR i.e., Chapter 4. • The primary researcher (PR) interviewed CGIS employees from different departments and with different demographics, e.g., experience at CGIS, to seek representativeness in terms of participants.
Interpretation	• The PR offered participants the opportunity to view and amend their input in the first cycle of diagnosis. • The study used differing forms of triangulation, e.g., multiple evidentiary sources, multiple methods, and multiple theoretical and conceptual frames to enhance insights in this study.
Reflexivity & evaluative rigour	• The study included and elaborated on limitations per ADR stage. Refer to section 2.4. • Ethics approval was obtained, ethical considerations were adhered to. Refer to section 2.5.
Transferability	• The problems identified as well as the Agile RE solution could be implemented/relevant to contexts that differ from the context in which the original study was undertaken. The generalisability of the main contributions is elaborated on in section 8.2.

© University of Pretoria

2.4 Limitations

Each ADR stage in this study, including two cycles of diagnosis, one cycle of design, and one cycle of implementation, has limitations. The limitations per ADR stage are included in the relevant chapters. For the first cycle of *diagnosis* (Chapter 3), the limitations are included in section 3.4. The systematic literature review, which is the second cycle of *diagnosis* (Chapter 4), has method-related limitations and result-related limitations which are included in section 4.3.2. In the *design* stage (Chapter 5), limitations are included in section 5.4 and in the *implementation* stage (Chapter 6), limitations are included in section 6.3.

2.5 Ethical considerations

Ethical issues can be alleviated using the following ethical principles namely autonomy, beneficence, and justice (Orb, Eisenhauer, & Wynaden, 2001).

The CGIS practitioners (i.e., participants) were informed about the study, their right to decide whether to participate or not, i.e., that participation is voluntary, and their right to withdraw from the study at any time to ensure *autonomy*. Written consent was obtained from participants before every interaction, i.e., before every round of interviews and the focus group session. Participants in this study provided consent that opinions, feedback, and any other data gathered could be included and directly referred to in the study and research. Consent was also specifically obtained per participant for recording the interview(s)/focus group session and including the answers/responses in this study. The Faculty of Engineering, Built Environment and IT's ethical committee approved the study.

For *beneficence*, confidentiality was crucial in the study. Although data, e.g. participant details, opinions, and feedback, obtained in the study was kept for traceability, the data was anonymised in the study so that it could not be linked back to a specific person. It could however be linked back to a specific role. Anonymity of the employees participating in the research was also ensured by not referencing the enterprise directly in the study/research, i.e. CGIS was used as a pseudonym.

According to Orb et al. (2001, p. 95), justice "refers to equal share and fairness" i.e. to understand the vulnerability of the participants and their contributions to the study. This was not applicable to this study.

The research methodology has been presented and motivated in this chapter, as well as other considerations relevant to the study. The ADR process for this study, as outlined in section 2.2, begins with the first cycle of the diagnosis stage. The following chapter elaborates on this first diagnostic cycle by detailing the initial problem identified at CGIS.

© University of Pretoria

Chapter 3– Background- initial problem

This study covers three of the four main ADR stages to develop a main artefact, as presented in Chapter 2. The diagnosis stage was the initial point of entry and during this stage, the study also follows two cycles, each cycle consisting of problem-identification, artefact-development, evaluation, reflection, and learning.

Chapter 3 details the *first cycle* of the diagnosis stage (refer to Fig. 3, initially presented in section 2.2) and the purpose of this chapter is to validate a problem at a real-world medium-sized enterprise in South Africa in the geographical information system (GIS) industry when they execute software development projects.

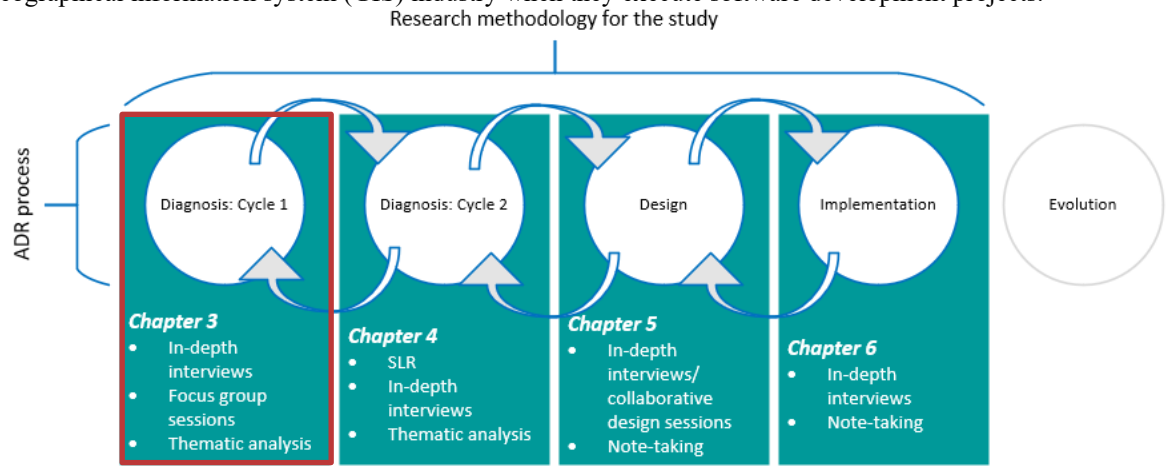


Fig. 3. Research methodology- Diagnosis cycle 1

This chapter therefore addresses *RQ1*: *What are the problems experienced at CGIS that hampers/impacts the delivery of projects within the project management triple constraint?*

The research method used to determine the main areas of concern is included in section 3.1, the results are included in section 3.2, and the findings are included in section 3.3. Limitations on the initial problem diagnosis is included in section 3.4.

3.1 Problem validation research method

To determine the nature and origin of some of the existing problems experienced at CGIS, data was gathered by interviewing employees (section 3.1.1). The data gathered is qualitative in nature and thematic analysis was therefore used to interpret the findings. A current reality tree (CRT) was established to identify the root causes of the problems experienced (section 3.1.2) and focus group sessions were conducted to gather employees' opinions, ideas, and beliefs regarding the identified root causes (section 3.1.3).

3.1.1 Problem validation interviews

Interviews were conducted with CGIS employees to determine and therefore validate a problem at CGIS. The researcher prepared a list of questions and explored a few general topics to assist participants in freely structuring a response. The interviews included questions formulated in assessing employees' knowledge of CGIS projects as well as determining problems experienced or suggested improvement opportunities on changing the way-of-working at CGIS.

The agenda of the individual interviews was as follows: (1) Welcome the interviewee, (2) Explain the focus and intent of the interview, (3) State that "projects should be delivered within cost, on time, and meet the agreed scope while also adhering to customer quality requirements", (4) Discuss the structured/predetermined interview questions and obtain the interviewees' opinions/responses, and (5) Thank the interviewees for their involvement, and close the interview.

© University of Pretoria

Data-gathering cycles were conducted on three different days in September 2019 with all interviewees being interviewed, using eight questions:

1. What evidence/examples exist indicating that CGIS does not deliver projects within cost?
2. What evidence/examples exist indicating that CGIS does not deliver projects on time?
3. What evidence/examples exist indicating that CGIS does not meet the agreed scope when delivering projects?
4. What evidence/examples exist indicating that CGIS does not adhere to customer quality requirements in projects?
5. What have been your biggest frustrations while working on CGIS projects and why? Please provide examples.
6. What was the impact of these problems on the project's cost, time it took to deliver the project, project scope, or the quality delivered?
7. What, according to your knowledge, has CGIS implemented or done to address or attempt to address these problems?
8. What do you think CGIS can do or implement to improve the way they approach and/or execute projects to ensure that projects are delivered within cost, on time, and meet the agreed scope while also adhering to customer quality requirements?

Interviews ranged between 7 and 42 minutes in duration. A total of seventeen (17) CGIS employees were interviewed, i.e., ±18% of the employees at CGIS, and interviewees were selected based on availability and involvement within seven (7) departments. Employees ranging from junior level to management were interviewed, ensuring that employees that recently joined CGIS were considered, as well as employees who have been working for the company for more than twenty years. Data analysts, a system analyst, product owner, and project manager, developers, an IT support technician, a software tester, a support centre team lead, a graphic designer, the operations director, operations manager, data unit manager and the office manager were interviewed.

Interviews were transcribed and were interpreted by using thematic analysis. According to Guest et al. (2012a), thematic analyses identifies and describes both implicit and explicit ideas within the data and identifies themes. To represent these identified themes, codes were identified, developed, and applied or linked to the raw data and then analysed. Three coding families were used to group emerging codes, namely *area of concern*, *cause*, and *suggested solution*. Once the codes were defined, a colleague, i.e., an Industrial engineer also affiliated with the University of Pretoria, coded two articles according to the codebook definitions. The coding results of the colleague and the researcher were compared, and the codebook was refined as required. In accordance with Guest et al. (2012a), an inter-coder agreement of at least 80% was required to proceed with coding. The interview results are included in section 3.2.

3.1.2 Root Cause Analysis (RCA)

According to Rooney and Heuvel (2004), root cause analysis (RCA) helps identify what, how, and why an event occurred, enabling the researcher to specify corrective measures to prevent future events of the same type. A comparison of three RCA techniques, namely the CED (the Cause-and-Effect Diagram), the ID (Interrelationship Diagram) matrix, and the CRT (Current Reality Tree) are documented by Doggett (2005). As suggested by Goldratt (1990), factors of problems are interdependent and are the result of a few root causes. The CRT assists in determining core problems by relating multiple factors instead of evaluating isolated events.

Based on the analysis, it was decided that a CRT would be used in this study. The CRT was created to determine root causes by relating multiple factors instead of evaluating isolated events at CGIS i.e., analysing relationships between the challenges in order to identify root causes. It is important to note that *one* CRT was initially created, but for ease of viewability and understanding in the focus group sessions, the CRT was simplified into four different figures/section and printed out on A4 pages. This was done by grouping some root causes in the same diagram so that it was easier for participants to follow the results of the root causes. The original CRT is depicted in Fig. 4 to indicate the size of the CRT. The four simplified diagrams which were shown to the focus group participants are included in Appendix A.

© University of Pretoria

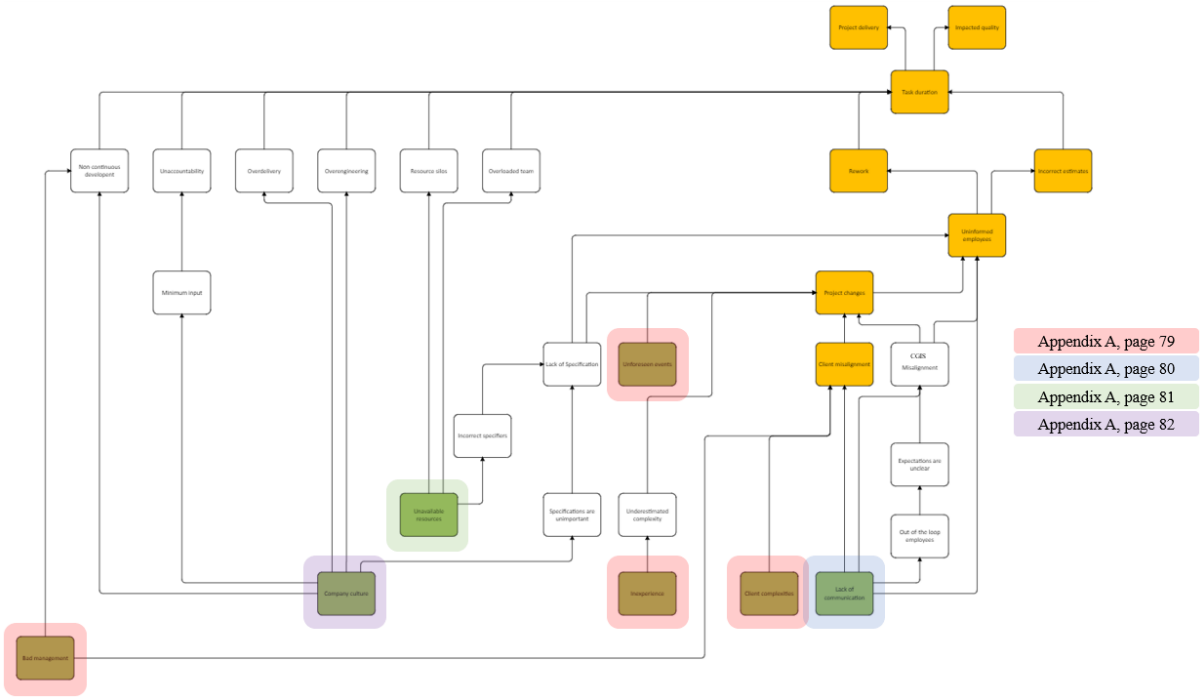


Fig. 4. The original CRT created

3.1.3 Problem validation focus group discussions

Although it is recommended by Krueger and Casey (2015) that at least three focus groups are selected to enable comparison of data between focus groups, two focus groups were used, each with a single-category design of either management or non-management. Only two focus groups were used as the focus group sessions were only conducted to further validate the root causes as determined by the comprehensive interviews. Due to resource availability, only two participants were included in the management focus group, while four participants were present in the non-management group. All focus group participants were previously interviewed.

Participants were allowed the opportunity to structure responses freely and the interviewer facilitated the conversation to extract suggested changes to the CRT (refer to section 3.1.2). The agenda of the focus group sessions was to welcome the participants, explain the focus and intent as well as confidentiality of opinions, explain the CRT, discuss the focus group questions while obtaining participant feedback, and thank the participants for their involvement. According to Krueger and Casey (2015) questions should evoke conversation, be short and clear, one dimensional and easy. The questions should further be open ended, include words that a participant would use, and be consistent between focus groups.

Following these guidelines, the focus group questions were as follows:

1. Do you agree with the symptoms identified?
2. Do you agree with the root causes identified?

In the sessions, participants were also encouraged to disagree with the systems and root causes identified. The results of the focus group session are included in section 3.2.

3.2 Problem validation results

Thematic content analysis was used to identify prominent areas of concern and causes in the transcribed interviews. Codes were identified by the primary researcher by reading the transcripts multiple times and deriving common themes from the interview data. A total of 8 codes were created for areas of concern and 21 codes were created for causes.

According to Guest et al. (2012a), code frequency comparisons, code co-occurrence analyses, and graphical display of relationships between codes within the data set could be used to analyse the interview transcriptions. The absolute code frequencies were calculated as the total number of times a theme appears in the data set. The same theme/code was only counted once per interviewee even if the theme was mentioned/addressed more than once in the interviewee's transcript. Absolute code frequencies per theme as analysed in transcribed interviews are included in Table III.

© University of Pretoria

Table III. Project absolute code frequencies as analysed in transcribed interviews

	Code label	Short description	Absolute frequency
Area of Concern	Rework (R)	Project rework	6
	Long task durations (TD)	Tasks taking longer than expected	3
	Impacted project delivery (PD)	Projects delivering over time/budget	13
	Impacted quality (IQ)	Project quality impacted	3
	Client misalignment (CM)	Misalignment between client and AfriGIS	11
	Project changes (PC)	Project factors keep on changing	4
	Incorrect estimates (IE)	Incorrect project estimates	8
	Uninformed employees (UU)	Employees are uninformed in general	13
Cause	Over-engineering (O)	Finetuning a project/product/feature more than required	5
	Non-continuous development (ND)	Interrupted development runs	2
	Unforeseen events (UF)	Factors or situations that were not planned for	8
	Unavailable resources (UR)	Resources not being available	8
	Inadequate communication (IC)	Communication lacking the quality or quantity required	13
	Inexperience (I)	Inexperience	3
	Negative company culture (CC)	The personality of the company is	6
	Unaccountability (U)	Employees not taking responsibility	6
	Minimum input provided (MI)	People only do what is expected	5
	Resource silos (RS)	Inability of resources to operate in all systems	2
	Overloaded team (OT)	Resources have too much to do	2
	Over delivery (OD)	Employees provide more than is required	4
	Inadequate management (IM)	Management lacking the quality or quantity required	14
	Client complexities (CX)	Factors in the clients' control	10
	Underestimated complexity (UC)	Underestimated factor or situation	8
	Inadequate specifications (S)	Specifications lacking the quality or quantity required	7
	Incorrect specifiers (IS)	Wrong people are doing specifications	4
	Specification unimportant (SU)	Specifications are not a priority	1
	Out-of-the-loop employees (OE)	Employees are uninformed due to inadequate communication	11
	Unclear expectations (UE)	Employees don't know/understand what is expected	8
	CGIS misalignment (AM)	Internal misalignment (between CGIS departments)	14

In Fig. 5, we illustrate the transcribed interviews' thematic content analysis results of the highest 13 frequencies based on the absolute code frequencies included in Table III. CGIS internal misalignment and inadequate management were the dominant themes, followed by impacted project delivery, inadequate communication, and uninformed employees.

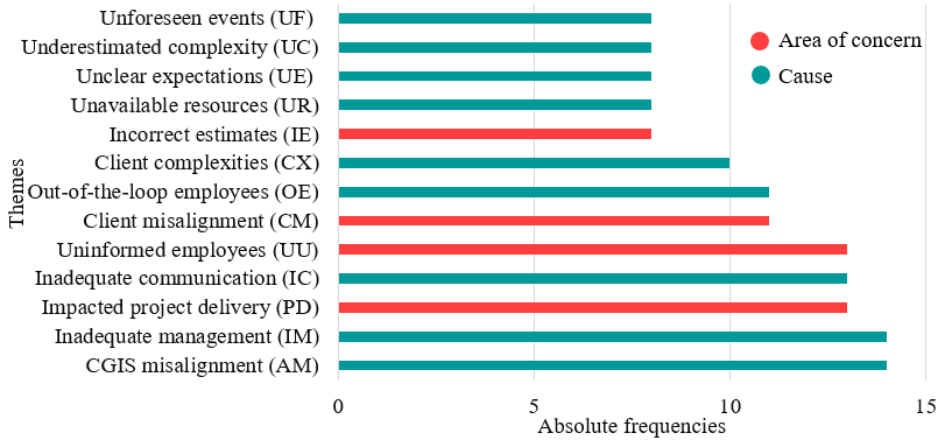


Fig. 5. Absolute code frequencies per theme as analysed in transcribed interviews

© University of Pretoria

The diagnosis revealed that project tasks take longer than expected at CGIS, which has an impact on projects not being delivered on time, within budget, and impacting the quality of the delivery. The root causes identified include unavailability of resources, negative company culture, inadequate management, client complexities, inexperience, unforeseen events, and *inadequate communication*.

Each focus group was shown printed-out versions of the four CRT diagrams (refer to Appendix A) and asked whether they agreed or disagreed with the symptoms and root causes identified. Both focus groups predominantly agreed with the CRT but had suggested alterations/improvements. Note that the CRT included in Appendix B depicts the alterations suggested in the focus group sessions.

The first group, consisting of non-management participants, suggested code-naming changes e.g., that *lack of specification* should be changed to *inadequate specification*. It was suggested that project changes could result in rework as well, instead of only uninformed employees resulting in rework. The focus group also emphasised that (1) Inadequate management should include insufficient planning, (2) Motivation, or the lack thereof, is considered as part of company culture, and (3) Misappropriation of resources is also a contributing factor to unavailable resources. The company's culture and the influence that this has on employee morale was emphasised as well as company culture often playing a role in inadequate communication e.g., employees purposefully not communicating due to previous negative encounters/situations forming certain attitudes and beliefs.

The second group, consisting of management participants, elaborated on inadequate communication stating that personality types and a lack of skills play a role. The focus group stated that inadequate communication is not a company culture issue, and that non-continuous development is sometimes as per design, i.e., not always due to inadequate management. It was also added that non-continuous development could be the result of unforeseen events as well as unavailable resources. This focus group insisted that an additional root cause should be added, namely employee attitude/personality. It was suggested that employees' attitudes/personalities are solely the cause of minimum input being provided and specifications being considered as unimportant, and jointly responsible for unaccountability, over-delivery, and overengineering (in addition to company culture).

In the following section, we summarise the results obtained during the *first cycle* of the diagnosis stage and focus our research on one root cause before commencing with the *second cycle* of the diagnosis stage.

3.3 Problem validation findings

The diagnosis revealed a key concern at CGIS. Project tasks take longer than expected, negatively impacting on-time delivery, quality of delivery and delivery within budget. One of the root causes identified included *inadequate communication*. In an enterprise consisting of ±90 employees, with all team members predominantly based at the same office space in a South African city, it is a concern that communication is neglected, and misalignment occurs frequently.

Although other root causes were identified, our research focuses on inadequate communication, since the primary researcher had observed similar communication challenges on a software development project while working at a different, larger enterprise within the telecommunications industry. Fig. 6 includes the section of the CRT (i.e., one of the four sections of the CRT) that includes the *inadequate communication* as the root cause.

© University of Pretoria

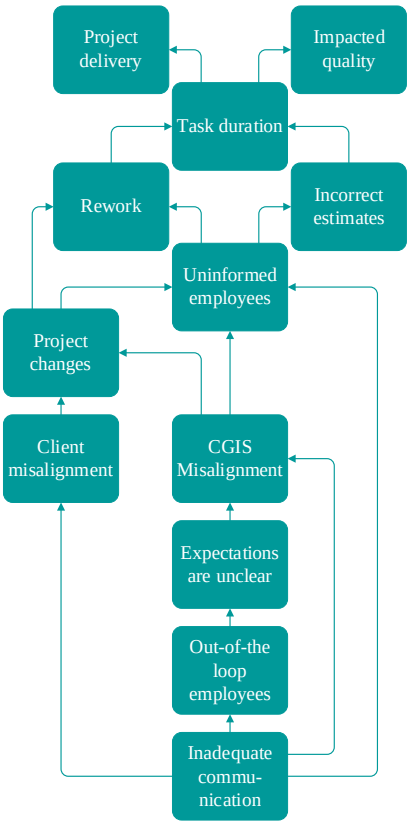


Fig. 6. Section of the CRT that depicts inadequate communication as the root cause

3.4 Limitations of the initial diagnosis

Considering that the researcher is an employee at CGIS, the researcher could have been biased towards certain problems at CGIS. To mitigate this risk, the data-gathering process included focus group discussions with the participants to validate the problem(s) identified by the researcher. Measuring and comparing the problems experienced by CGIS practitioners were dependent on the specific participants' and researcher's opinions and it could therefore be subjective. It is also possible that some themes were missed. Mitigating this threat to validity, the codebook was refined iteratively until an inter-coder agreement of 80% was achieved.

Less than 20% of CGIS employees were interviewed and not all interviewees were included in the focus group sessions due to availability. The results were the perceptions of the participants that were present/included and are therefore not necessarily a representation of all CGIS participants. The primary researcher also acknowledges that disciplinary perspectives could influence the way problems are perceived. Participants were chosen from different departments/functions to attempt to mitigate this risk.

The objective of the next chapter is to determine whether the CGIS communication challenges identified and discussed in this chapter exist within the broader software development sector. Chapter 4 therefore includes the *second* cycle of the diagnosis stage, i.e., a literature study which should reveal more insight into this kind of problem and extract existing solutions from the knowledge base.

© University of Pretoria

Chapter 4– Systematic literature review - further problem diagnosis

The researchers in collaboration with CGIS practitioners identified problems with how software development projects are being executed at CGIS in Chapter 3. *Inadequate communication* is one of the main concerns that result in project tasks taking longer than expected, negatively impacting on-time delivery, quality of delivery and delivery within budget.

In this Chapter, the *second cycle* of the diagnosis stage (refer to Fig. 7, initially presented in section 2.2) is elaborated on. The purpose of this chapter is to determine whether the communication challenges identified at CGIS exist within the broader software development sector.

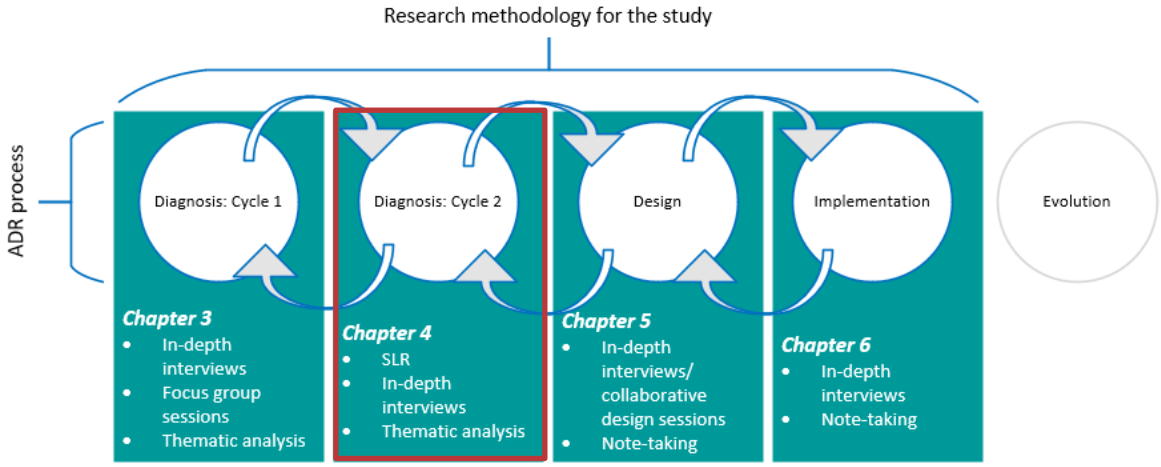


Fig. 7. Research methodology- Diagnosis cycle 2

The following research question is addressed in this chapter, i.e., *RQ 2*: *Do the communication challenges identified at CGIS exist within the broader software development sector?* The following secondary research questions were identified for the SLR:

- *RQ 2.1*: What existing SLRs focus on inadequate communication within a software development context? *Rationale*: To determine whether the problem instance features as a class-of-problems in literature and whether the SLRs focus on our specific context.
- *RQ 2.2*: How does the context of the enterprise and the study performed in section 1.4 compare to the principle studies included in the SLR? *Rationale*: To gain insights on the similarities and differences between the study conducted at CGIS and that of the principle studies that exist in literature.
- *RQ 2.3*: During which phase(s)/step(s) of software development projects do inadequate communication (and/or misalignment) most often occur? *Rationale*: To determine whether a specific phase is more susceptible to communication challenges.
- *RQ 2.4*: Is there evidence that other enterprises also have difficulties due to inadequate communication within a software development context? *Rationale*: To determine whether the challenges experienced at CGIS occur within the broader software development sector.
- *RQ 2.5:* Is there evidence that other enterprises also experience negative project impacts when the software development project experiences inadequate communication? *Rationale:* To determine whether the impact of communication challenges on certain project performance variables is similar within the broader software development sector.
- *RQ 2.6:* Is there evidence that other enterprises also experience misalignment between stakeholders as a result of inadequate communication within a software development context? *Rationale:* To determine

© University of Pretoria

whether the impact of communication challenges on stakeholder alignment is similar within the broader software development sector.

After conducting the SLR, the researcher determined that the problem identified was too vague and that an additional intervention was required. To identify and understand the *specific communication problems* at CGIS, as perceived by the CGIS employees, an additional round of interviews was conducted with CGIS employees. Therefore, the following research question is also addressed in this chapter, i.e., *RQ 3*: *What are the specific communication challenges that exist during the requirements elicitation/analysis phase at CGIS that should be addressed?*

The SLR research method is included in section 4.1 and the SLR results are included in section 4.2. The findings of the SLR are discussed in section 4.3 and the perceived communication challenges at CGIS are included in section 4.4.

4.1 SLR research method

An SLR was performed, following the guidelines from Okoli (2015). The review was conducted between 1 September 2020 and 31 October 2020, identifying peer-reviewed articles published/available up to 31 October 2020. We now discuss the SLR protocol that was used.

Searching for literature:

After defining our research goals and questions, a single search method was used to address the six secondary research questions (i.e., *RQ 2.1 – RQ 2.6*). Keywords for the search string were extracted from secondary research questions that define our main research inquiry, namely *RQ 2.4*, *RQ 2.5*, and *RQ 2.6*. The first three secondary research questions provided additional guidance during the data extraction phase, extracting contextual data from the identified sources.

Synonyms for keywords were identified by reviewing related work (see section 4.2.1), also using wildcards ("*") to accommodate grammatical variants of a base word. The synonyms were joined with the OR operator and the different keywords, related to key concepts of our inquiry, were joined with the AND operator. The following keyword string was used, presented as parts:

[Part A] *ti:*(difficult* OR problem* OR challeng* OR inadequa* OR concer* OR issues OR complicat* OR hurdles OR barriers OR risks) AND
[Part B] *kw:*(knowledg* OR align* OR communicat* OR coordinat* OR collaborat* OR integrat*) AND
[Part C] *kw:*(software OR software develop* OR software engineer*) AND
[Part D] *ti:*(project OR deliver*) AND
[Part E] *kw:*(horizontal* OR horisontal* OR depart* OR inter* OR intra* OR function*).

Part A, B, and C of the search string address the challenges, inadequate communication, and software development context reflected in *RQ 2.4*, *RQ 2.5*, and *RQ 2.6*. Part D addresses negative impacts on performance variables when the software development project or the delivery of software development in general experiences inadequate communication. Part D highlights one of the performance variables, i.e., *delays in project delivery* (see *RQ 2.5*), whereas Part E addresses another performance variable, i.e., the possible *misalignment between stakeholders* (see *RQ 2.6*).

The University of Pretoria's online library was used as the platform for the initial search, since it is based on WorldCat.org, the world's largest bibliographic database. All articles held by libraries worldwide were included in the initial search, which includes Science Direct. The basic search string was adapted if necessary based on the search engine. Other search engines were also included as motivated below.

- Scopus, Elsevier's abstract and citation database, is known to include software engineering systematic reviews (Scopus, 2021).
- IEEE Xplore digital library provides web access to more than five-million full-text documents from some of the world's most highly cited publications in computer science and other allied fields (IEEE Xplore, 2020).
- SpringerLink was manually searched for relevant *Requirement Engineering: Foundation for software quality* (REFSQ) articles due to their specific focus on requirements engineering and our theory that *inadequate communication* at CGIS is closely related to ineffective requirements engineering during the software development process.
- Google scholar was used during forward snowballing as a data extraction technique.

Practical screening:

Filters were applied on the search engines to ensure that all *inclusion criteria* were met. The titles of the remaining sources were reviewed. Regarding *exclusion criteria*, the titles of extracted sources were evaluated for relevance. If the title was deemed relevant to the study, the abstract and the page count of the source were reviewed. If the source was still deemed relevant, the source was extracted for further consideration. If the source's title, abstract, or page limit were not relevant, the source was excluded. We now present the inclusion criteria and exclusion criteria that were applied during practical screening.

© University of Pretoria

Inclusion criteria:

- Only peer reviewed literature sources, written in English, where full text could be accessed either online by the researcher or via an inter-lending request were included.
- To focus our research, only human communicative interaction(s) within the software development context was considered.
- Online book chapters were included, whereas full books were excluded.
- Sources were included regardless of publication date i.e., all sources published/available up to 31 October 2020 were included, ensuring that all relevant articles were analysed.

Exclusion criteria:

- Literature with a title or an abstract that indicates that the content of the literature is not relevant to the research questions. *Rationale*: Extracting studies that are aligned with our inquiry.
- Studies that are not focused on a software development context/environment were not considered e.g., *new product development* if software development is not specifically mentioned in the study. *Rationale*: The purpose is to explore literature in search of communication challenges within the software development context.
- Studies that refer to communication challenges due to a change in software development context e.g., if an article describes communication challenges due to the enterprise changing from a waterfall to Agile approach, the article is excluded. *Rationale*: Our research focuses on the software development context and different phases of software development, not on methodology change management or communication challenges as a result of the change management.
- Studies focused on open-source software environment(s)/crowdsourcing i.e., the communication/relationships between strangers. *Rationale*: Focusing our research on software development and project stakeholder communication.
- Studies that refer to difficulties on implementing ERP software. *Rationale*: This study is solely focused on the software *development process* and not the *implementation* of software or commercial-off-the-shelf software packages.
- Studies consisting of less than five pages. *Rationale*: The assumption is that studies consisting of less than five pages will not be comprehensive enough to provide insights into the causes of ineffective communication within the software development process.

Quality appraisal:

To systematically assess the literature and judge trustworthiness, value, and relevance, quality assessment questions have been defined, adapted from The pocket guide to critical appraisal (Crombie, 1996).

The sources extracted during *practical screening* were scanned to identify section(s) addressing the quality assessment question(s). While in some sources, finding the answer(s) were simple (based on the document's structure), others had to be read in full to answer the quality assessment questions.

For inclusion as a principle source, all questions had to result in a positive outcome:

1. Is the study developed/supported by an organisation/institute/consortium?
2. Does the study have a bearing on the defined research question(s)?
3. Does the study address a clearly focused question?
4. Was a comprehensive literature search (with listed sources) conducted for the study?
5. Are the inclusion and exclusion criteria clearly defined?

Data extraction:

The quality appraisal step extracted SLRs as well as other studies. To ensure a comprehensive search, forward and backward snowballing were conducted on studies that were included after the quality appraisal step, including SLRs. The SLRs extracted have not been considered as principle studies and were only included in section 4.2.1 (Related work). The studies identified via the snowballing process on the extracted SLRs have however been considered for inclusion as principle studies if the source was not eliminated during the practical screening and quality appraisal process.

An iterative snowballing procedure was followed. Backward snowballing was applied by analysing the reference lists cited for a principle study, say principle Study A, to identify new studies to include. Using Google Scholar, forward snowballing was applied by identifying new studies that had cited Study A. The new studies identified via snowballing were then examined, applying the practical screening techniques, as well as quality appraisal techniques to supplement the list of *principle studies*.

Once the backward and forward snowballing was done for one iteration, the newly identified principle studies were put aside for the next iteration. To ensure traceability, one iteration was performed at a time. The process was stopped once data saturation was achieved, i.e., no additional papers were identified during a snowballing iteration.

The SLRs and principle studies, as well as their authors, were then reviewed to identify duplicate studies or studies which had an updated version of the research published. Duplicate studies or those for which a more recent version of the study exists, were excluded.

© University of Pretoria

A codebook was developed that included both *structural codes* and *content codes*. Fig. 8 indicates structural codes to extract data from principle studies regarding Source title (C1), Author(s) (C2), Year (C3), Publisher (C4), and SLR-indication (C5). The SLR-indication addresses *RQ 2.1*. Data extracted for structural codes are available in Appendix C.

Content codes emerged as new themes emerged, related to *RQ 2.2* to *RQ 2.6*. The content codes were classified into two main coding families, Context and Class-of-Problems, as indicated in Fig. 8.

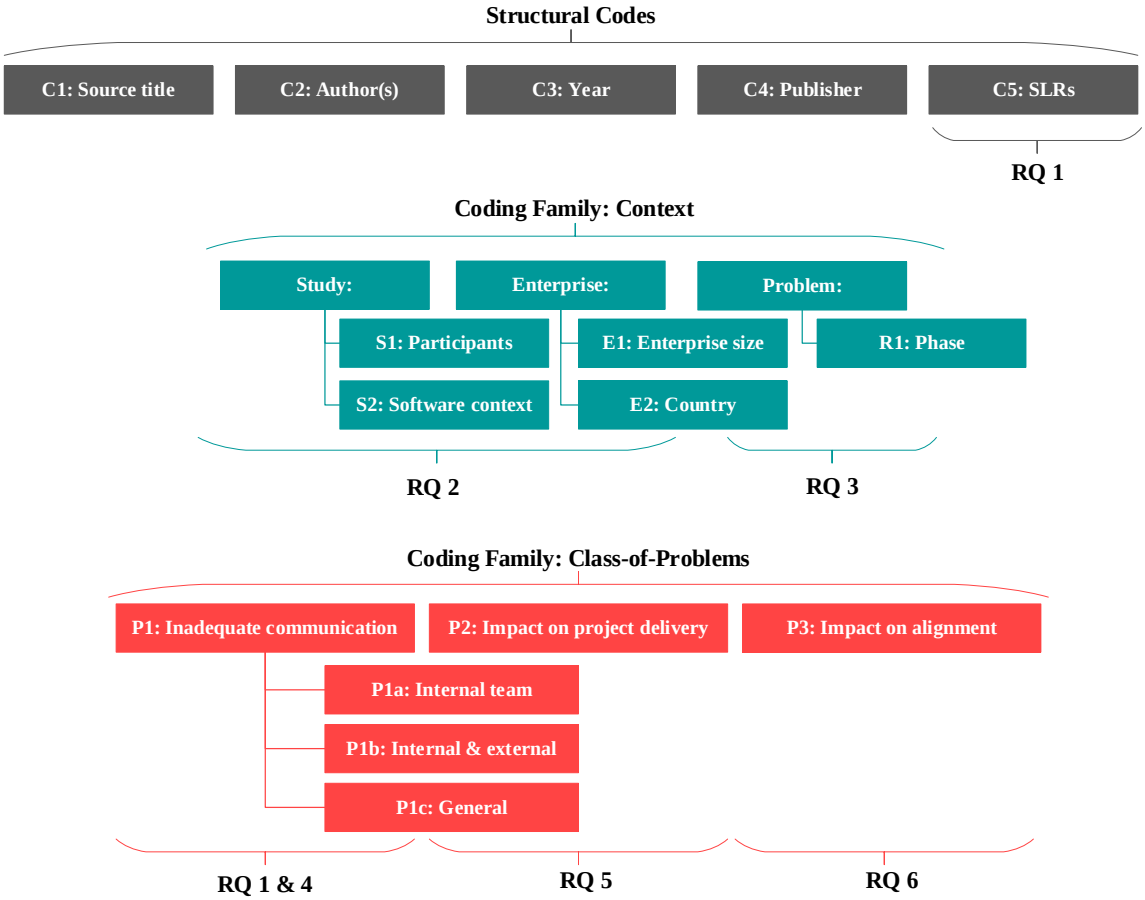


Fig. 8. Themes extracted

Once the codes were defined, a colleague i.e., an industrial engineer also affiliated with the University of Pretoria, coded two articles according to the codebook definitions. The coding results of the colleague and the researcher were compared, and the codebook was refined as required. In accordance with Guest et al. (2012a), an inter-coder agreement of at least 80% was required to proceed with coding.

Analysis of findings: Thematic analysis was used to analyse the content of the principle studies and to identify themes. Atlas.ti was used as the qualitative analysis tool to analyse themes in terms of absolute code frequencies. The frequency of a theme was calculated as the unique number of studies that referred to the particular theme. Therefore, each occurrence of a theme was only counted once per study.

The SLR results are included in section 4.2. We present the SLRs that were extracted from the selected repositories as Related work (section 4.2.1) and the results of the principal studies in terms of: (1) Context results (section 4.2.2); and (2) Themes that highlight the existence of *inadequate communication* as a class-of-problems within software development (section 4.2.3).

4.2 SLR results

Based on our research method (section 4.1), we now present the results of the SLR. *RQ 2.1* is addressed in section 4.2.1 (Related work), *RQ 2.2* and *RQ 2.3* are addressed in section 4.2.2 (Context results), and *RQ 2.4*, *RQ 2.5*, and *RQ 2. 6* are addressed in section 4.2.3 (Class-of-problems results).

Fig. 9 depicts the different stages of the SLR protocol, as discussed in section 4.1. A total number of 51 principle studies were identified.

© University of Pretoria

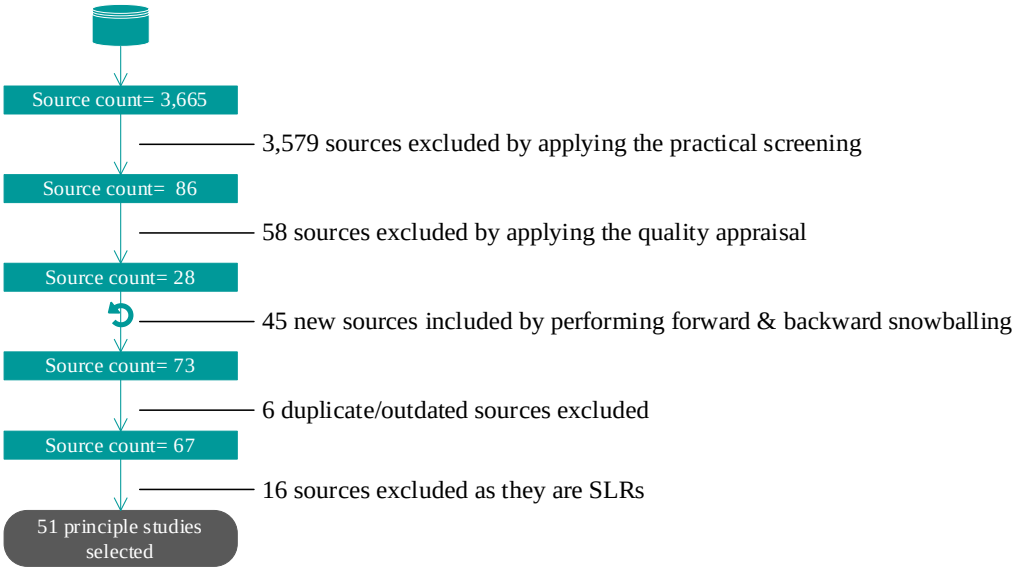


Fig. 9. Number of studies identified per step in the SLR

Most of the principle studies were published in 2014 (8 studies), and 2007 (5 studies). One principle study was published in 2020 (Javed et al., 2020) and two principle studies were published in 2019 (Alsaqaf, Daneva, & Wieringa, 2019; Mtsweni & Mavetera, 2019). The 2019 studies focused on requirement management challenges, whereas the 2020 study focused on issues that limit tacit knowledge sharing within software development project teams.

The 16 SLRs identified and excluded from the principle studies are discussed in section 4.2.1, whereas the 51 principles studies are presented in section 4.2.2 and section 4.2.3.

4.2.1 Related work

Of the studies identified, 16 are SLRs. Whereas all 16 studies referred to inadequate communication, only one study focused on inadequate communication within a software development context, i.e. Alzoubi, Gill, and Al-Ani (2016) systematically reviews literature on geographically distributed Agile development (GDAD) communication challenges.

Although DeFranco and Laplante (2017) investigate the type and quality of software development team communication, the study does not focus solely on inadequate communication. In addition, two studies (i.e., Nidhra, Yanamadala, Afzal, and Torkar (2013) and Zahedi, Shahin, and Ali Babar (2016)) focus on knowledge sharing challenges in global software development (GSD).

4.2.2 Context results

This section compares contextual variables of CGIS, with contextual factors that are evident from the 51 principal studies.

CGIS is a medium-sized enterprise in South Africa. The medium-sized context is also represented in some of the principle studies. The size of the enterprise(s) in which the principle studies were conducted is mentioned in 19 of the 51 principle studies (37% of principles studies), an indication of *large* was mentioned in 14 studies, *medium* in 6 studies, *small* in 4 studies, and *micro* in 1 study. It is possible that different enterprise sizes were mentioned in a single study.

The county/countries in which the principle studies were conducted are mentioned in 29 principle studies (57% of principle studies) as depicted in Fig. 10. Most studies were conducted in American enterprises (18%), whereas two studies (Mtsweni and Mavetera (2019) and Addison (2003)) focused on South African enterprises. For our study at the South African company CGIS, we would only be able to extract context relevant knowledge from two studies.

© University of Pretoria

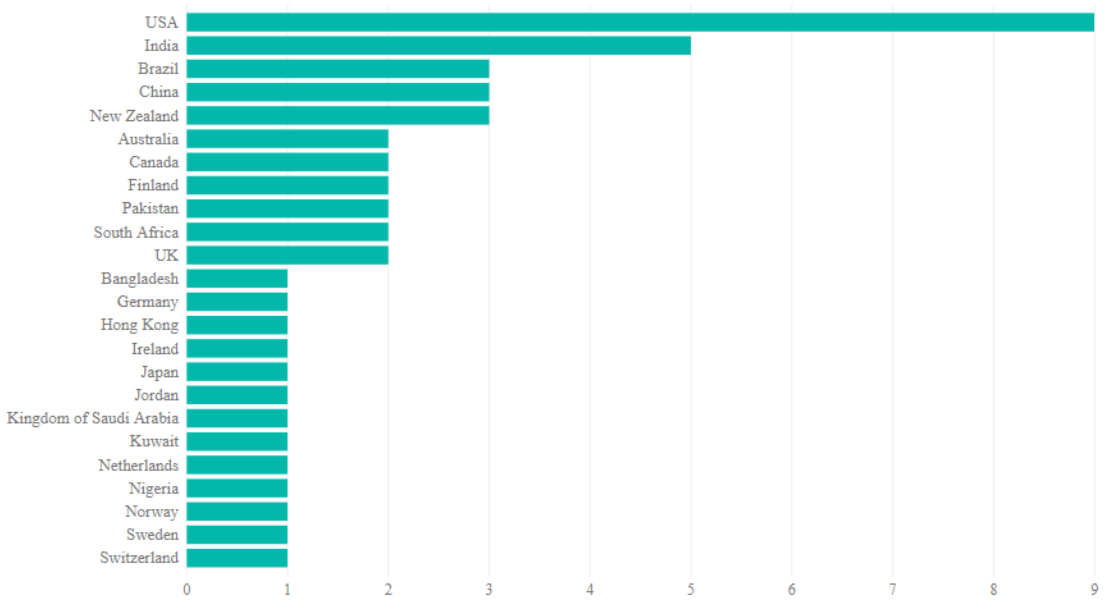


Fig. 10. Count of countries in which primary studies were conducted

Software development contexts were mentioned in 28 of the 51 principle studies (55% of principle studies). Fig. 11 depicts the number of times each software development context was referred to with the most studies (9 studies) referring to problematic communication in a distributed software development (DSD) context. More than one context could have been mentioned per study.

Whereas a mixture of the waterfall approach and Agile practices is followed at CGIS (depending on the project and/or client), seven (7) principle studies included communication challenges in an Agile context and the waterfall approach, also called software development life cycle (SDLC) approach, was mentioned in one study.

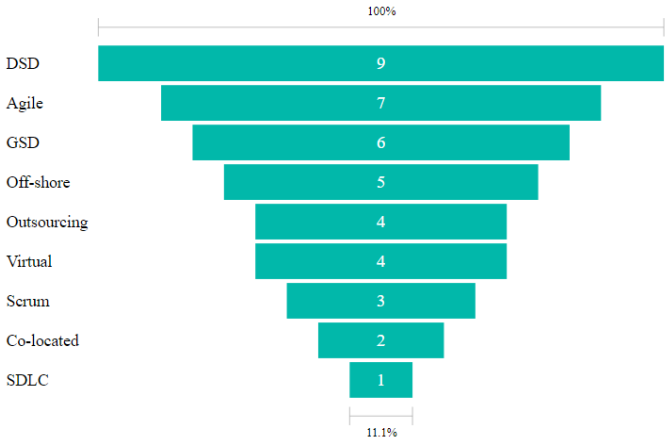


Fig. 11. Software development contexts referred to in principle studies

In the study conducted at CGIS (refer to Chapter 3), the roles of the participants involved in the study included developers (4), management (4, of which 1 was a senior executive, and 3 were general managers), and data analysts (2). The remaining participant roles included one of each of the following: Business analyst, project manager, product manager, IT support technician, quality assurer, call centre agent, and marketer.

Role representation in principle studies indicated that 48 principle studies (94% of principle studies) referred to the roles of study participants. Fig. 12 depicts which roles were most involved in the studies. Roles mentioned in less than five studies are not shown. The results indicate that project managers were included in 22 studies as participants whereas 14 of the studies included software developers as participants.

© University of Pretoria

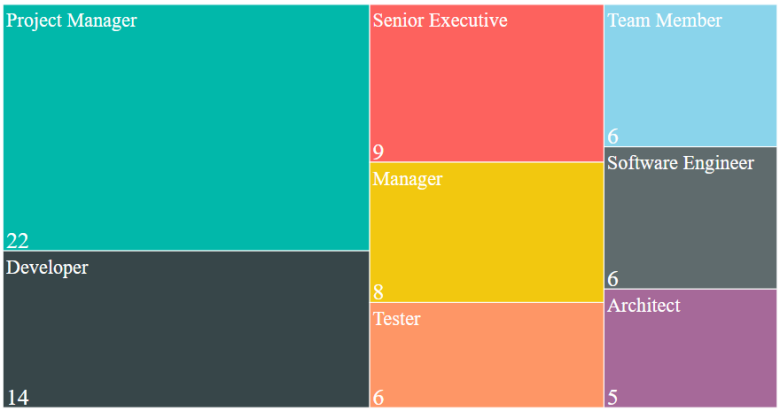


Fig. 12. Count of studies per role of empirical study participants

Of the principle studies, 27 (53% of principle studies) referred to a specific phase in which communication problems occur, of which 2 studies were sourced from SpringerLink's Requirement Engineering: Foundation for Software Quality (REFSQ) articles. Of these 27 studies, 25 (93% of phase-indicated studies) refer to inadequate communication during requirement elicitation/analyses, 2 studies refer to design, 1 study to implementation, and 3 studies to testing. There were no studies that referred to the deployment and maintenance phases. Of the 51 principle studies, 8 focus specifically on challenges in requirements engineering (RE), referencing inadequate communication as concerns.

4.2.3 Class-of-problems results

In this section, we convey the main results of this SLR. Three codes were extracted during thematic analysis, related to inadequate communication as a class-of-problems: (1) *Internal team*, used to reference inadequate communication among internal stakeholders, (2) *Internal and external*, used to reference inadequate communication among internal stakeholders and external stakeholders, and (3) Inadequate communication in *general*, used to reference communication problems that do not specifically refer to the involved stakeholders.

Internal communication deficiencies amongst team members were mentioned in 19 principle studies (37% of principle studies), whereas 15 studies (29% of principle studies) mentioned communication deficiencies among internal and external stakeholders. Inadequate communication in general featured in 48 of the 51 principle studies (94% of principle studies).

Examples of inadequate communication among internal stakeholders include "this process of channelling the 'right' information towards and between teams is difficult and time-consuming" (Kasauli, Liebel, Knauss, Gopakumar, & Kanagwa, 2017, p. 7), "communication and the flow of information between system development and operations are poor" (Iden et al., 2011, p. 398), "lack of communication between the team members" (Shrivastava & Rathod, 2015, p. 382), and "product owner lack of sharing client feedback with development team" (Ghobadi & Mathiassen, 2017, p. 712).

Inadequate communication among internal and external stakeholders includes "there is improper correspondence between customer and vendor" (Javed et al., 2020, p. 19), "lack of involvement and effective communication with the client" (Islam, Joarder, & Houmb, 2009, p. 350), "identifying and gaining access to appropriate users in the beginning of the project caused communication issues between the clients and the development team" (Hanisch & Corbitt, 2007, p. 799) and "lack of communication between team and the client" (Shrivastava & Rathod, 2015, p. 382).

Inadequate communication in general featured as terminology differences (Herbsleb, Paulish, & Bass, 2005; Junior, de Azevedo, de Moura, & da Silva, 2012), delays in communication (Damian & Zowghi, 2002; Hanisch & Corbitt, 2007; Holmstrom, Conchuir, Agerfalk, & Fitzgerald, 2006; Javed et al., 2020; Junior et al., 2012), knowledge transfer challenges (Damian & Zowghi, 2002; Kasauli et al., 2017), issues with infrastructure/technology (Junior et al., 2012; Reed & Knight, 2010a), infrequent communication (Herbsleb & Mockus, 2003; Iden et al., 2011; Junior et al., 2012), lack of casual correspondence (Damian & Zowghi, 2003; Herbsleb et al., 2005; Javed et al., 2020), lack of face-to-face meetings (Alnuem, Ahmad, & Khan, 2012; Iden et al., 2011; Islam et al., 2009; Junior et al., 2012; Nakatsu & Iacovou, 2009), and language barriers (Damian & Zowghi, 2002; Iden et al., 2011; Islam et al., 2009; Reed & Knight, 2010b; Sarker & Sahay, 2004).

General terms are used for inadequate communication in the principle studies identified (e.g., communication problems or inadequate communication). By referring to general terms rather than defining and/or classifying the inadequate communication, it is difficult to determine the cause of the communication problem and whether the problem experienced at CGIS is similar to the problem(s) experienced in the principle studies.

© University of Pretoria

Project impact(s) were mentioned in 15 principle studies (29% of principle studies). Examples include "increased implementation cost or test effort were named as consequences of communication gaps" (Abelein & Paech, 2011, p. 8), "the time taken to rectify miscommunication began to impact on the project deadline" (Hanisch & Corbitt, 2007, p. 797), "gross estimation errors, conflicting and continuous requirement changes, and inaccurate requirement analysis, thus delaying the recruitment of the team, which results in a gross miscalculation of the cost of the project" (Sharma, Sengupta, & Gupta, 2011, p. 82) and "may alter the scope of the project and thus affect schedules and budget" (Aundhe & Mathew, 2009, p. 421).

Alignment impact(s) were mentioned in 23 principle studies (45% of principle studies). Examples include "do not know which requirement is being addressed by whom" (Javed et al., 2020, p. 22), "often only the product owner is aware of which user stories originate from which requirements" (Kasauli et al., 2017, p. 7), and "not aware of the different interpretations of the requirements until late in the project" (Parviainen & Tihinen, 2014, p. 257). Misalignment can also lead to people "not being on the same page" (Reed & Knight, 2010a, p. 423) and difficulties to align the definition of "*done*" and "*testing activities*" (Vlietland & van Vliet, 2014, p. 305).

The key findings of the SLR results are presented in the following section.

4.3 SLR findings

Based on the SLR results presented in section 4.2, we now present the key findings in section 4.3.1, answering the secondary research questions. Section 4.3.2 provides a summary of limitations that may have an impact on the validity of the results.

4.3.1 Synthesis per research question

In this section we repeat the initial secondary research questions and communicate the main insights that were extracted from our study.

RQ 2.1: What existing SLRs focus on inadequate communication within a software development context? Only one of the 16 SLRs identified in this study focuses on *inadequate communication* within a software development context. Our results therefore indicate that although there are studies conducted within the software development context that *refer* to inadequate communication, most of these studies focus on a software development context specifically (e.g., Agile RE or distributed software development) and elaborate on *all* the challenges and risks within that context.

RQ 2.2: How does the context of the enterprise and the study performed in section 1.4 compare to the principle studies included in the SLR presented in this article? Few studies (4% of the principle studies) were conducted in the South African context, and only 1 other African study was conducted in Nigeria. Furthermore, most studies focused on large enterprises, whereas CGIS is a medium-sized enterprise. The lack of studies that are focused on our enterprise's context emphasizes the need for research within this medium-sized, South-African context. Regarding the studies conducted, most studies focused on distributed software development (DSD) which has become relevant to CGIS since the COVID-19 pandemic. Although CGIS employees reside in the same country (and are not impacted by factors such as different time zones), remote working and virtual correspondence (i.e., distributed teams) now exist. The roles from different phases of the software development project were included in the principle studies, similarly to the study conducted at CGIS, if the data is aggregated. Some principle studies only included participants with similar roles in the studies, e.g. Karlsson, Dahlstedt, Regnell, Natt och Dag, and Persson (2007) only included "*team members*", whereas Abelein and Paech (2011) only included the "*Software Development Professional*", and Herbsleb et al. (2005) only included management roles (i.e. project manager, manager, and senior executives). We believe that inclusion of only one type of participant is problematic, since the rest of the software development project teams' views are excluded in the analyses/findings.

RQ 2.3: During which phase(s)/step(s) of software development projects does inadequate communication (and/or misalignment) most often occur? Of the studies that referred to inadequate communication during a specific phase, 93% identified the requirement elicitation/analyses phase as problematic, emphasizing that other enterprises also experience communication challenges in this phase of the software development project. The assumption was made that by adding SpringerLink as a database (to ensure that articles on Requirements Engineering are also included), the requirements elicitation/analysis phase would be overrepresented. Yet, most sources referring to the requirements elicitation/analysis phase were sourced from other databases.

Our theory is that inadequate communication at CGIS is also closely related to ineffective RE during the software development process. To determine whether the communications challenges experienced at CGIS are closely related to ineffective requirements engineering during the software development process, and to ensure that the researcher obtains a better understanding of the exact communication challenges at CGIS, an additional round of in-depth interviews was conducted with fifteen (15) representatives to investigate the areas of concern regarding communication between project stakeholders on a software development project (refer to section 4.4).

© University of Pretoria

Fourteen of the fifteen interviewees referred to ineffective communication within the requirements elicitation/analysis phase specifically.

RQ 2.4: Is there evidence that other enterprises also have difficulties due to inadequate communication within a software development context? All 51 principle studies referred to inadequate communication as a problem within a software development context. Communication challenges were reported between internal stakeholders, and internal and external stakeholders, whereas some studies did not specifically refer to the stakeholders that experience communication challenges. The general terms used for inadequate communication in both the study conducted at CGIS (refer to Chapter 3) as well as the principle studies in the SLR (regardless of the stakeholders involved) is problematic. Examples include "*communication gaps*" in Bjarnason et al. (2011) and another study only referring to communication and coordination challenges (Alsaqaf et al., 2019). By referring to general terms rather than defining and/or classifying the inadequate communication, it is difficult to determine the cause of the communication problem and whether the problem experienced at CGIS is similar to the problem(s) experienced in the principle studies. It is necessary to understand the nature of communication challenges since potential solutions would also depend on the specific challenge. Communication gaps at CGIS, for example, could be caused by issues with infrastructure/technology whereas the communication gaps in Junior et al. (2012) could be caused by infrequent communication, in which case the solution(s) to each problem would be different. In the additional round of in-depth interviews (refer to section 4.4), the perceived areas of concern regarding communication between project stakeholders on a software development project were investigated. We were therefore able to better define and understand the nature of inadequate communication within their context.

RQ 2.5: Is there evidence that other enterprises also experience negative project impacts when the software development project experiences inadequate communication? Project impact(s) were mentioned in 29% of the principle studies, indicating that other enterprises also experience negative impacts on project delivery due to inadequate communication. At CGIS, project tasks take longer than expected, negatively impacting on-time delivery, and quality of delivery. Although the principle studies did not specifically refer to project tasks taking longer than expected, 4 principle studies referred to an impact on the schedule (Abelein & Paech, 2011; Aundhe & Mathew, 2009; Hanisch & Corbitt, 2007; Reed & Knight, 2010a), 1 principle study referred to an impact on quality (Bjarnason et al., 2011), and 4 principle studies referred to an impact on the project costs (Abelein & Paech, 2011; Aundhe & Mathew, 2009; Reed & Knight, 2010a; Sharma et al., 2011). Other impacts (e.g., over scoping, rework, and scope creep) were also mentioned. The SLR thus highlights additional themes extracted from principle studies, indicating that inadequate communication induces a number of negative effects on performance variables within software development projects. Although some negative project impacts (as a result of inadequate communication) were identified from the SLR, the evidence was lacking, and no significant insights emerged from the SLR findings.

RQ 2.6: Is there evidence that other enterprises also experience misalignment between stakeholders as a result of inadequate communication within a software development context? At CGIS, misalignment causes project tasks to take longer than expected. Regarding the SLR, alignment impact(s) were mentioned in 45% of the principle studies, indicating that other enterprises also experience an impact on misalignment as a result of inadequate communication within a software development context. The impacts mentioned in the studies refer to different types of misalignment (e.g. lack of team awareness, misunderstandings, misinterpretation and uncertainty). Our systematic study of existing literature rendered some examples of misalignment that occurs in practice but did not contribute to significant new insights.

4.3.2 Limitations of the literature review

Limitations of this study may pose threats to validity and can be classified as method-related limitations and result-related limitations.

Method-related limitations relate to the research method or protocol that was used in extracting a comprehensive set of sources from literature to answer the research questions. We acknowledge that the search method could have excluded some valid primary studies, i.e., some relevant studies might be excluded despite the researcher's best efforts to perform a comprehensive search. Certain exclusion criteria, e.g., studies that referred to methodology change management, could have contributed to valid studies being excluded. In addition to this, the focus on problems (and excluding solutions in the search) could have contributed to valid studies being excluded. By not limiting the scope of the search (e.g., by not including the requirement elicitation/analysis phase specifically) important articles within the RE field (which has been identified as a problematic area) could have been excluded. Due to time restrictions, some databases (e.g., Ebscohost and ProQuest) were excluded from the study. The search string in section 4.1 is very focused, searching for fragments within the title and keywords of sources. However, forward and backward snowballing was performed to obtain data saturation, mitigating the risk of excluding valid sources.

Result-related limitations relate to the thematic analysis phase and the objective to extract a comprehensive set of themes from the principle studies. It is possible that some themes were missed within the two main coding

© University of Pretoria

families, i.e., context results and class-of-problems results. Mitigating this threat to validity, the codebook was refined iteratively until an inter-coder agreement of 80% was achieved. Considering that a timeframe was not specified, studies are included from a large range of years, which could impact the results in terms of the relevance of problems. We acknowledge that disciplinary perspectives could influence the way communication, and communication problems, is perceived. Some of the principle studies may be flawed, inducing certain biases. As an example, some principle studies only included the opinions or observations of participants in the same roles/departments that may lead to an under-representation of valid themes. The use of general terms in the CGIS study and in the principle studies (e.g., inadequate communication, misalignment) indicate that the existing content may not be adequately expressive to elaborate on challenges and project impacts. Content limitations of existing studies (i.e., the principle studies) also highlight the need for further exploration within a real-world context and the need to develop a comprehensive set of themes that relate to inadequate communication.

As discussed in section 4.3.1, general terms were used for inadequate communication in both the study conducted at CGIS (refer to Chapter 3) as well as the principle studies in the SLR. By referring to general terms rather than defining and/or classifying the inadequate communication, it is difficult to determine the cause of the communication problem and whether the problem experienced at CGIS is similar to the problem(s) experienced in the principle studies. It is necessary to understand the nature of communication challenges since potential solutions would also depend on the specific challenge. The need was therefore identified to conduct an additional round of in-depth interviews, presented in the following section. This analysis to better define and understand the nature of inadequate communication within their context is included in this chapter as it contributes to the second cycle of the diagnosis stage.

4.4 Perceptions of communication challenges at CGIS

This section addresses *RQ3: What are the specific communication challenges that exist during the requirements elicitation/analysis phase at CGIS that should be addressed?* Although the objective is to identify and understand *communication problems in software development projects,* the interviews and analysis conducted enable the researcher to learn and understand what CGIS practitioners *perceive* to be communication problems within project management.

Two secondary research questions are used to answer *RQ3*:

- *RQ3.1:* What are the specific communication problems experienced at CGIS between project stakeholders on a software development project and how can these problems be structured? *Rationale:* To understand and analyse the communication-related problems for software development projects in a medium-sized enterprise in South Africa.
- *RQ3.*2: What are the specific communication problems experienced at CGIS in the *requirement elicitation/analyses* phase of software development projects and how can these problems be structured? *Rationale:* To understand and analyse the communication-related problems during the requirement elicitation/analyses phase of software development projects in a medium-sized enterprise in South Africa.

Interviews were used to further investigate communication-related problems at CGIS, as perceived by CGIS employees. In structuring the interviews, we searched for existing taxonomies on communication problems with the intention of using communication structures/taxonomies found in literature to facilitate the conversation with the CGIS participants, by for example asking them with which of the communication problems presented in the taxonomy they could relate to.

The literature search used to identify existing taxonomies on communication problems is detailed in section 4.4.1. The research methodology and interview strategy used is introduced in section 4.4.2, and the interview results are included in section 4.4.3.

4.4.1 Existing taxonomies on communication problems

In structuring the interviews, we searched for existing taxonomies on communication problems. Existing literature was reviewed, and the search was conducted between 1 March 2021 and 31 May 2021, identifying articles published/available up to 31 May 2021. A single search method was used:

ti:(communicat* OR knowledg* OR interact* or align* OR collabora* OR coordinat*) AND kw:(software development OR development) AND ti:(framework OR taxonomy OR classification OR glossary OR terminology OR ontology OR model*).

The basic search string was adapted if necessary based on the search engine. Search engines included were (1) Scopus, (2) IEEE Xplore digital library, and (3) Web of Science. Filters were applied on the search engines to ensure that all inclusion criteria were met. The titles of the remaining sources were reviewed. Regarding exclusion criteria, the titles of extracted sources were evaluated for relevance. If the title was deemed relevant to the study, the abstract and the page count of the source were reviewed. If the source was still deemed relevant, the source

© University of Pretoria

was extracted for further consideration. If the source's title, abstract, or page limit were not relevant, the source was excluded. We now present the inclusion criteria and exclusion criteria that were applied during practical screening.

Inclusion criteria:

- Only literature sources, written in English, where full text could be accessed either online by the researcher or via an inter-lending request were included.
- Only articles in which communication problems are categorised in a software development context were included.
- Online book chapters were included, whereas full books are excluded.
- Only sources published/available between 2006 and 2021 at the time of the search were included.

Exclusion criteria:

- Literature with a title or an abstract that indicates that the content of the literature is not relevant to the research questions. *Rationale:* Extracting studies that are aligned with our inquiry.
- Studies consisting of less than five pages. *Rationale:* The assumption is that studies consisting of less than five pages will not be comprehensive enough to provide insights into the causes of ineffective communication within the software development process.

The researcher expected to find an article that synthesizes, in a taxonomy, typical causes for inadequate communication in software development projects e.g., a categorisation of communication-related problems, linked to standard phases of the software development cycle, as well as solution pathways for the prevalent causes. This could however not be found in the search conducted, and the results of the SLR were therefore deemed irrelevant and excluded from the study.

This second SLR was initiated to find a taxonomy to assist the researcher, but instead highlighted the need to create a vocabulary, i.e. taxonomy, that could be used to describe communication problems in software development. In the following section we present the interviews conducted with CGIS practitioners to better understand the communication-related problem at CGIS before initiating the design stage.

4.4.2 Communication-related problem interviews at CGIS

A fundamental assumption in qualitative research is that a participant's and not the researcher's perspective should unfold during the interview. It was therefor decided that the interview guide approach would be used as the interviewer/researcher prepares a list of questions and explores a few general topics to assist interviewees/participants in freely structuring a response.

The agenda of the individual interviews was as follows: (1) Welcome the interviewee, (2) Explain the focus and intent of the interview, (3) Discuss the structured/predetermined interview questions, and obtain the interviewees' opinions/responses, (4) Thank the interviewees for their involvement, and (5) Close the interview.

Interviews were conducted to understand the specific communication issues experienced by CGIS employees. The researcher prepared a list of questions and explored a few general topics to assist participants in freely structuring a response. The interviews included questions formulated in identifying communication problems experienced by employees as well as any suggested improvement opportunities. Even though we did not structure our interview questions, based on a standard taxonomy of communication-related problems, we formulated some questions that were based on the PIECES check list. Even though Whetherby's PIECES checklist (Fatoni, Adi, & Widodo, 2020) is useful to diagnose information-system-related problems at an enterprise, the Information outputs ("I" in PIECES) category would highlight communication problems that relate to information outputs. We also used the main phases of software development as a means to elicit development phase-related communication problems, i.e. scope definition, problem analysis, requirements analysis, logical design, decision analysis, physical design, construction and testing, and installation and delivery (Bentley, Whitten, & Randolph, 2007).

The interview questions were as follows:

1. What are the existing areas of concern regarding communication between project stakeholders on a software development project? Can you provide evidence (documents, images, etc.) to support your claims?
2. In terms of Whetherby's PIECES check list, please provide evidence (documents, images, or "stories") of the Information Outputs.
3. I need to model the existing process context for communication between project stakeholders, including the client, on a software development project. Could you provide inputs regarding the process flow, the actor roles that are involved, the channels used for the communication, and the problematic areas in the existing process? Please provide evidence to support your claims about the problematic areas.

Data-gathering cycles were conducted on different days in April and May 2021. Interviews ranged between 11 and 45 minutes in duration. A total of fifteen (15) CGIS employees were interviewed, and interviewees were selected based on availability and involvement on software development projects. Employees ranging from junior

© University of Pretoria

level to management were interviewed, ensuring that employees that recently joined CGIS were also considered, as well as employees who have been working for the company for more than twenty years. From a management perspective, the following people were interviewed: (1) The operations manager, (2) The person who fulfilled the operations manager role for more than twenty years before them, (3) The ITC manager, and (4) The sales manager. In addition, the call centre team lead, a product owner, an account manager, the UI/UX specialist, a business analyst, four developers, and two project managers were interviewed.

Interviews were transcribed and were interpreted by using thematic analysis, in accordance with the guidance provided by Guest et al. (2012a). To represent emerging themes, codes were iteratively defined, developed, and applied or linked to the raw data. The interviews were not validated with an inter-coder agreement, since the *interviews are confidential*. The thematic analysis results are included in section 4.4.3.

4.4.3 Communication-related problems at CGIS

Thematic content analysis was used to identify prominent, perceived areas of concern regarding communication on software development projects at CGIS. Of the fifteen (15) interviews conducted, fourteen (14) referred to the requirements elicitation/analysis phase of the software development process when discussing their communication concerns and problems with the primary researcher. Although the first analysis of the interviews assisted the researcher in defining and understanding the communication problems in general, the interviews were reassessed to evaluate the communication problems in the requirements elicitation/analysis phase specifically, considering the number of interviewees that experienced concerns in this phase. The following should be noted: CGIS has internal projects where one internal department could be the 'client' and the rest of the company would deliver according to the requirements.

Codes were identified by reading the transcripts multiple times and deriving common themes from the interview data. A total of 33 codes were created. Using a bottom-up approach, the 33 themes are presented in a rudimentary taxonomy that is still grounded in theory, linking to the main categories of communication, identified by Littlejohn and Foss (2005). Considering that the researcher was unable to find a general framework for communication problems in literature in the search conducted (refer to section 4.4.1), a search was initiated for a different structure in which the *first version* of our theory-ingrained taxonomy could be created. The researcher decided to use The theories of human communication (Littlejohn & Foss, 2005) as a first draft, as it differentiated between communication *components* in a logical way and assisted the researcher in categorising the problems based on the fundamentals of communication itself.

The categories of human communication as presented by Littlejohn and Foss (2005) include the *elements of the communication model* (that consists of communicator, message, and conversation) and the *context of communication* (that consists of relationship, group, organisation, health, culture, and society). For the initial taxonomy, we did not apply the sub-categories for the main category *context of communication*. We reason that every enterprise differs in context and the contextual categories will also differ.

The themes, i.e., perceived communication problems at GCIS, were therefore categorised as depicted in Fig. 13. The *Communicator* is a unique individual with characteristics, and organises information into attitudes, beliefs, and values, which in turn affect behaviour (Littlejohn & Foss, 2005). Communicator problems are therefore communication problems as a result of an individual's shortcomings. The *Message* is text, or organised sets of signs, that have meaning for *Communicators*, and which individuals use strategically to achieve goals. (Littlejohn & Foss, 2005). Message problems are therefore communication problems as a result of the content, whether text, or other forms in which a message can be communicated, being transferred incorrectly/inefficiently between *Communicators*. The *Conversation* consists of individuals' social behaviour, and are processes in which *Communicators* coordinate or organise interaction, in ways that create coherent patterns of meaning (Littlejohn & Foss, 2005). Conversation problems are therefore communication problems as a result of inefficient interactions between *Communicators*. The main category *context of communication* includes problems that could not be categorised into the *elements of communication* category, relating to any other enterprise contextual variable that may impair communication, e.g., team members with weak domain knowledge or unavailability of stakeholders.

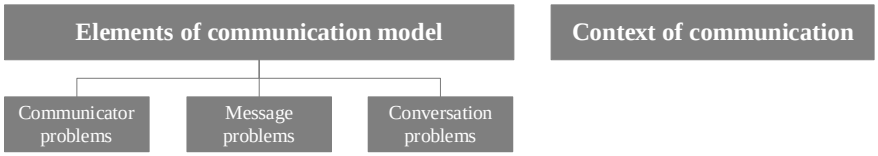


Fig. 13. The rudimentary taxonomy's structure, based on Littlejohn and Foss (2005)

According to Guest et al. (2012a), code frequency comparisons, code co-occurrence analyses, and graphical display of relationships between codes within the data set could be used to analyse the interview transcriptions.

© University of Pretoria

In Table IV we depict the absolute code frequencies in the transcribed interviews for the communication problems in general, and the communication problems in the requirements elicitation phase specifically. The absolute code frequencies were calculated as the total number of times a theme appears in the data set, acknowledging that the same theme/code is only counted once per interviewee, even if the theme was mentioned/addressed more than once in the interviewee's transcript.

The main communication problems identified *in general* are requirements that cannot be defined and/or translated, a lack of communication structure e.g., roles, responsibilities, and expectations, who needs to communicate what to whom etc., and insufficient information being provided during communication. The main communication problems identified *in the requirement elicitation/analysis phase* are requirements that cannot be defined and/or translated, misalignment between sales and operations regarding client requirements, and misalignment between the client and sales regarding requirements.

Table IV. Communication absolute code frequencies as analysed in transcribed interviews

Theory	Category	Problem	Absolute Frequency	
			General	RE Phase
Elements of Communication Model	Communicator	C1: Inadequate comprehension	3	2
		C2: Difference is psychological/personality traits	1	0
	Message	M1: Unnecessary requirements provided	4	3
		M2: Incorrect information provided	4	2
		M3: Insufficient information provided	8	4
		M4a: Lack of any documentation/requirements	5	2
		M4b: Outdated/inaccurate information in documentation	4	2
		M4c: Insufficient amount of relevant information in the documentation	4	4
		M4d: Irrelevant information included for a specific stakeholder	1	1
		M4e: Too much information in specification	4	1
		M4f: Lack of standard regarding content and volume in documentation	1	1
	Conversation	D1: Lack of body language/facial reactions in virtual communication	4	0
		D2: Asynchronous communication channels	4	0
		D3: Information is spread across different channels	2	1
		D4: Requirements cannot be defined/translated	8	8
		D5: Differences in native language	1	0
		D6: Differences in terminology/definitions	3	1
		D7a: Important information is obtained/provided too late	5	2
		D7b: Important information is obtained/provided too early	1	0
		D8: Different information is communicated to different stakeholders	2	1
		D9: Communication from client to development team takes time	2	0
		D10: Information is translated incorrectly if passed through many stakeholders	2	1
		D11: Misalignment between client/sales regarding requirements	6	5
		D12: Misalignment between sales/operations regarding capabilities (technical and operational)	4	3
		D13 Misalignment between sales and operations regarding client requirements	7	6
Context of Communication		X1: Absent client representative	1	1
		X2: Communication silos/islands of communication	5	4
		X3: All relevant stakeholders aren't involved in requirements elicitation	4	4
		X4: Different client representatives with different requirements	3	3
		X5: Unavailable stakeholders	2	4
		X6: Lack of communication structure/requirements	8	4
		X7: Inadequate context	7	2
		X8: Weak knowledge of application domain	4	1

The absolute code frequencies of the themes identified for communication problems *in general* during the execution of software development problems were compared to the absolute code frequencies of the themes identified for communication problems during the *requirement elicitation/analysis* phase in Fig. 14. An example is that D1, a lack of body language, was mentioned as a communication problem but could not be linked to the requirement elicitation/analysis phase specifically.

© University of Pretoria

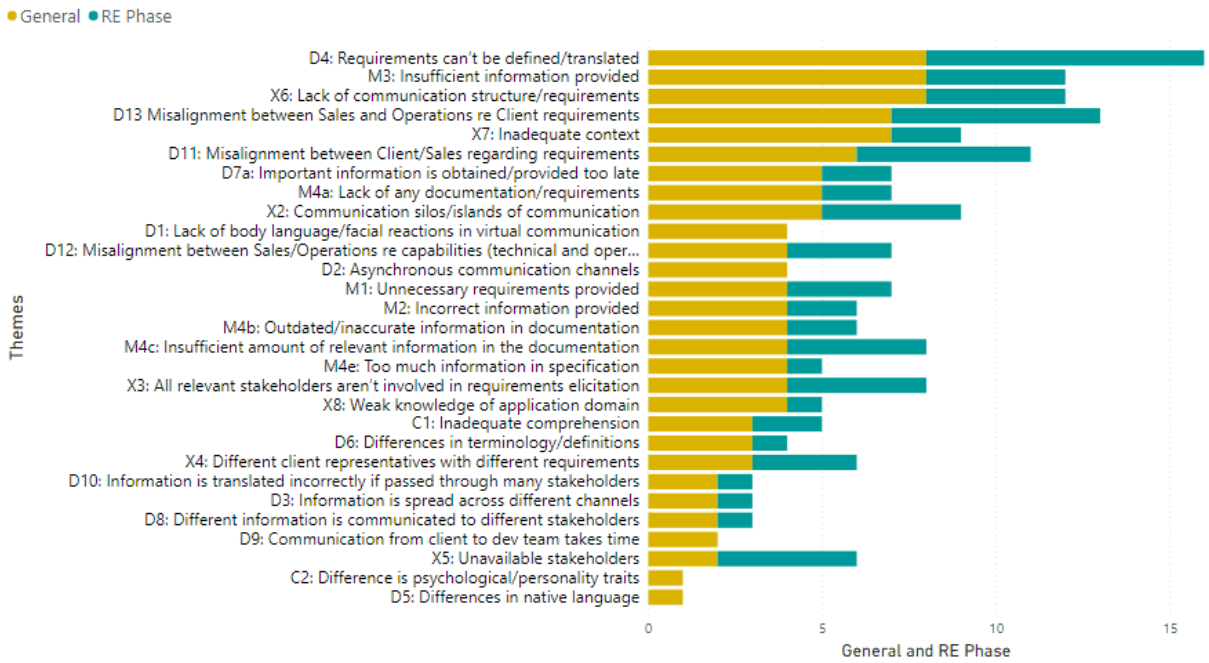


Fig. 14. Comparison of absolute frequencies between general and RE-related communication problems

The problems experienced by the interviewees regarding communication between stakeholders on software development projects is presented in Fig. 15. We therefore structured communication-related problems at CGIS in a theory-ingrained taxonomy. The purpose is to obtain a more in-dept understanding of the problems at CGIS that could also relate back to communication-related problems and associated solutions presented in literature. All problems in Fig. 15 were mentioned during the interviews, whereas the communication problems indicated in white blocks with no shading were not mentioned when analysing the *requirement elicitation/analysis* phase specifically.

We now present the insights extracted from the secondary research questions for *RQ3*.

RQ3.1: *What are the specific communication problems experienced at CGIS between project stakeholders on a software development project and how can these problems be structured?* Most interviewees, i.e., 53%, referred to requirements that cannot be defined and/or translated, a lack of communication structure/requirements, and insufficient information being provided. Most problems could be categorised as conversation problems. CGIS currently follows an approach somewhere between waterfall and Agile i.e., no specific Agile framework is currently implemented. In the Agile Manifesto (Beedle et al., 2001), a core value is *working software over comprehensive documentation* but concerns regarding inadequate documentation frequented the interviews. Another observation is that, in a country with eleven (11) official languages, a language barrier was only mentioned by one interviewee as a communication problem. A limitation of the interviews was however that fourteen (14) of the fifteen (15) interviewees were of the same race and spoke the same native language. This was due to the company's demographics, and due to the specific participants that agreed to take part in the research. The book, Theories of Human Communication (Littlejohn & Foss, 2005), was used to structure 33 themes/communication problems, demonstrating how we applied our theory-ingrained taxonomy at CGIS.

RQ3.2: What are the specific communication problems experienced at CGIS in the requirement elicitation phase of software development projects and how can these problems be structured? Communication problems in the requirement elicitation/analysis phase were mentioned by fourteen (14) out of fifteen (15) interviewees. The only interviewee that did not mention problems in the requirements elicitation/analysis phase was the call centre team lead and they are not typically involved in requirements elicitation/analysis. Most problems were related to requirements that cannot be defined and/or translated (8 interviewees), and misalignment between stakeholders (the Sales and Operations team at CGIS, and the client and the CGIS sales team). Of the 33 themes/communication problems identified, 27 could be classified as requirement elicitation/analysis problems (excluding M4 and D7 as these problems only group other problems).

The second cycle of the design stage was concluded with a better understanding of the communication problem at CGIS. In the following chapter, the *design* stage of ADR is discussed in which the main artefact of this study is designed between the primary researcher and CGIS practitioners.

© University of Pretoria

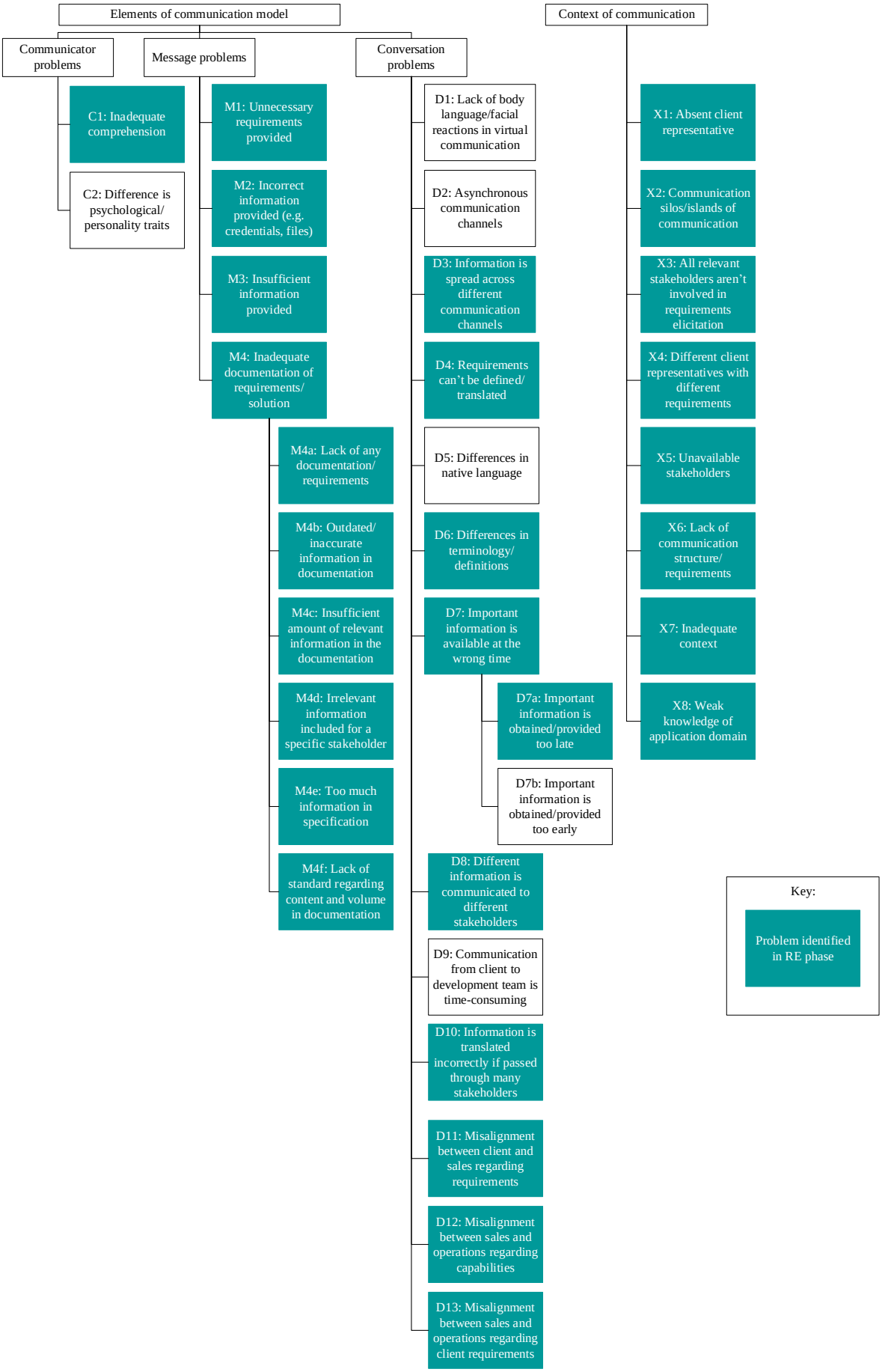


Fig. 15. Rudimentary taxonomy of perceived communication problems at CGIS

© University of Pretoria

Chapter 5– Agile RE solution (ARES) – main artefact design

The problem at CGIS was collaboratively diagnosed in Chapter 3 and Chapter 4. The ADR *design* stage (refer to Fig. 16, initially presented in section 2.2), which consists of designing a solution that could address the problem-class identified via diagnosis, is included in this chapter.

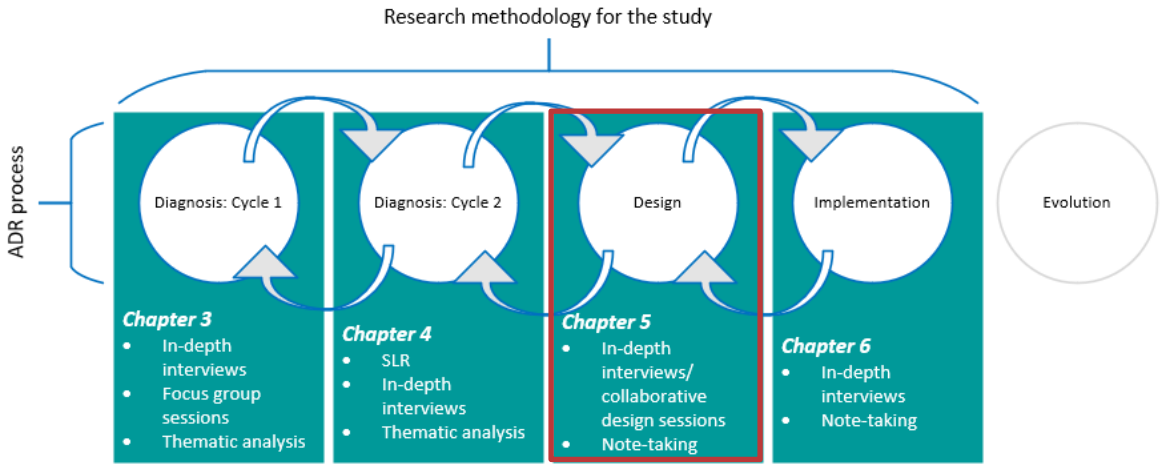


Fig. 16. Research methodology- Design

In the 15th State of Agile Report (Digital.ai, 2021), it was reported that the implementation of Agile has positively impacted visibility, business/IT alignment, delivery speed/time to market, and managing distributed teams. It was decided that an Agile framework could positively impact projects and stakeholder communication at CGIS, and our aim is therefore to implement an Agile framework and Agile RE practices to attempt to address the challenges identified at CGIS. The main artefact i.e., ARES that would address the problem, is presented in this chapter.

In section 5.1 we provide an overview of the differences between traditional and Agile RE, and a summary of Agile methodologies, frameworks, and practices from literature. The Agile RE solution presented to CGIS employees for feedback i.e., the requirements specification for developer approach in conjunction with Scrum, is included in section 5.2, and the modified solution, after discussing the proposed solution with CGIS employees, is included in section 5.3. The limitations of the main artefact design are included in section 5.4

5.1 Agile methodologies, frameworks, and RE practices

In this section we address *RQ4: What are existing Agile methodologies, frameworks, and Agile RE practices?* We began by comparing traditional RE to Agile RE in section 5.1.1. In section 5.1.2 we provide an overview of existing Agile methodologies, frameworks, and RE practices in literature.

5.1.1 Traditional RE vs. Agile RE

RE includes identifying, defining, communicating, and documenting a system's requirements as well as the context in which the system will be used (Paetsch et al., 2003). Traditional and Agile are two different software development approaches or philosophies that utilise values and principles. Within these approaches, there are different methodologies and frameworks, which prescribe certain practices. The waterfall model, spiral model, and V-model (refer to Shah, Jinwala, and Patel (2016)) are examples of traditional methodologies, whereas Scrum

© University of Pretoria

(Schwaber & Sutherland, 2020), Extreme Programming (XP) (Wells, 1999), and Kanban (Atlassian Agile Coach, 2021b) are examples of Agile frameworks. Agile scaling frameworks address problems associated with implementing Agile in a context it was not initially intended for, and examples include Large-Scale Scrum (LeSS) (Larman & Vodde, 2017), Scaled Agile Framework (SAFe) (© Scaled Agile, 2021), and Scrum of Scrums (SoS) (Atlassian Agile Coach, 2021a).

According to Sillitti, Ceschi, Russo, and Succi (2005), the assumptions underlying traditional RE include (1) The customer specifying all their needs in the initial phase, (2) One or more stakeholders being in charge of the requirements gathering activity, (3) The development team readily understanding customer needs, and (4) A sharp separation of different functions. According to Harned (2018), a traditional requirements document might contain use case diagrams, sequence diagrams describing events and the flow of information, data models detailing data and rules, user experience mock-ups and graphic design elements, and paragraphs and paragraphs of text. The document often contains many pages and requires a great deal of time and effort to compile. Traditional methodologies are based on preorganised phases, or stages of the software development lifecycle (Alhazmi & Huang, 2020). With regards to RE in traditional software development, there is a focus on comprehensively defining and understanding what should be built before system development starts (Alhazmi & Huang, 2020). In this approach, developing a system is executed in a sequential order, in which progress steadily flows downwards through the different phases (Curcio et al., 2018). The process begins with the elicitation activity where requirements and the system boundaries are discovered through the stakeholders (Curcio et al., 2018). Requirements must therefore be complete, consistent, and relevant at the beginning of software development. The traditional way of RE has been challenged by the rapidly changing business environment in which most organisations operate today (Ramesh, Cao, & Baskerville, 2010).

In the Agile Manifesto (Beedle et al., 2001), it is stated that *priority should be given to individuals and interaction over processes and tools*, *working software over comprehensive documentation*, *customer collaboration over contract negotiation*, and *responding to changes over following a plan*. RE is informal in ASD and is performed iteratively during the process rather than exclusively at the beginning of the process. As ASD usually works with small iterations consisting of frequent deliverables, the development process is flexible and dynamic (Curcio et al., 2018), with the role of RE still unknown in this environment. A few reasons to avoid spending a large amount of time on requirement documentation is that requirements are volatile, technical detail is often unknown and will affect implementation, and the customer can only clearly define the requirements once they see deliverables (Ramesh et al., 2010).

Literature on Agile methodologies, frameworks, and RE practises is presented in the following section.

5.1.2 Literature on Agile methodologies, frameworks, and RE practices

Agile frameworks, like Scrum (Schwaber & Sutherland, 2020) and Extreme Programming (Wells, 1999), focus on being adaptive to change and creating software iteratively. Agile frameworks are ideal for, and were originally designed for, small, self-organising, co-located, single-team projects and where customers are on-site as issues can be communicated and dealt with almost immediately (Dikert, Paasivaara, & Lassenius, 2016; Reifer, Maurer, & Erdogmus, 2003).

According to the 15th State of Agile Report (Digital.ai, 2021), Scrum was identified as the most popular Agile framework. Scrum was developed by Ken Schwarber and Jeff Sutherland and is considered to be a framework for developing, delivering and sustaining complex products (Mircea, 2019). This framework is iterative and incremental, and is based on three pillars of empirical process control which include transparency, inspection and adaption (Measey, 2015). Scrum is people-centric and adjusts to the characteristics of each environment (Measey, 2015). In order to define Scrum, all the roles, events, artefacts, and rules have to be known and understood (Mircea, 2019). The characteristics and features of Scrum, according to Flewelling (2018), are included in Table V.

Table V. The characteristics and features of Scrum

Characteristic/Feature	Scrum
Planning style	Empirical/adaptive
Delivery style	Iterative/incremental
Iteration length	Ranges from 1 to 4 weeks, most popular is 2 weeks
Values	Commitment, courage, focus, openness, and respect
Roles	Product owner, Scrum master, development team
Team size	Small, 5-9
Artefacts	Product backlog, sprint backlog, sprint progress tracking
Events	Sprint planning, daily Scrum, sprint review, sprint retrospective
Special features	All events are time-boxed
Lacks	A product/project/feature initiation phase and does not specify technical practices

© University of Pretoria

Extreme Programming, or XP, was conceived in the late 1990s to address the then-current methodologies' inability to deal with change (Measey, 2015) and quickly became the dominant Agile framework until Scrum took over in the early 2000s. According to Flewelling (2018), XP is the second most popular framework. Communication is ingrained into XP, with tasks such as planning and estimating, requiring stakeholders to work together and teams being encouraged to communicate through practices such as co-location and pair-programming. Some basic characteristics and features of XP, according to Flewelling (2018), are included in Table VI.

Table VI. The characteristics and features of XP

Characteristic/Feature	XP
Planning style	Adaptive
Delivery style	Iterative/incremental, sustainable pace
Iteration length	Ranges from 1 to 3 weeks, with a preference for the shortest possible
Values	Communication, simplicity, feedback, courage, and respect
Roles	Customer, development team
Team size	Small, 2-10
Artefacts	Release plan, iteration plan, user stories, tasks, and Class, Responsibilities, Collaborators cards (CRC cards)
Technical practices	Pair programming all production code, Test driven development (TDD), metaphor, refactoring, collective code ownership, Continuous integration (CI), daily builds, spikes, sustainable pace
Events	Release planning/iteration planning, daily stand-up
Special features	Prescribes technical practices, gathers requirements with user stories, promotes working at a sustainable pace
Lacks	Compromise - it is a fully committed, all-or-nothing approach

According to Measey (2015), common *Agile* practices include short feedback loops, face-to-face communication, daily stand-ups, show and tells, emergent documentation, visual boards, sustainable pace, and a focus on quality. In the Agile Manifesto (Beedle et al., 2001), one of the twelve principles directly relate to *requirements*: *Changes in requirements are welcome, even late in development.* Schön et al. (2017) show that a variety of different artefacts are applied in *Agile RE*, including user stories, prototypes, use cases, scenarios, and story cards. Inayat et al. (2015) provide a summary of 17 *RE practices* that have been adopted in ASD, which include face-to-face communication, customer involvement, user stories, iterative requirements, requirement prioritisation, change management, cross-functional teams, prototyping, testing before coding, requirements modelling, requirements management, review meetings/acceptance tests, code refactoring, shared conceptualisations, pairing for requirements analysis, retrospectives, and continuous planning.

Some of the challenges of traditional RE that could be addressed with Agile RE practises according to Alhazmi and Huang (2020) and Inayat et al. (2015) are:

- *Communication Gaps*: In traditional RE, there is an interruption in the provision of the required information to stakeholders.
- *Requirements Documentation*: In traditional RE, there are challenges regarding the reliability of documentation and/or the accuracy of the content, as documentation is often modified during the development process. Requirements are therefore often incorrect, outdated, and/or missing.
- *Customer Involvement*: Customer involvement is rare in traditional RE, where in Agile RE, customers remain aware of the progress during the development life cycle and provide timely feedback for the development team.

By using Agile RE practices, these challenges can be addressed. *Communication gaps* could be addressed by frequent face-to-face communication, collocated teams, an onsite customer, alternate customer representations, cross-functional Agile teams, and an integrated RE process (Inayat et al., 2015). *Requirements Documentation* could be addressed with face-to-face communication and tacit knowledge, and documentation which includes practices such as user stories and product backlogs (Inayat et al., 2015). *Customer Involvement* can be addressed as customers remain aware of the progress during the development life cycle (timely feedback is provided for the development team) and requirements are prioritised by the customer.

Agile RE practices however also pose several challenges. In Rasheed et al. (2021), challenges include less direct communication with clients/stakeholders, minimum focus on documentation, and missing, ambiguous, and conflicting requirements. Challenges identified by Inayat et al. (2015) include customer availability concerns, non-functional requirements being neglected, changing requirements, and minimal documentation. The lack of requirements specification/documentation is a challenge also supported by Medeiros, Vasconcelos, Silva, and Goulão (2020).

It can therefore be concluded that RE, regardless of traditional or Agile approaches, face communication challenges, specifically regarding customer involvement/participation, and missing and ambiguous requirements. Considering the literature on Agile methodologies, frameworks, and RE practises discussed, we present the selected Agile RE solution in the following section.

© University of Pretoria

5.2 Selected Agile RE solution

This section addresses *RQ5 i.e. Which Agile RE solution would be the most suitable solution for CGIS to address the communication challenges identified?*

Currently, CGIS use a traditional RE approach. Most customer needs are specified and documented at the beginning of the project, a business analyst and/or a business stakeholder gathers requirements, the development team uses a specification compiled by the business analyst and test cases compiled by the tester at the beginning of the project to develop the solution, and different functions have specific goals during the process. In addition to this, some Agile practices are used during the delivery of requirements at CGIS such as post-it boards and daily stand-ups.

The main communication problems identified at CGIS *in the requirement elicitation/analysis phase* are requirements that cannot be defined and/or translated, and misalignment between stakeholders. Considering that CGIS doesn't currently adhere to an Agile framework, and only incorporates some Agile practices, Scrum is the correct place to start (Measey, 2015). Scrum is simple to pick up because the framework gives you everything you need to get started and it is easy to add practices once the basics have been mastered (Measey, 2015). According to the 15th State of Agile Report (Digital.ai, 2021), Scrum is also the most popular Agile approach with 66% identifying it as the methodology they follow most closely. It was therefore decided that Scrum would be the most suitable framework to implement at CGIS, and the impact of this framework on communication problems would be investigated.

Ramesh et al. (2010), say that the term 'requirements engineering' is avoided in the Agile community as it is often taken to imply heavy documentation with significant overhead. In place of detailed upfront requirements, Scrum uses product backlog items (PBIs) as placeholders, which are discussed and progressively refined throughout the project (Rubin, 2017). While there is no standard format for a PBI in Scrum, the most common approach is user stories (Rubin, 2017).

User stories are widely adopted in Agile software development (ASD) as Software Requirements Specification (SRS), but user stories are often poorly written, are written in the language of the problem domain, and focus on the customer instead of design requirements (Medeiros et al., 2020), which results in a brief, vague, ambiguous and insufficient summary of the requirements. Curcio et al. (2018), also refer to challenges regarding user stories, including the lack of consistency and verifiability, and document maintenance.

Medeiros et al. (2020), created the Requirements specification for developer (RSD) approach to support the requirements specification activity in ASD and overcome the limitations of user stories. They report that the RSD approach assists in describing the requirements in a clear and objective manner, which improves the developers' productivity, and that the RSD approach describes design requirements through the systematic use of conceptual modelling, interface prototyping, and acceptance criteria, which addresses user stories' *lack of expressiveness to describe design requirements*.

Medeiros, Vasconcelos, Goulão, Silva, and Araújo (2017), experimented with the RSD approach and *XP*, but state in Medeiros et al. (2020, p. 2), that the RSD approach can replace other techniques used in ASD, such as user stories and use cases, and can be used with *"*XP, Scrum, or any other Agile method where the client validates the requirements through working software, as established in the Agile Manifesto*"*. It was decided that we would implement Scrum on a project at CGIS and use the RSD approach to create PBIs. It was therefore decided that the RSD approach would be used instead of user stories to attempt to address the shortcomings and challenges identified in literature with regards to user stories in requirements engineering.

The RSD approach and Scrum framework are elaborated on in section 5.2.1 and section 5.2.2 respectively.

5.2.1 The requirements specification for developer (RSD) approach

Medeiros et al. (2020), experimented with the requirements specification for developer (RSD) approach, built on the findings from a systematic mapping study (Medeiros, Alves, Vasconcelos, Silva, & Wanderley, 2015), the results from six case studies in industry (Medeiros, Vasconcelos, Silva, & Goulão, 2018), and the Agile Manifesto (Beedle et al., 2001).

Medeiros et al. (2018), investigate and discuss the characteristics which affect the quality of software requirements specifications (SRSs) in the context of Agile software development (ASD) and how those factors affect the work of software engineers in a cross-case analysis of six organisations. The goal was to evaluate the specification of requirements in different contexts and build an explanatory model of the factors that should be considered when writing more useful SRSs in ASD. The results indicated that (1) The SRSs in ASD should be directed to the development team, (2) The fragmentation of the description of the requirements in various artifacts compromise SRSs, and (3) SRSs are indirectly impacted by factors such as inadequate experience of the team, low customer availability, organisational factors such as the use of SRSs to validate requirements, late validations, and external factors such as contract agreements and characteristics of the developed software.

© University of Pretoria

The RSD approach was proposed, using the knowledge gained on how to write more efficient SRSs on agile projects, and experimented with by Medeiros et al. (2020). Two industrial case studies were conducted using a multiple-case design over 15 months. It was determined that the RSD approach met the developers' expectations and proved to be effective in producing a more objective SRS, suitable for coding activities. In addition to this, the results suggested that the RSD approach does not add extra effort for specification and may also reduce the effort for coding, testing, and maintaining software.

Medeiros et al. (2020), propose that frequent software deliveries, i.e., working software, should be used to validate requirements, rather than using SRSs. Documentation must therefore be tailored for the development team, and not the customer. The RSD approach can replace other Agile techniques used in ASD, such as user stories and use cases, while overcoming the limitations of these techniques, e.g., RSD includes technical aspects and consolidated information as well as non-functional requirements.

RSD provides an integrated view of the requirements by linking, in a systematic way, the problem domain concepts (conceptual modelling), the visual representation of interface requirements (mock-ups), the acceptance criteria composed of business rules, non-functional requirements, and technical constraints (Medeiros et al., 2017). Their objective is therefore to systematise the use of these practices in ASD, integrating functional and technical requirements in a single view, and therefore providing sufficient information required for coding (Medeiros et al., 2020).

From the fundamental practices, RSD incorporates *iterative development* and *incremental development*, whereas *iterations* and *frequent releases* are used from the XP practices (Medeiros et al., 2017). In an article published after Medeiros et al. (2017), Medeiros et al. (2020), state that Scrum can also be used to validate client requirements. RSD also includes three design practices, namely *Modelling Concepts, Modelling Mock-ups,* and *Acceptance Criteria (AC).* The AC also includes the *Behaviour Driven Development* (*BDD)* and *Acceptance Test Driven Development (ATDD)* testing practices (Medeiros et al., 2017). Only the design practices, i.e., *Modelling Concepts, Modelling Mock-ups,* and *Acceptance Criteria*, are elaborated on in this section, and *iterative development, incremental development*, i*terations,* and *frequent releases* are discussed in section 5.2.2.

Modelling Concepts describe the business concepts, which includes the entities, attributes, and relationships between entities and attributes. The objective is to model the concepts related to the requirements of a sprint, i.e., the requirements should not be analysed in isolation. *Modelling Mock-ups* are drawings that show how the user interface is supposed to appear during the interaction between the software and the end-user (Ricca, Reggio, Astesiano, Scanniello, & Torchiano, 2014) and the popular Agile methods do not typically include mock-ups as part of their process (Medeiros et al., 2017).

Acceptance criteria (AC) is based on the concept of the acceptance tests (AT) from XP which defines constraints for user stories. Since Medeiros et al. (2017) updated the concept of acceptance criteria (AC), they defined AC+ which can be reused by several requirements, is directed to the developer and not the customer, and can be written by any stakeholder. In addition to this, AC+ includes not only business rules, but also validation rules, interface, technical or any other type of constraint necessary for the system coding. A complete view of the differences between AC and AC+ is included in Medeiros et al. (2017). BDD focuses on the behavioural aspect of testing conditions and is usually written so that domain experts can understand the implementation rather than exposing the code level tests. In ATDD, stakeholders create one or more acceptance-level tests for a feature before implementing it, and therefore create concrete examples of business rules which clarifies requirements. BDD and ATDD only address functional requirements, whereas AC+ are defined under the developer's point of view, uses a developer-oriented language, and describes more than just the functional requirements (Medeiros et al., 2017).

The metamodel of RSD is depicted in Fig. 17, as detailed in Medeiros et al. (2017).

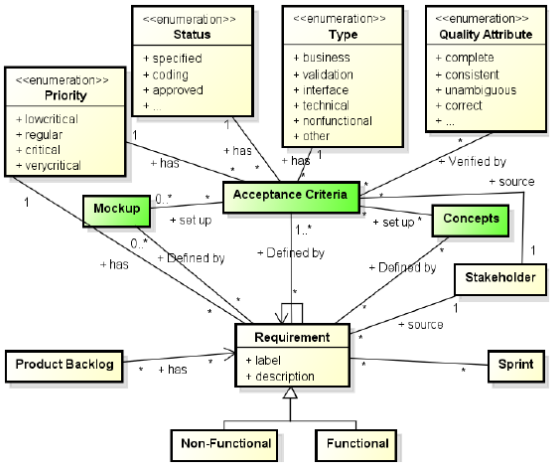


Fig. 17. The RSD approach metamodel (Medeiros et al., 2017)

© University of Pretoria

The RSD approach is detailed in Fig. 18, as described in Medeiros et al. (2017). The process starts with the conceptual model, then mock-ups and AC+ are defined in parallel. The coding of each requirement starts as soon as the AC+ and related mock-ups are specified.

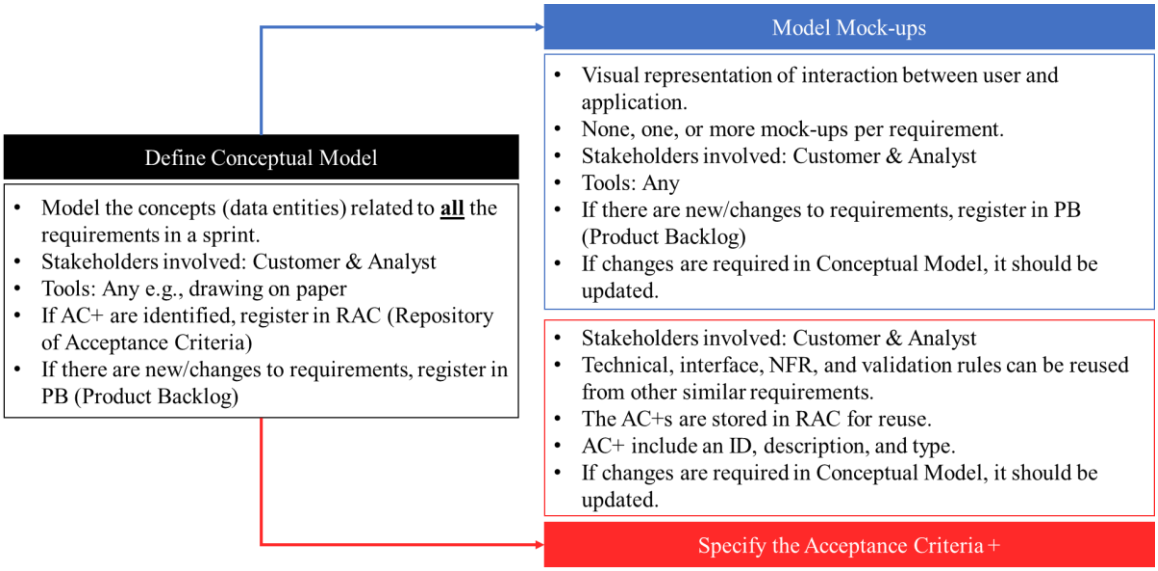
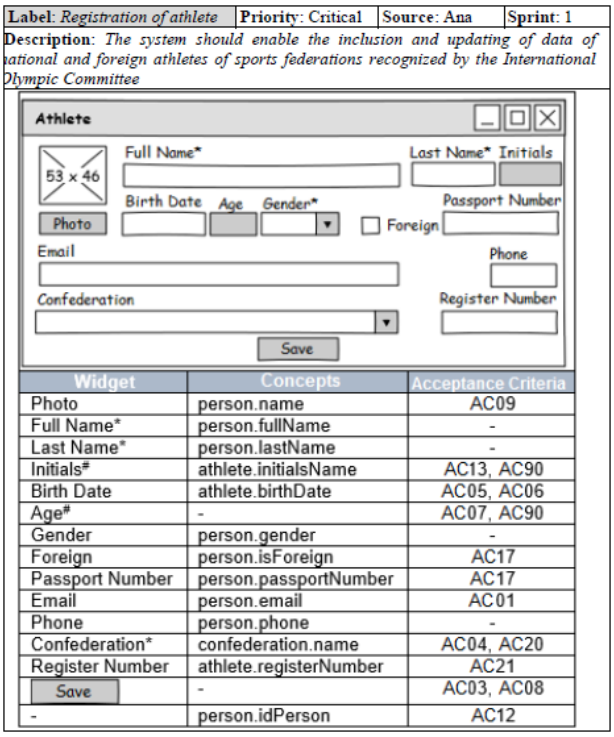


Fig. 18. Summary of the RSD approach

In Fig. 19, the RSD structure is divided into four (4) parts, namely the mock-up associated to the requirement if applicable, the widgets relevant to the mock-up, the data entities and attributes extracted from the conceptual model which relate to each widget, and lastly the AC+ related to the widgets and the data entities.



Widget	Concepts	Acceptance Criteria
Photo	person.name	AC09
Full Name*	person.fullName	-
Last Name*	person.lastName	-
Initials#	athlete.initialsName	AC13, AC90
Birth Date	athlete.birthDate	AC05, AC06
Age#	-	AC07, AC90
Gender	person.gender	-
Foreign	person.isForeign	AC17
Passport Number	person.passportNumber	AC17
Email	person.email	AC01
Phone	person.phone	-
Confederation*	confederation.name	AC04, AC20
Register Number	athlete.registerNumber	AC21
Save	-	AC03, AC08
-	person.idPerson	AC12

Fig. 19. RSD Example

The RSD approach is used to create product backlog items (PBIs) in our implementation instead of user stories as the RSD approach addresses user story limitations (Medeiros et al., 2020) and has the potential to produce a more objective specification, tailored for the development team. In addition to this, Medeiros et al. (2020) state that more evaluations of the RSD approach are still required to assess, for example, how the RSD approach works in different contexts.

© University of Pretoria

5.2.2 Scrum

Scrum is the selected Agile framework that will be used to validate client requirements through working software during the experiment at CGIS. The objective is to address inadequate communication between stakeholders regarding requirements.

There are three accountabilities in Scrum: (1) The *product owner*, responsible for *what* and *when*, The *Scrum master*, responsible for *facilitation, fostering a Scrum environment,* and *impediment management*, and (3) the *developers*, who typically include 2-6 *dedicated* developers, responsible for the *How* and *How Much*. Scrum teams are dedicated to one project/product and are cross-functional. Therefore, the team collectively has all the skills and expertise necessary to create value within each sprint (Schwaber & Sutherland, 2020). Skills and expertise are also shared between team members. In addition to this, Scrum teams are self-managing i.e., it is decided internally who does what, when, and how (Schwaber & Sutherland, 2020). There are no sub-teams or hierarchies, and the team is a cohesive unit of professionals focused on one objective at a time, the product goal.

Scrum incorporates an adaptive planning style or life cycle, also known as an agile or change-driven cycle. It is iterative or incremental, meaning that the scope is defined and approved before the start of an iteration and working software is delivered in small increments (Flewelling, 2018). Therefore, in the beginning, a high-level understanding of the requirements is obtained rather than a detailed specification of requirements, and serves as a starting point to plan the initial release cycle of the project (Ramesh et al., 2010).

According to Ramesh et al. (2010), the focus of RE in an agile environment is to effectively transfer ideas from the customer to the development team, rather than creating extensive requirements documents. Requirements are discussed with the customers before and/or during the implementation and the customer needs to be available and knowledgeable to provide the right requirements. In the RSD approach, the customer is involved in the conceptual modelling, mock-up modelling, and specification of AC+ so that their input is taken into consideration while creating the RSD for the developers. If the customer has limited availability, the team can start with the planning but must finalise all decisions with the customer present. The customer is also involved with validating the delivery.

Incremental delivery delivers small chunks of value early on and often to get feedback from customers to reduce uncertainty. This allows stakeholders to determine early on whether the right thing is being built (Flewelling, 2018), and enables stakeholders to quickly change direction, if required, based on new information which facilitates flexibility.

In the Scrum framework, there are five events: (1) The sprint, (2) Sprint planning, (3) Daily Scrum, (4) Sprint review, and (5) Sprint retrospective. Before the sprint starts, the product owner creates the product backlog, which is an ordered list of what is needed to improve the product and is the single source of work undertaken by the Scrum team (Schwaber & Sutherland, 2020). Scrum does not prescribe how the product backlog should be created and prioritised. The sprint then starts with sprint planning, a meeting where the whole Scrum team decides and forecasts which items from the already prioritised product backlog, can be achieved in the coming sprint (Flewelling, 2018). Next, the development team determines how the work will be done, decompose the product backlog items up into tasks, and plan the next sprint. The sprint backlog is then added to the team's Scrum board. According to Flewelling (2018), a Scrum board is usually a physical board.

Once the sprint is in progress, the Scrum team meets each day to coordinate the work in a time-boxed daily scrum of 15 minutes (Flewelling, 2018). Face-to-face communication and synchronisation meetings are essential (Measey, 2015). The board is updated based on the feedback, tasks are coordinated, and risks are identified.

The definition of *done* is a formal description of the state of an increment when it meets the quality measures required for the product, as determined by the Scrum team (Schwaber & Sutherland, 2020). The definition of *done* creates transparency by providing everyone a shared understanding of what work was completed as part of the increment (Schwaber & Sutherland, 2020).

At the end of the sprint cycle, a sprint review is conducted. The attendees include the Scrum team and product stakeholders, and the working software for each completed requirement is demonstrated by the developers (Flewelling, 2018). Based on feedback received, the product owner can adjust the backlog. The sprint retrospective usually follows immediately after the sprint review and is an opportunity for the Scrum team to inspect and adapt the process and identify actional improvements for the next sprint.

As part of the design stage, the existing constructs identified in this section are adapted by the researcher and the CGIS practitioners. The adaption is presented in the following section.

5.3 Adaptation of existing constructs

This section addresses *RQ 6: How (and why) should the selected Agile RE solution be adjusted/adapted for CGIS?* According to Mullarkey et al. (2019), collaborative intervention with co-creation activities is essential as the researcher–practitioner team creates designs that incorporate innovative ideas to solve the given problem(s).

© University of Pretoria

The researcher proposed a few changes to the Scrum framework as detailed in section 5.3.1, and then discussed the solution and proposed changes with CGIS practitioners. The research methodology and interview strategy used to determine how existing constructs (refer to section 5.2 and section 5.3.1) should be adapted/modified according to *CGIS practitioners' inputs* is introduced in section 5.3.2 and the interview results are included in section 0. In section 5.3.4 we present the adaption of existing constructs that will be implemented at CGIS and in section 5.3.5 we discuss the expected impact of the ARES on the communication problems at CGIS.

5.3.1 Adaptations proposed by the researcher

The researcher did not initially propose any changes to the RSD approach but suggested a few alterations to the Scrum framework (refer to section 5.2.2).

Before the sprint starts, the product owner should have created the product backlog, which is an ordered list of what is needed to improve the product and is the single source of work undertaken by the Scrum team (Schwaber & Sutherland, 2020). Scrum does not prescribe how the product backlog should be created and prioritised. In this implementation, it was proposed that the product owner should start by defining the product goal, and ensure that the goal is aligned to the CGIS strategy, with a model recommended by Moore (1999) as depicted in Fig. 20.

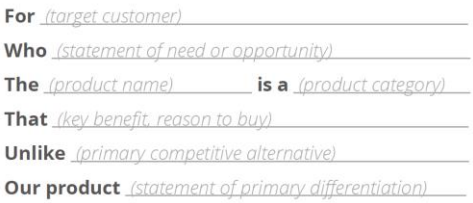


Fig. 20. The product goal model (Bastow, 2015; Moore, 1999)

Once the product goal statement is defined by the product owner, a holistic view of the required product features to achieve the product goal, should be created and prioritised. In Scrum, the sprint then starts with sprint planning, a meeting where the whole Scrum team decides and forecasts which items from the already prioritised product backlog, can be achieved in the coming sprint (Flewelling, 2018). Next, the development team determines how the work will be done, decomposes the PBIs up into tasks, and plans the next sprint. The sprint backlog is then added to the team's Scrum board.

Although a Scrum board is usually a physical board (Flewelling, 2018), an electronic Scrum board was used in this implementation as most employees still work remotely due to COVID-19 precautions. The researcher proposed that a Mural board (Tactivos Inc. dba MURAL, 2021) should be used to facilitate all Scrum practices and events after being introduced to Mural in a Scrum master certification course. Mural is a digital visual collaboration tool where teams can contribute by adding notes, comments, and drawings on a whiteboard as depicted in Fig. 21.

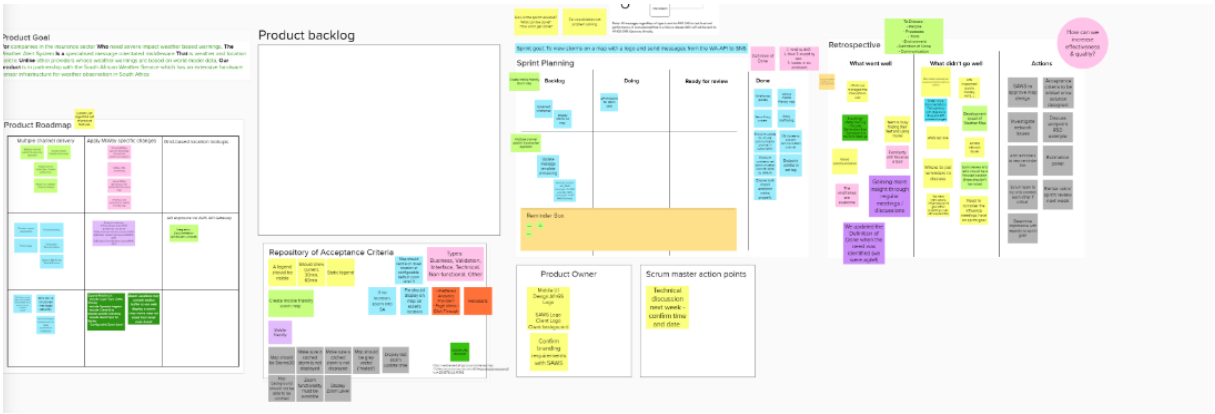


Fig. 21. Mural board example

Considering that the RSD approach would be used to elicit and document requirements, RSD was added to the Scrum board and not user stories, which are often used in Scrum. The Scrum board would be discussed daily in the daily Scrum, and the researcher suggested that the Scrum team either attended the daily Scrum at the CGIS

© University of Pretoria

office, or switch on their camera for the duration of the daily Scrum to encourage "face-to-face" communication as best as possible, while being respectful of social distancing.

The researcher also initially proposed that *swarming*, *planning poker*, and *automated testing* should be used in the implementation based on recommendations made during a Scrum master certification course attended by the researcher. *Swarming*, which occurs when as many team members as possible work simultaneously on the same priority item until it is done (Sutherland, 2020), could be used to ensure that the highest priority is completed first. *Planning poker*, an agile estimating and planning technique that is consensus based (Mountain Goat Software, n.d.), could be used to ensure that the Scrum team members are aligned with regards to estimates. *Automated testing* could be used to optimise testing during delivery.

This section included adaptions to the solution, i.e., the existing constructs, proposed by the primary researcher of the study. To ensure that the solution is collaboratively designed with the CGIS practitioners, the initial solution and proposed adaptions were discussed with three key stakeholders that had agreed to partake in the design and implementation stages of this study. The *process of interviewing* these three CGIS practitioners, including the interview questions, is detailed in the following section.

5.3.2 Adaptation of existing constructs based on interviews

The proposed solution (refer to section 5.2 and section 5.3.1) was discussed with three key stakeholders at CGIS before the solution was implemented, namely a QA employee, i.e. a tester, the development unit manager, and the product owner of the selected project. All three employees have worked for CGIS for more than eight years.

Some Scrum experts believe that the lack of automated testing is a common pitfall in Scrum (Platinum Edge Agile Experts, 2021). Automated testing is not currently implemented or used at CGIS, and the researcher wanted to obtain the tester's feedback, specifically with regards to the possibility of implementing automated testing. At CGIS, employee time is scheduled one week at a time. Therefore, the researcher would only know which employees would be available for the selected project i.e., the main artefact implementation project, at a later point in time. The development unit manager was therefore selected to participate, as he would guide the developers selected for the project at a later point in time. Lastly, the selected project's CGIS product owner was included, as he would be the product owner in the implementation. Although CGIS currently has a product owner role, the expectations and accountabilities vary significantly from the Scrum accountabilities (refer to Schwaber and Sutherland (2020)). It was therefore important to ensure that the product owner understands the proposed solution and expectations and can provide input into the design of the solution.

The three participants were interviewed individually. The sessions started with the researcher presenting a Microsoft PowerPoint presentation on the suggested solution, which included an introduction on the study, a summary of the problem identified (refer to section 4.4), a summary of the RSD approach (refer to 5.2.1) and Scrum (refer to section 5.2.2), the proposed process to be followed during project implementation, as well as some proposed *ground rules* and proposed changes (refer to section 5.3.1). The Scrum Guide (Schwaber & Sutherland, 2020) was also referenced as and when required.

Aligned with the participative nature of action design research (ADR), participants were encouraged to ask questions during the presentation and encouraged to collaboratively decide on and design a solution. They were also encouraged to disagree with the proposed solution or any aspect thereof. At the end of the presentation, the following questions were asked:

1. Which components of this proposed solution are already being used, according to your knowledge, within the enterprise?
2. Which components of this proposed solution do you think could address the existing areas of concern regarding communication between project stakeholders on a software development project? Please provide reasons for your answers.
3. Which components of this proposed solution, if any, do you think could not address the existing areas of concern regarding communication between project stakeholders on a software development project? Please provide reasons for your answers.
4. What are reasons, if any, that you think the proposed solution would not be feasible/be able to be implemented within your enterprise's context?

The sessions were recorded, and the researcher was able to take notes and reference the recordings as necessary. The results of the interviews with the three CGIS practitioners are included in the following section. The feedback obtained was incorporated to create a collaborative researcher-practitioner solution.

5.3.3 Adaptation of existing constructs results

The development unit manager thought that the implementation could "definitely work" and supported the experiment, stating that he wanted to experiment with similar practices at CGIS for a while, but making a change

© University of Pretoria

is challenging *(*"like turning the Titanic*"*). The tester believed all suggestions made in the proposed solution could be beneficial and was especially excited about the Scrum *retrospective* practice being implemented, as CGIS currently does not review the project to determine what was fixed and how fast it was fixed. According to the tester, CGIS currently only cares about whether the deadline was met and whether the team stayed within a budget that was defined by somebody else. The product owner was enthusiastic about the proposed solution, especially with regards to the definition of *done* being implemented. He stated his belief that everyone in the Scrum team is equal, and that the thinking and designing will be a collaborative effort to ensure that everyone is on the same page.

The product owner stated that *daily stand-ups* and having sessions in which the entire team designs a solution together are components of the proposed solution that are already being used at CGIS. The tester stated that there are similarities between the proposed solution and what is currently being used at CGIS, but that "it doesn't mean much" if the Scrum process isn't followed from start to finish. According to the tester, the *daily stand-up* currently used at CGIS, is a daily meeting between all project managers and team managers. It is problematic that the meeting is not project specific, and it does not order or prioritise items.

The tester supported *automated testing* being implemented but warned that addition to an already existing solution/product could cause complications with regards to the set-up. The development unit manager acknowledged that implementing automated testing on an existing system could be a challenge, but did not foresee concerns on the specific project selected, as the product is built on Vue.js, a JavaScript Framework (You, 2021) which has automated testing support built in. It was recommended by the development unit manager that Selenium, testing software that has been experimented with by the CGIS tester in the past, should be avoided as the team could experience issues if the HTML elements did not contain IDs. Cypress, a next generation front end testing tool (Cypress.io, 2021), was therefore recommended for end-to-end automated testing on this specific project. The product owner indicated that the team should attempt to implement automated testing and swarming ("stretch ourselves"), even if experimentation is limited to these two components on specific sections/requirements.

With regards to the RSD approach, the product owner mentioned that it is important to consider, and find the balance between, both the customer's and the developers' requirements. A concern raised by the development unit manager is that the RSD approach could hinder developers from providing their input and creatively solving the problem if the RSD is dictated/created by a business analyst.

The tester was concerned regarding the *dedicated team*, as some employees were "pillars in the company" that could be essential on other projects during the implementation. The product owner was also sceptical that a dedicated team would be possible for this implementation. The development unit manager said that he proposed that the development teams should be split into separated, dedicated teams in the past, and this experiment could therefore support, and accelerate such a change within the teams.

The tester required clarity on whether he would be included in the *developer accountability* of Scrum. CGIS currently differentiates between developers and testers, i.e., two different functions within CGIS. The tester stated that the developer accountabilities as detailed in the Scrum guide (Schwaber & Sutherland, 2020), are to some extent similar to what is expected and what is currently being used by the CGIS developers. He also stated that the *product owner accountabilities* as detailed in the Scrum guide (Schwaber & Sutherland, 2020), and the current CGIS product owner responsibilities and accountabilities are very different, since the CGIS product owner does not prioritise backlog items. The product owner enquired whether the project Scrum team would be trained before implementation started i.e., whether information would be provided on expectations, accountabilities, and Scrum in general.

A concern raised by the tester is that all members in the Scrum team have different perspectives and would therefore evaluate the problem and requirements in isolation, e.g., the Scrum master, product owner, and developers will all have different expectations for estimates. Another concern raised by the tester was that the client, i.e., the product owner in this experiment, would change the requirements often, regardless of the fact that the team meet frequently. The product owner stated that the "human effect" would always be a challenge, e.g., people raising concerns/questions in the wrong communication channels, team members not speaking up for a long time, a lack of a sense of urgency, etc. The development unit manager stated that, based on his experience on a Scrum team, "Scrum fails before it works". It was advised that the researcher should become acquainted with the process of *forming, storming, and norming*, and that the focus should not be on "we have to make this work". He also advised that the Scrum team and specifically the product owner be informed that the first few weeks of implementation could be challenging.

The feedback obtained from the CGIS practitioners was incorporated into the proposed artefact named the ARES, the Agile Requirements Engineering Solution. The ARES is presented in the following section.

5.3.4 The proposed artefact named ARES

The ARES is a combination of the Scrum framework and additional Agile practices, such as the use of the RSD approach, encouraging face-to-face communication, and defining the product goal statement before creating the

© University of Pretoria

product backlog. A graphical representation of the ARES, including the process and Agile practices, is depicted in Fig. 22. The colour-coding in Fig. 22 aims to distinguish the Scrum components from the other Agile practices incorporated into the ARES.

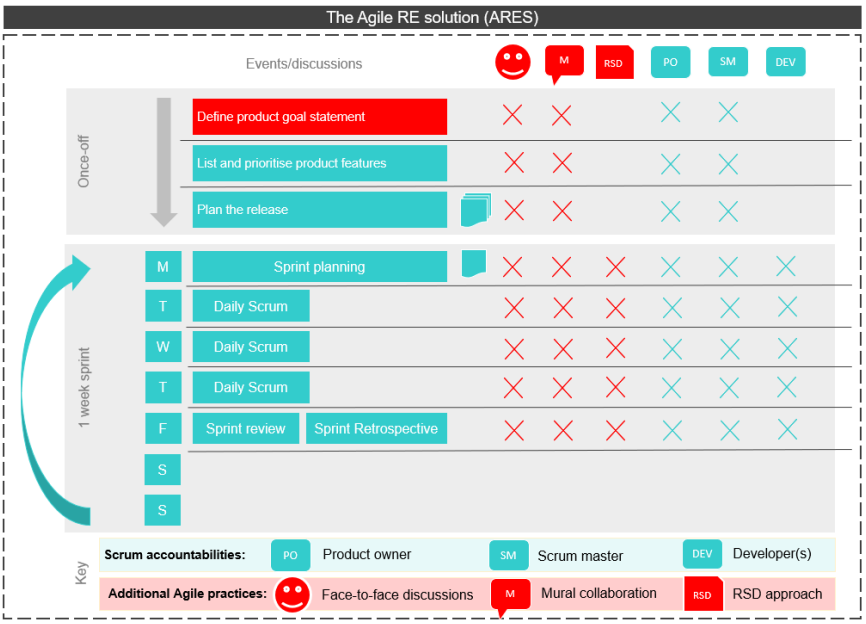


Fig. 22. Graphical representation of Agile RE solution (ARES)

Some of the practices initially proposed by the researcher and discussed with the CGIS practitioners (refer to section 5.3.1-0) were *not* implemented due to time restraints and complexity. The practices that the researcher believed could improve communication but were not implemented included *swarming* and *planning poker*. In addition to this, *automated testing* was not implemented within the first two weeks.

It is important to note that product planning, i.e., the product roadmap, which was prescribed to ensure that the product owner understood the desired functionality, and that it was aligned to the company's strategy, were practices implemented to enhance and improve the *efficiency* of Scrum but were not implemented to address the communication problems at CGIS specifically.

The ARES was designed in collaboration with CGIS practitioners to address communication challenges in CGIS. The researcher expected the ARES, and certain components of the ARES, to address different communication challenges. The expected impact of the ARES on the communication problems identified at CGIS is included in the following section.

5.3.5 The expected impact of the ARES on communication

The expected impact of the ARES on communication problems, as detailed in section 4.4, is now discussed.

The aspects of Scrum that we expected to improve communication included the collaborative and frequent *planning and feedback sessions*, e.g., sprint planning, the retrospective, and sprint review, and the specific *accountabilities within the Scrum team*. It was also believed that the daily Scrum, with the *"face-to-face" discussions*, would improve communication. We believed that these aspects could address inadequate comprehension (C1), incorrect information being provided (M2), important information being obtained/provided too late (D7a), different information being communicated to different stakeholders (D8), information being translated incorrectly if passed through many stakeholders (D10), inadequate context (X7), and communication silos/islands of communication (X2). The *dedicated team* should address stakeholder unavailability (X5), and the *product owner's accountabilities,* and the fact that the product owner would also be the client in the selected project, would address absent client representatives (X1).

As a team, we did not expect that misalignment would be removed completely by implementing Scrum and the RSD approach. We did however think that relevant stakeholders would collaborate on designing the solution and frequently review working software, and therefore expected misalignment to be identified earlier and therefore be addressed earlier. The sales team was not involved in the selected project as it was an internal project, and the product owner was therefore also the client. The experiment would not evaluate misalignment between the client and the sales team regarding requirements (D11), misalignment between the sales team and operations regarding capabilities (D12), or misalignment between the sales team and operations regarding client requirements (D13).

© University of Pretoria

In Scrum, the *Scrum master* is responsible for the Scrum team's effectiveness (Schwaber & Sutherland, 2020). A few examples of how this is done, is by coaching the team members in self-management and cross-functionality, helping the Scrum team focus on creating high-value increments that meet the definition of *done*, removing/avoiding impediments to the Scrum team's progress, and ensuring that all Scrum events are positive, productive, and kept within the timebox (Schwaber & Sutherland, 2020). The Scrum master's accountabilities could have the potential to address the following communication problems:

- Information is spread across different communication channels (D3): The Scrum master would ensure that team members understood which communication channels were applicable and adhered to the standard agreed on between Scrum team members.
- Differences in terminology/definitions (D6): The Scrum master would ensure that team members were referring to the same terminology and/or definition during Scrum events.
- Different information being communicated to different stakeholders (D8): The Scrum master would ensure that all stakeholders were involved when project information was being communicated
- Addressing the lack of communication structure/requirements (X6): The Scrum master would ensure that the Scrum team was aware of their accountabilities and expectations regarding communication.

Regarding the *RSD approach*, requirements were decomposed into smaller sections, rather than one large document that included all requirements at the beginning. Furthermore, requirements were always required per PBI. The developers and client, i.e., the product owner, were involved in defining and refining the RSD collaboratively, using a standard with regards to the RSD that was agreed on between the Scrum team members at the beginning. The following communication problems could potentially be addressed by the RSD approach: Inadequate comprehension (C1), insufficient information being provided (M3), a lack of documentation/ requirements (M4a), outdated/inaccurate information in documentation (M4b), an insufficient amount of relevant information in the documentation (M4c), irrelevant information being included for a specific stakeholder (M4d), too much information being included in the specification (M4e), no standard regarding the content and volume in documentation (M4f), requirements not being able to be defined/translated (D4), and all relevant stakeholders not being involved in requirements elicitation (X3).

We summarise the expected impact of the ARES on the communication problems (section 4.4) in Fig. 23.

Communication problem in RE	Scrum events	"Face-to-face" discussion	Dedicated team	Scrum master role	The RSD approach
C1: Inadequate comprehension					
M1: Unnecessary requirements provided					
M2: Incorrect information provided					
M3: Insufficient information provided					
M4a: Lack of any documentation/requirements					
M4b: Outdated/inaccurate information in documentation					
M4c: Insufficient amount of relevant information in the documentation					
M4d: Irrelevant information included for a specific stakeholder					
M4e: Too much information in specification					
M4f: Lack of standard regarding content and volume in documentation					
D3: Information is spread across different channels					
D4: Requirements cannot be defined/translated					
D6: Differences in terminology/definitions					
D7a: Important information is obtained/provided too late					
D8: Different information is communicated to different stakeholders					
D10: Information is translated incorrectly if passed through many stakeholders					
D11: Misalignment between client/sales regarding requirements					
D12: Misalignment between sales/operations regarding capabilities					
D13 Misalignment between sales and operations regarding client requirements					
X1: Absent client representative					
X2: Communication silos/islands of communication					
X3: All relevant stakeholders aren't involved in requirements elicitation					
X4: Different client representatives with different requirements					
X5: Unavailable stakeholders					
X6: Lack of communication structure/requirements					
X7: Inadequate context					
X8: Weak knowledge of application domain					

Key:	
	Expected to be addressed by ARES
	No impact was expected/realised

Fig. 23. Expected impact of ARES on communication problems

The primary researcher expected the main artefact designed, i.e., the ARES, to have an impact on the communication problems identified at CGIS, but certain limitations may pose threats to the validity of the design. These limitations are discussed in the following section.

© University of Pretoria

5.4 Limitations of the main artefact design

Only three CGIS practitioners as well as the researcher were included in the ARES design process. Therefore, only a limited number of participants and roles were included in the design process. Other stakeholders and demographics and/or roles such as .net developers and the operations manager could have contributed valuable insights and considerations if they had been included.

CGIS does not currently implement the ARES, or any component thereof. The Scrum master, who is also the researcher, was the only team member that had obtained formal Scrum training. The lack of knowledge and experience could have had an impact on the design of the main artefact.

© University of Pretoria

Chapter 6–ARES – main artefact implementation

The main artefact presented in the previous chapter, i.e., the Agile RE solution (ARES), is a combination of the Scrum framework, the RSD approach, which replaces traditionally used user stories, as well as a few other Agile practices. In this chapter, we detail the ADR *implementation* stage (refer to Fig. 24, initially presented in section 2.2) with the objective of evaluating the impact of the ARES on the communication problems identified at CGIS (refer to Chapter 3 and section 4.4). . The implementation and interviews to evaluate the implementation took place during October and November in 2021.

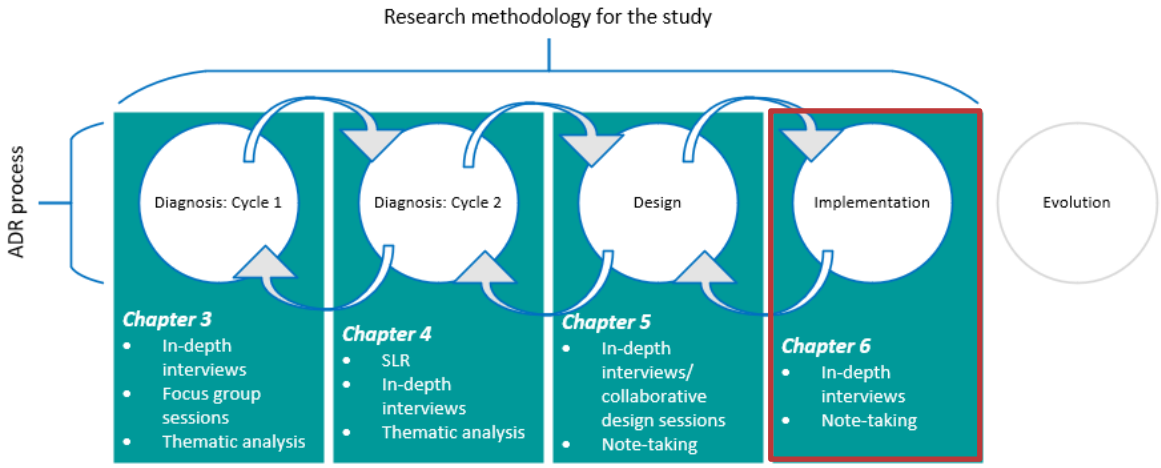


Fig. 24. Research methodology- Implementation

In section 6.1 we provide an overview of the implementation. In section 6.2 we include the results obtained after discussing the implementation with key stakeholders, which includes the impact of the solution on communication challenges as well as the deficiencies of the implemented solution. Limitations of the implementation are discussed in section 6.3.

6.1 Implementation summary

In this section, we address *RQ7*: *How were the Agile RE components implemented?* We provide an overview of the implementation by describing the CGIS project selected for the implementation and the team involved (section 6.1.1), how the Agile RE components were implemented on the selected project (section 6.1.2), and observations made by the researcher during the implementation (section 6.1.3).

6.1.1 The selected project and Scrum team

The project selected included a combination of internal and external requirements, i.e., there was an internal CGIS client as well as an external, paying client. It was however decided that only the internal client, which is also the Scrum product owner, would be included in the Scrum and would therefore also represent the external client in the implementation. This decision was made as we did not want to expose the client to an experimental process in which most team members were still familiarising themselves with the Scrum framework.

The Scrum team consisted of five members, including the product owner, the Scrum master, and three developers, of which two were .net developers and one was a QA developer. A graphic designer/UX specialist attended one of the sprint planning events to assist with a wireframe for one of the requirements. Of the five Scrum team members, one had been exposed to Scrum and had been part of a Scrum team as a developer at a previous company, although he stated that it was an adjusted version of Scrum. Another team member had worked in an

© University of Pretoria

Agile environment before, but many years ago. Two team members had not worked or previously been exposed to an Agile or Scrum environment. Lastly, the Scrum master had obtained Scrum certification, but had not previously been exposed to, or implemented, an Agile or Scrum framework. Therefore, the team was unfamiliar with Scrum.

Of the Scrum team members, excluding the researcher, three were interviewed in the diagnosis phase (refer to Chapter 3), two were interviewed in the design phase (refer to Chapter 5), and all four were interviewed in the implementation phases (refer to section 6.2).

The implementation of the ARES at CGIS is discussed in the following section.

6.1.2 The implementation of the ARES

Scrum teams are typically small, consisting of 5-9 team members. In the 2020 Scrum Guide (Schwaber & Sutherland, 2020), the authors stated that smaller teams communicate better and are more productive. Once the Scrum team, consisting of five team members was identified, the Scrum master conducted a training session in which everyone's accountabilities and expectations, as well as the Scrum framework and other information were discussed. The ARES was then implemented on the selected project at CGIS to investigate whether the communication problems identified (refer to 4.4) could be reduced/addressed with ARES as expected (refer to section 5.3.5).

The implementation of the ARES started on the 20th of October, with the product owner and the Scrum master discussing the product goal, i.e., product planning, and creating the product roadmap on Mural. The first sprint started on the 25th of October 2021. The team decided on using one-week sprints, since CGIS employees generally schedule resources on a Friday for the next week. Even though the Agile RE solution would be implemented over the duration of the project, only two sprints (i.e., two weeks) could be evaluated and included in this study due to time constraints.

A two-hour sprint planning session was scheduled and conducted on Monday mornings. The RSD, including the conceptual model, mock-ups, and acceptance criteria, were created collaboratively between the product owner and the developers, with the assistance of the Scrum master/researcher. An analyst was therefore not involved in defining and constructing the RSD.

All members had access to and could contribute to the Mural board. On a Monday morning, the previous week's Mural board was duplicated, ensuring that there was traceability of the previous weeks' notes, action points, and other information. The second week's Mural board is depicted in Fig. 25. The team members could easily zoom in and out of different sections, add notes where required, and move tasks on the Scrum board. In addition to the Mural board, a Microsoft Teams group consisting of the Scrum team members was used for communication and document sharing, and GitLab (GitLab B.V., 2021) was used by the developers for more detailed issue tracking, continuous integration, and deployment pipeline features. GitLab is the prescribed tool at CGIS.

Fig. 25. The second week's Mural board with labels

The team attended a 15-minute daily Scrum on Tuesdays, Wednesdays, and Thursdays, and an hour session on a Friday afternoon, in which the sprint was reviewed followed by a retrospective. All five primary Scrum team members attended the events and all team members' cameras were on in *all* daily Scrums as well as most of the other events.

The developers showed the rest of the team what they had been working on during the week in the sprint review, and the team then collaboratively decided whether the definition of done was met or not. If the requirement, i.e., sticky note, adhered to the definition of done, the ticket was moved to the relevant column. After the first sprint,

© University of Pretoria

the team decided that an additional column called *Ready for review* should be added where they could move tickets that were ready to be demonstrated in the sprint planning session. Therefore, the Scrum board had a section for *backlog*, *doing*, *ready for review*, and *done*. The definition of done was amended after the first sprint during the retrospective.

The retrospectives were also conducted on the Mural board. Mural has a "private mode" in which members that have access to the board can contribute to the board and the content is hidden until the "private mode" is terminated. The content was revealed once everyone had added their tickets, but the authors of the content remained anonymous.

Of the five primary Scrum team members, only one was fully dedicated to the project in the first two sprints. There were unfortunately other responsibilities, e.g., production issues on other products and projects that could not be moved or avoided during the implementation. The Scrum master did however remove impediments as best as possible.

During the implementation of the ARES, the primary research participated in the setting and was able to be involved as the Scrum master. This allowed the researcher to experience reality as the participants do. The observations made by the researcher are discussed in the following section.

6.1.3 Observations made by the researcher

This section includes feedback from the first two retrospective events, as well as observations made by the researcher during the first two sprints before the interviews were conducted with the other four team members.

Table VII depicts feedback relevant to this study obtained in the first two weeks. The author of the feedback remained anonymous unless the author wanted to elaborate on a statement. Please note that the Scrum master, and therefore researcher's, feedback is excluded from Table VII.

Table VII. Retrospective feedback for first two sprints

Week	Retrospective section	Feedback obtained
1	What went well	• "Daily sprints helped and open communication" • "Working together as a team with a single goal" • "Feels like we're a team and are working towards the same goal" • "Regular check-ins" • "Daily scrums work well"
	What didn't go well	• "Not ideal to get more info after some implementation" • "No active or dedicated time for planning/QA structure" • "Other/intervening project, issues or meetings"
2	What went well	• "Everything! Better planning, focused deliverables/goal, concise and to the point meetings" • "Good communication" • "Gaining more insight through regular meetings/discussions"
	What didn't go well	• "Prefer more documentation/transparency with regards to update/changes" • "Need to consider the influence meetings have on sprint goal"

According to the researcher, the Scrum team was eager to try the Scrum implementation, i.e., try something else than currently implemented and to learn something new. The Scrum master had to train, and frequently remind the Scrum team, on their accountabilities. The researcher suspects that this is due to the current culture and roles at CGIS that often conflict with the accountabilities recommended by the Scrum framework, e.g., the *project manager* is often responsible for making decisions and prioritising items instead of the product owner and the developers are often prescribed what to do rather than having the freedom to creatively solve problems on their own.

The RSD approach was an adjustment for the team and would take more time to get used to. In addition to this, the time dedicated to planning sessions was often insufficient to elaborate and discuss the RSD and the related time estimates. There were still many questions and uncertainties regarding Scrum and accountabilities, which took up more time than expected. The researcher predicts that the planning sessions will become more efficient as the team settles into the new framework. Finding the balance between too little and too much detail for the RSD was also challenging. Some developers expected more while some expected less, for example, the QA developer expected more detailed acceptance criteria that could be "ticked off" during the sprint review to ensure that *all* expectations were met, whereas the .net developers were content with less detailed RSD and acceptance criteria.

How diligently the system would be tested and by whom was also a topic which reoccurred and had not been sufficiently decided on by the end of the second sprint. CGIS currently has a dedicated tester that ensures that the software is sufficient and meets the test cases specified at the beginning of the project.

© University of Pretoria

Although we tried to ensure that the team was as dedicated to this project as possible, there were important system migrations in CGIS that took place in the second week of the Scrum implementation which impacted the Scrum team's dedication and time. Considering that the rest of CGIS were continuing as usual, it was difficult to remove environmental and organisational noise.

Interviews were conducted with CGIS practitioners, i.e., the Scrum team involved in the ARES implementation, to evaluate their perception of the impact of the ARES on communication during software development. The implementation's results are discussed in the following section.

6.2 Implementation results

CGIS practitioners were interviewed after two sprints were completed (section 6.2.1) to determine the impact of the solution on the communication problems (section 6.2.2) as well as the deficiencies of the implemented solution (section 6.2.3). We summarise the findings by comparing the results of the interviews to the expected impact on communication problems (refer to section 5.3.4).

6.2.1 Implementation interviews

Interviews were conducted with four Scrum team members, which included three developers and the product owner, to determine whether the solution implemented addressed the communication challenges and what the impact of the solution was on the challenges identified at CGIS. Of these four stakeholders, three had been interviewed in previous rounds of data-gathering.

The interviews were conducted after the ARES had been implemented for two sprints, i.e., the solution had been implemented for two weeks. Data-gathering cycles were conducted on two different days in November 2021 with all interviewees being individually interviewed, using three questions:

1. Evaluating the selected solution, do you believe that the solution addresses the areas of concern regarding communication between project stakeholders on a software development project?
2. In terms of Whetherby's PIECES check list (refer to Fatoni et al. (2020)), please provide evidence (documents, images, or "stories") of the impact of the proposed solution on the Information Outputs.
3. Do you believe that the solution has certain deficiencies? What are these deficiencies and how should the solution be adapted to address the deficiencies?

The sessions, which ranged between 6 and 35 minutes, were recorded and the researcher was able to take notes and reference the recordings as necessary. The interview results are elaborated on in three sections. In section 6.2.2 we include feedback obtained during the interviews on how communication was impacted by the solution. In section 6.2.3 we discuss the perceived deficiencies of ARES. Lastly, we discuss the impact of the solution on the communication problems at CGIS in section 6.2.4.

6.2.2 Communication challenges addressed by the implemented solution

In this section we include feedback obtained from the CGIS participants regarding the impact of the solution on the communication problems. We address *RQ8,* i.e., *Based on the implementation and evaluation results, how well are the communication challenges addressed by the implemented solution?* in section 6.2.4 by comparing the expected impact of the solution on communication problems (refer to section 5.3.4) with the actual feedback obtained from the CGIS participants.

All four interviewees stated that the implemented solution improved communication and that it was an improvement on what was being done at CGIS. Feedback included "light years ahead of what we had" and "a lot more information was available than before".

The product owner said that he was initially sceptical about the planned duration of meetings/events, but that it was "worth it" and "priceless". One developer commented on the positive impact of the daily Scrum, which ensured that the Scrum team knew "what was going on" and understood who was busy with what. He further stated that "the information load adjusted as we focused on what was being done and what needed to be done", i.e., the right amount of relevant information was available when required.

Regarding the retrospective, the product owner valued the honest communication and feedback, and stated that the fact that cameras were on contributed to him feeling that we were "an actual team" and could be more comfortable with each other. The product owner therefore thought that the "face-to-face" communication made a difference whereas one developer commented that if the cameras being on made a difference, it was very subtle. The QA developer stated that the benefit of the retrospective was that an issue could be raised that the other stakeholders had not yet considered and that the team could discuss and address the issues together.

© University of Pretoria

A statement made by most of the Scrum team was that Scrum created opportunities for them to obtain information when they needed it and discuss information with the relevant stakeholders. One developer said that having a Scrum master and product owner on his "beck and call" was helpful as he could discuss an in-progress ticket and did not need to wait for feedback, in which case he could potentially forget about the question or make his own assumptions due to the unavailability of stakeholders. In a specific example, he said that he would have made assumptions and solved the problem differently, but he was able to raise his question in the daily Scrum, ensuring that we didn't only find out later about the misalignment.

Problems were also identified faster as the team felt they had a place that they could raise issues and address issues within 24 hours. This was the "greatest win" according to the product owner. The QA developer said the Scrum events highlighted what the team didn't know, and that the team could quickly realise that we didn't all understand one task and had the opportunity to discuss it. The senior developer interviewed commented on the solution ensuring that "information is not lost" and topics are not discussed too late in time, i.e., "information is in your face", visible, and everyone can contribute. He specifically commented on the benefits of frequent group communication and that there was lots of information as well as relevant information. He said that he doesn't always understand in the beginning when somebody makes a statement or asks a question, so he answers as best as he can. In a group setting, however, somebody else could clarify what the other person meant so that he doesn't need to rely on only his own interpretation of the message.

Regarding Mural as a tool, one developer stated that he enjoyed the Mural board and that everyone could contribute, whereas the product owner stated that he still needed to decide whether he thought it was the best option. The product owner mentioned that he would have preferred a physical Scrum board in the office.

6.2.3 Deficiencies of the implemented solution

In this section we address *RQ9: What were the deficiencies of the implemented solution?*

The RSD documentation, with regards to the quantity and quality, was a concern after the first two weeks of implementation. The product owner stated that we needed to improve documentation to ensure that the QA developer had enough information to test the system. In addition to this, the product owner was concerned that some stakeholders were expecting too much documentation, which could result in irrelevant information. Finding the balance was therefore difficult, as well as determining what information should be available in what format and when.

The QA developer stated that the documentation, i.e., the RSD, was sufficient and more comprehensive for the front-end requirements. Wireframes and acceptance criteria were sufficient and documented for the front-end requirements, whereas the APIs and back-end requirements were not sufficiently documented.

Testing was also still a topic of debate after the second week of the implementation. The team was unsure who should be responsible for which testing and when. The product owner mentioned that he believed that the QA developer still needed to be responsible for system testing, as he suspected that the .net developers would not consider all scenarios and exceptions. The QA developer stated that a lot had been discussed, but that very little had been sufficiently tested after two weeks, which would result in a lot of work for him at the end of the implementation. He suggested that we deploy the changes to the QA environment as well, instead of only the development environment, so that he could conduct more testing once the developers had demonstrated the solution, and that we could discuss and add additional requirements in the sprint planning sessions rather than only identifying testing concerns right at the end.

One Scrum team member said that we were taking too long to move forward and that he suspected that it was due to the solution being an adjustment. He believed that the solution and the time spent on planning and discussion would be worth it in the end but said that he was intrigued to see the impact on the quality of the end-product delivered.

A concern raised by one of the developers was the scalability of the solution. Even though he found the solution to be beneficial and would want CGIS to incorporate the ARES into their operations, he was unsure how the solution would practically be implemented considering that CGIS has many different projects and products, e.g., how would Scrum teams be created and on which projects/products could ARES be implemented and why.

6.2.4 Summary of the ARES impact on communication problems

In this section we address *RQ8*: *Based on the implementation and evaluation results, how well are the communication challenges addressed by the implemented solution?*

The Scrum events addressed all problems as expected, including inadequate comprehension (C1), incorrect information being provided (M2), important information being obtained/provided too late (D7a), different information being communicated to different stakeholders (D8), information being translated incorrectly if passed through many stakeholders (D10), communication silos/islands of communication (X2), and inadequate context

© University of Pretoria

(X7). These problems were predominantly addressed by the team knowing "what was going on" and understanding who was busy with what, and the fact that they could frequently discuss the project with relevant stakeholders in the Scrum events. The fact that the Scrum team were always involved also ensured that everyone was on the same page and could correct each other if something was being misunderstood. In addition to this, the Scrum events addressed absent client representatives (X1), all relevant stakeholders aren't involved in requirements elicitation (X3), and unavailable stakeholders (X5). The dedicated events at specific times on specific days created a routine, and stakeholders were committed to attending these events.

The same team members being involved in all Scrum events addressed absent client representatives (X1), communication silos/islands of communication (X2), unavailable stakeholders (X5), and inadequate context (X7). The dedicated team therefore addressed more problems than expected. The Scrum events created opportunities for the Scrum team to obtain information when they needed it and discuss information with the relevant stakeholders. One developer said that having the Scrum master and product owner on his "beck and call" was helpful as he could discuss an in-progress ticket and did not need to wait for feedback.

We expected the Scrum master's accountabilities to address the lack of communication structure/requirements (X6), information being spread across different channels (D3), different information being communicated to different stakeholders (D8), and differences in terminology/definitions (D6). The CGIS participants did not comment on these problems during the interviews, and we therefore can't assume that these problems were addressed. Structure was however created seeing as we adhered to the 2020 Scrum guide (Schwaber & Sutherland, 2020) as best as possible, with all team members present in all events, and only using three dedicated communication channels, namely Microsoft Teams, the Mural Board, and GIT.

We believed that the following communication problems could be addressed by the RSD approach, namely inadequate comprehension (C1), insufficient information being provided (M3), a lack of documentation/ requirements (M4a), outdated/inaccurate information in documentation (M4b), an insufficient amount of relevant information in the documentation (M4c), irrelevant information being included for a specific stakeholder (M4d), too much information being included in the specification (M4e), no standard regarding the content and volume in documentation (M4f), requirements not being able to be defined/translated (D4), and all relevant stakeholders not being involved in requirements elicitation (X3). It was too early in the process to know whether the RSD addressed outdated/inaccurate information in documentation (M4b). Regarding all relevant stakeholders not being involved in requirements elicitation (X3), we believe that this was addressed by the Scrum framework and not the RSD approach. The RSD approach did not address the lack of a standard regarding the content and volume in documentation (M4f). Regarding the other communication problems listed, the team commented on the RSD approach being sufficient for some requirements, but not for all, e.g., the front-end requirements were well documented whereas backend requirements were not as comprehensive. The RSD approach did therefore not sufficiently address elicitation and documentation challenges during RE.

In Fig. 26 we depict the actual impact of the ARES components on communication problems identified. For some problems, insufficient evidence could be gathered. For example, the Scrum master/researcher felt that the Scrum master role had an impact on certain communication problems, but the CGIS participants interviewed did not provide any feedback to support this. Some RSD addressed the problems, e.g., in the front-end specific requirements, but there were still limitations, and the problems were therefore only partially addressed by the RSD approach. Another example is that the researcher believed that the regular Scrum events, the dedicated team, and the RSD approach addressed the weak knowledge of the application domain, but insufficient evidence was provided by the CGIS participants to support this.

© University of Pretoria

Communication problem in RE	Scrum events	"Face-to-face" discussion	Dedicated team	Scrum master	The RSD approach
C1: Inadequate comprehension					
M1: Unnecessary requirements provided					
M2: Incorrect information provided					
M3: Insufficient information provided					
M4a: Lack of any documentation/requirements					
M4b: Outdated/inaccurate information in documentation					
M4c: Insufficient amount of relevant information in the documentation					
M4d: Irrelevant information included for a specific stakeholder					
M4e: Too much information in specification					
M4f: Lack of standard regarding content and volume in documentation					
D3: Information is spread across different channels					
D4: Requirements cannot be defined/translated					
D6: Differences in terminology/definitions					
D7a: Important information is obtained/provided too late					
D8: Different information is communicated to different stakeholders					
D10: Information is translated incorrectly if passed through many stakeholders					
D11: Misalignment between client/sales regarding requirements					
D12: Misalignment between sales/operations regarding capabilities					
D13 Misalignment between sales and operations regarding client requirements					
X1: Absent client representative					
X2: Communication silos/islands of communication					
X3: All relevant stakeholders aren't involved in requirements elicitation					
X4: Different client representatives with different requirements					
X5: Unavailable stakeholders					
X6: Lack of communication structure/requirements					
X7: Inadequate context					
X8: Weak knowledge of application domain					

Key:	
	Addressed by ARES
	Insuficient evidence/partially addressed
	No impact was expected/realised

Fig. 26. Actual impact of ARES on communication problems

We provide a comparison of the expected impact and the actual impact of the ARES on communication problems in Fig. 27. There were five instances in which the researcher expected a problem to be addressed by the ARES, but it was not. Insufficient evidence was obtained that the face-to-face communication had an impact on communication. One developer, when asked whether he found the face-to-face communication beneficial, stated that if the cameras being on made a difference, it was very subtle. He said that it may have been beneficial to conduct the first sprint *without* face-to-face communication and the second sprint *with* face-to-face communication for comparison. The product owner stated that the face-to-face communication made it feel like we were a team, but there was no evidence that it impacted the communication problems identified.

Regarding the RSD approach, it was too early in the process to know whether the RSD addressed outdated/inaccurate information in documentation (M4b). Whether the RSD approach addressed this problem would only be realised in the future. The relevant stakeholders were involved in requirements elicitation as a result of the Scrum events and not as a result of the RSD approach. The RSD approach only facilitated the elicitation and there was no evidence that the RSD approach addressed this problem. Lastly, statements were made that the RSD approach was sufficient for some requirements but not for all. The RSD approach did therefore not address the lack of a standard regarding the content and volume in documentation (M4f).

The limitations of the ARES implementation are elaborated on in the following section.

© University of Pretoria

	Scrum events		"Face-to-face" discussion		Dedicated team		Scrum master role		The RSD approach	
Communication problem in RE	Expectation	Results	Expectation	Results	Expectation	Results	Expectation	Results	Expectation	Results
C1: Inadequate comprehension										
M1: Unnecessary requirements provided										
M2: Incorrect information provided										
M3: Insufficient information provided										
M4a: Lack of any documentation/requirements										
M4b: Outdated/inaccurate information in documentation										
M4c: Insufficient amount of relevant information in the documentation										
M4d: Irrelevant information included for a specific stakeholder										
M4e: Too much information in specification										
M4f: Lack of standard regarding content and volume in documentation										
D3: Information is spread across different channels										
D4: Requirements cannot be defined/translated										
D6: Differences in terminology/definitions										
D7a: Important information is obtained/provided too late										
D8: Different information is communicated to different stakeholders										
D10: Information is translated incorrectly if passed through many stakeholders										
D11: Misalignment between client/sales regarding requirements										
D12: Misalignment between sales/operations regarding capabilities										
D13 Misalignment between sales and operations regarding client requirements										
X1: Absent client representative										
X2: Communication silos/islands of communication										
X3: All relevant stakeholders aren't involved in requirements elicitation										
X4: Different client representatives with different requirements										
X5: Unavailable stakeholders										
X6: Lack of communication structure/requirements										
X7: Inadequate context										
X8: Weak knowledge of application domain										

Key:	
	Expected to be addressed by ARES
	Addressed by ARES
	Insuficient evidence/partially addressed
	No impact was expected/realised

Fig. 27. Comparison of the expected and actual impact of ARES on communication problems

© University of Pretoria

6.3 Limitations of the implementation

Limitations of the implementation approach may pose threats to validity and are included in this section.

CGIS does not currently implement Scrum and most CGIS practitioners do not have Scrum experience. The Scrum master, who is also the researcher, was the only team member that had obtained formal Scrum training. In addition to this, this Scrum implementation was the first Scrum project implemented by the Scrum master. The lack of knowledge and experience could have had an impact on the efficiency of the implementation, as well as how the implementation was done. The RSD approach was also a new practice, and the lack of knowledge and experience could have also impacted the efficiency of the solution.

CGIS management wanted to first experiment and evaluate the process and benefits before potentially impacting paying clients. Therefore, only the internal client was included in the process. The product owner and client were the same person, and the CGIS sales team was not involved in the project. This reduced the number of stakeholders involved and resulted in only CGIS employees being included in the implementation. Therefore, the results could be different on future implementations, as more stakeholders could impact the findings.

Considering that this study had to be submitted within a certain timeframe, the researcher could not select from multiple projects a project that would be most suitable for experimentation. A different project could have been a more suitable option for a first time Scrum implementation within CGIS. In addition to this, the study only evaluated two sprints, i.e., two weeks. If the solution had been implemented for a longer period, and the Scrum team had become more familiar with the ARES before discussing the impact on communication problems with them, the results could have been different. The team was also not fully dedicated to the project and team members were often distracted by other issues and external factors. This could have impacted the efficiency of Scrum and therefore the outcome of the ARES.

During the implementation interviews (refer to section 6.2.1) the Scrum team was asked whether the solution addressed the areas of concern regarding communication between project stakeholders on a software development project. The Scrum team members were however not necessarily the same CGIS practitioners interviewed during each ADR stage. The Scrum team for the ARES was selected based on availability and did not necessarily have any background on the study, i.e., they did not necessarily contribute to the communication problems initially identified in section 4.4. Therefore, the perception of the communication problems of the practitioners interviewed in the implementation stage could have been different to the practitioners' perception of the problems initially identified.

Measuring and comparing the communication problems experienced by CGIS practitioners, as well as the perceived impact of the solution on these problems, is dependent on the specific participants' and researcher's opinions and it is therefore subjective. Considering that the project was still in progress at the time of evaluation, it was also challenging to evaluate the impact on *on-time delivery*, *quality of delivery*, and *delivery within budget*. The final stage of this study, i.e., the implementation of the solution, was discussed in this chapter. Considering that the ARES artefact has been implemented and evaluated, we can reflect on the study's findings. In the next chapter, we repeat the 9 research questions, initially presented in section 1.6, and include the insights obtained per research question.

© University of Pretoria

Chapter 7– Discussion

In this chapter, we repeat the research questions from section 1.6 and communicate the main insights that were extracted from our study.

RQ1: What are the problems experienced at CGIS that hampers/impacts the delivery of projects within the project management triple constraint? A few root causes were identified at CGIS, including unforeseen events, client complexities, inadequate management, inexperience, inadequate communication, unavailable resources, negative company culture, and employee attitudes (refer to Appendix A and Appendix B). Narrowing down the scope of analysis, it was decided that the focus of this study would be on addressing *inadequate communication*, since the primary researcher had observed similar communication challenges on a software development project while working at a different, larger enterprise within the telecommunications industry. In addition to this, addressing root causes such as unforeseen events and client complexities, for example, could be challenging as these root causes are factors external to CGIS. Inadequate communication causes employees to be uninformed, i.e., unaware of project details/considerations/factors etc. applicable to them, which causes project rework and/or incorrect estimates. Tasks therefore take longer than expected, which impacts project delivery and quality. The problems experienced at CGIS are the perceptions of the specific interviewees/practitioners selected, and are the problems as perceived at the time of the interview. It was therefore important to determine whether the problem selected, i.e., inadequate communication, exists within the broader software development sector.

RQ2: Do the communication challenges identified at CGIS exist within the broader software development sector? Inadequate communication exists as a class-of-problems within software development projects, i.e., this study will not only address a problem that exists at CGIS alone. After conducting the SLR (Chapter 4), however, it was determined that general terms are used for *inadequate communication,* e.g., communication gaps, and communication problems, and that these general terms could be referring to vastly different problems which would require different solutions. For example, the term "communication gaps" at CGIS could be caused by issues with infrastructure and/or technology whereas the "communication gaps" in a study identified in literature could be caused by infrequent communication, in which case the solution(s) to each problem would be different. Our research at CGIS indicated that communication problems are more prominent within the requirements elicitation phase of software development projects, which is supported by Fernández et al. (2017) and Bjarnason et al. (2011) in literature. Retrospectively, the researcher should have comprehensively understood the specific communication problem(s) at CGIS before conducting the SLR. Although inadequate communication exists as a class-of-problems, it would have been more beneficial to obtain insights into the specific communication problems at CGIS, i.e., requirements that cannot be defined and/or translated, and misalignment between stakeholder regarding requirements. The scope should have therefore been limited to RE in the SLR.

RQ3: What are the specific communication challenges that exist during the requirements elicitation/analysis phase at CGIS that should be addressed? By conducting interviews with CGIS practitioners, we were able to comprehensively define and classify the inadequate communication and understand the nature of the challenge. The taxonomy presented in section 4.4.3 includes the communication problems at CGIS, and highlights problems identified in RE specifically. In the taxonomy, we base the grouping of the problems on the categories of human communication as presented by Littlejohn and Foss (2005) which includes the *elements of the communication model* (that consists of communicator, message, and conversation) and the *context of communication* (that consists of relationship, group, organisation, health, culture, and society). For the initial taxonomy, we did not apply the sub-categories for the main category *context of communication*. We reason that every enterprise differs in context and the contextual categories will also differ. Communication problems in the requirement elicitation/analysis phase (RE) were mentioned by fourteen of the fifteen interviewees. The only interviewee that did not mention problems in RE was the call centre team lead, and he is not typically involved in RE. Most problems were related to requirements that cannot be defined and/or translated (8 interviewees), and misalignment between stakeholders. Of the 33 themes of communication problems identified, 27 could be classified as requirement elicitation/analysis problems. Most problems were related to *message* and *context of communication* problems. Each of these categories (refer to Fig. 28) had nine RE issues each.

© University of Pretoria

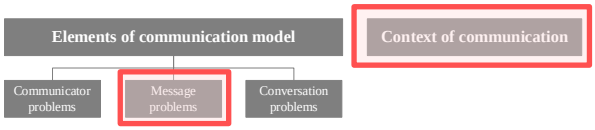


Fig. 28. The rudimentary taxonomy's structure with emphasise on problem areas

RQ4: What are existing Agile methodologies, frameworks, and Agile RE practices? Many Agile frameworks and RE practices exist. Examples of Agile frameworks include Scrum (Schwaber & Sutherland, 2020), Extreme Programming (XP) (Wells, 1999), and Kanban (Atlassian Agile Coach, 2021b). In addition to this, Agile *scaling* frameworks address problems associated with implementing Agile in a context it was not initially intended for, and examples include Large-Scale Scrum (LeSS) (Larman & Vodde, 2017), Scaled Agile Framework (SAFe) (© Scaled Agile, 2021), and Scrum of Scrums (SoS) (Atlassian Agile Coach, 2021a). In the 15th state of Agile report (Digital.ai, 2021), Scrum was identified as the most popular Agile approach, with 66% of survey participants stating that it was the framework they follow most closely. According to Measey (2015), common Agile practices that are generic across most or all frameworks include short feedback loops, face-to-face communication, daily stand-ups, show and tells, emergent documentation, visual boards, sustainable pace, and a focus on quality. Two other examples of Agile practices include swarming and planning poker. *Swarming*, which occurs when as many team members as possible work simultaneously on the same priority item until it is done (Sutherland, 2020), ensures that the highest priority is completed first. This practice is not prescribed by Scrum, but it is believed that this practice, which ensures that team members work collaboratively, could increase knowledge transfer between team members and therefor address communication problems. *Planning poker*, an agile estimating and planning technique that is consensus based (Mountain Goat Software, n.d.), could be used to ensure that the Scrum team members are aligned with regards to requirements, estimates, and expected obstacles per PBI.

RQ5: Which Agile RE solution would be the most suitable for CGIS to address the communication challenges identified? A solution was designed and implemented to address the communication challenges experienced at CGIS. This Agile RE solution, ARES, primarily consisted of the Scrum framework and the RSD approach. Scrum was selected as it is the most used Agile framework according to the 15th state of Agile report (Digital.ai, 2021), and according to Measey (2015) it is simple to pick up. Scrum is therefore ideal for CGIS that does not currently follow an Agile framework. The RSD approach replaced the typically used user stories, which is an Agile practice, as research conducted by Medeiros et al. (2020) indicated that the RSD approach integrates functional and technical requirements in a single view, therefore providing sufficient information required for coding, addressing user story shortcomings. The primary researcher had the intention of implementing multiple Agile practices from the beginning of the project to address communication problems. Practices that *were* implemented included product planning, product roadmap, the RSD approach, and face-to-face virtual meetings in which attendees had their cameras on and could see their team members' faces. Practices that the researcher would have wanted to implement included swarming, planning poker, and automated testing. The implementation of the Agile solution was however challenging, considering that most CGIS practitioners were unfamiliar with Scrum and the RSD approach, and incorporating too many changes and new practices would induce too much change at once. According to Measey (2015), it is easy to add practices once the basic practices of Scrum have been mastered. It is therefore suggested that these three additional Agile practices, i.e., swarming, planning poker, and automated testing, are introduced at a later point in time when the Scrum team has adjusted to the new Scrum practices, as the researcher believes that these additional practices could also impact communication problems.

RQ6: How (and why) should the selected Agile RE solution be adjusted/adapted for CGIS? The CGIS practitioners were involved in designing and refining the solution, and therefore assisted in determining the adjustments. An adjustment made to the RSD approach was that an analyst was not involved in defining the RSD. The Scrum team created and managed the RSD as it was believed that the developers had to creatively solve the problem instead of being directed by an analyst. Scrum recommends that every member of the Scrum team should be fully dedicated to the project. We, as a Scrum team, did our best to ensure that the Scrum team was dedicated to the project, but this proved difficult. CGIS as a company, has not adopted Scrum as a standard methodology within the enterprise, e.g., there are no dedicated teams, product owners are often responsible for many products and the Scrum master role has not been implemented as a separate role. Even though CGIS management approved the implementation of Scrum on a specific project, other current projects and responsibilities could not be neglected and all team members were at some point involved on other projects during the two-week period included in this study. Agile practices not prescribed in Scrum, such as a product roadmap, product planning, and face-to-face communication were incorporated into our solution to increase the efficiency of Scrum and facilitate communication as much as possible. Adjustments made due to social-distancing constraints posed by the COVID-19 pandemic, included an electronic Scrum instead of a physical Scrum board, as well as cameras being turned on during events to facilitate "face-to-face" communication.

RQ7: *How were the Agile RE components implemented?* We were able to demonstrate a real-world implementation of the adapted Agile RE solution over a two-week period. It was difficult to select the project for experimentation as different stakeholders had different opinions on which project should have been selected. The

© University of Pretoria

project that was selected was mostly selected due to timing constraints, i.e., the project had to start during the last quarter of 2021 and the Scrum master, who is also the researcher, had to be assigned to the project. We implemented the Scrum framework as close as possible to the guidelines provided in the 2020 Scrum guide (Schwaber & Sutherland, 2020) and had sprint planning sessions, daily Scrums, sprint reviews, and sprint retrospectives every week. The Scrum team attended all Scrum-scheduled meetings except for the QA developer who may have missed a daily Scrum or two. Cameras were also turned on to facilitate "face-to-face" communication. The RSD approach was used to document requirements on the Mural board. Acceptance criteria were discussed between the Scrum team and then added to the RSD with "sticky notes". We decided that we would perform one-week sprints and adjust the duration if required. During the implementation documented in this study, the need was not identified to change the sprint duration.

RQ8: Based on the implementation and evaluation results, how well are the communication challenges addressed by the implemented solution? Most communication problems identified were related to requirements that cannot be defined and/or translated, and misalignment between stakeholders (the Sales and Operations team at CGIS, and the client and the CGIS sales team). We were able to determine that communication problems were addressed/impacted by the solution developed and implemented at CGIS, i.e., by the ARES, but that the problems that occurred *most frequently* were not necessarily addressed. Scrum events and the dedicated team addressed the expected problems as well as additional problems. The role of the Scrum master and the RSD approach addressed some problems, but limitations and/or insufficient evidence were apparent. In addition to this, there was no evidence to support that face-to-face communication impacted communication problems. Problems that *were not* addressed by the ARES included unnecessary requirements provided (M1), outdated/inaccurate information in documentation (M4b), lack of a standard regarding content and volume in documentation (M4f), misalignment between stakeholders (D11, D12, and D13), and different client representatives with different requirements (X4). It was not expected that the ARES would impact the weak knowledge of the application domain, but the regular Scrum events, dedicated team, and the RSD approach impacted this problem according to the researcher. The CGIS participants did not however provide evidence to support this. We were able to determine that the ARES increased effective communication between stakeholders during RE at CGIS, but it was difficult to determine the impact on project delivery.

RQ9: What were the deficiencies of the implemented solution? Three primary concerns were identified, namely documentation, testing, and the scalability of the solution. Documentation was the only communication-related deficiency identified. It was difficult to determine what information should be available in what format and when. Identifying and documenting requirements was not a concern, but adding acceptance criteria before the developers had decided how they were going to address the requirement was difficult. Considering that the ARES had only been implemented for two weeks, it is difficult to determine whether this was a solution deficiency or whether the Scrum team needed more time to adjust to the new way of working. By the end of the second week, the team had agreed that the RSD should be created as best as possible in the sprint planning event, and that the developers had to add information during the week, and before the sprint review, based on the decisions made during the week. The testing of the system was also a concern. With Scrum, the objective is to prepare shippable code at the end of the sprint and not necessarily working code in production.

According to Skelton and Betteley (2017), Scrum was designed in a time when scalability, deployability, monitoring, and maintenance was not a concern. As stated by the International DevOps Certification Academy™ (2020), DevOps prioritises test automation and continuous software integration whereas software delivery is an afterthought in the Scrum framework. In addition to this, the entire DevOps team is concerned about the quality of the software, whereas software quality assurance is not detailed in the Scrum framework. Samarawickrama and Perera (2017) built a methodology that incorporates the managerial aspect of the software development lifecycle from Scrum and the rapid delivery requirement from DevOps. It could be beneficial to consider investigating and incorporating DevOps into the ARES. The scalability of the ARES was also identified as a potential concern by one of the Scrum team members. According to Paasivaara, Behm, Lassenius, and Hallikainen (2018), recent scaling frameworks are largely unvalidated and there seems to be a need for organisations to tailor an agile approach to fit its own organisational, business, and product context. It should be determined how the ARES can be adjusted and scaled at CGIS so that other projects and products can also benefit from the improved information flow/communication between stakeholders during RE. This study's 9 primary research questions, initially stated in section 1.6, have been answered in this chapter. Valuable insights were extracted, and we conclude our findings, discuss the main contributions of the study, as well as recommendations for future research in the following chapter.

© University of Pretoria

Chapter 8– Conclusion and future research

CGIS, a medium-sized enterprise in South Africa that executes software development projects, experiences inadequate communication and its negative impact on stakeholder alignment and project delivery. In section 8.1 we provide a summary of the study, and the contributions of the study are included in section 8.2. How the findings can be used in future research endeavours is included in section 0 and the study is concluded in section 8.4.

8.1 Summary of the study

This study followed an Action Design Research (ADR) methodology to address a problem identified at a medium-sized enterprise in South Africa in the geographical information system (GIS) industry referred to as CGIS. The ADR methodology was used to address the research questions in a systematic way.

Inadequate communication between software development project stakeholders is a problem at CGIS which causes project tasks to take longer than expected, negatively impacting on-time delivery, quality of delivery, and delivery within budget. By conducting an SLR, it was determined that communication problems exist as a class-of-problems in software development projects. In addition to this, it was determined that the diagnosis, i.e., "inadequate communication", was too vague and that the researcher had to obtain a better understanding of the problem to sufficiently address the concern. The first contribution, a rudimentary *taxonomy of perceived communication problems* at CGIS, was therefore created which identified the requirement elicitation/analysis phase as the primary concern. To address the communication problems in this phase, it was decided that the second contribution, an Agile RE solution (ARES), would be designed, collaboratively between the researcher and CGIS practitioners, and implemented on a specific CGIS project. This ARES included the Scrum framework with the RSD approach as a replacement to the traditional user story. The ADR implementation stage followed, during which the ARES was implemented and evaluated. The ARES improved the information flow/communication between stakeholders during RE and therefore reduced misalignment between project stakeholders. Considering that the solution was only implemented for two weeks, we could not determine the impact on project delivery in this study. Potential concerns included the process of documenting acceptance criteria, testing, and the scalability of the solution.

The study's contributions, including a rudimentary *taxonomy* of perceived communication problems and an *Agile RE solution* (ARES), are discussed in the following section.

8.2 Contributions

As discussed in section 1.9.1, there is a need for more practical demonstrations of Agile RE. In this study, we demonstrated a practical implementation of an Agile RE solution, applied to a real-world project, therefore providing empirical evaluation of agile practices in an industry case. More studies are also needed in which Agile software development (ASD) is applied in Africa (refer to section 1.9.5). The ARES was implemented and evaluated at a medium sized enterprise in South Africa. By discussing the solution and implementation with South African CGIS practitioners, we were able to provide insights and empirical evaluation into the benefits and challenges of applying ASD in Africa. In a study by Sebega and Mnkandla (2017) it was identified that elicitation, documentation, and non-functional requirements integration need special attention in Agile RE in the South African software industry. The RSD approach, incorporated into our solution (refer to section 5.2.1), aims to address elicitation and documentation challenges during RE and specifically focuses on the inclusion of non-functional requirements. By evaluating the RSD approach in our practical demonstration, we were able to provide insights into how this approach could address the concerns raised by Sebega and Mnkandla (2017).

Within the stated context, the study produced two contributions, namely a *taxonomy of perceived communication problems*, and an *Agile RE solution (ARES)*. Although the taxonomy is an artefact that was developed as a theory-ingrained artefact, we have not evaluated the taxonomy and therefor present it as a secondary contribution that needs further refinement in the future. The *ARES* is the main contribution that demonstrates the three stages of an ADR project, namely diagnosis, design and implementation. The next two sub-sections present the two contributions.

© University of Pretoria

8.2.1 Taxonomy of perceived communication problems

Insight into effective communication in software development teams was identified as a need in section 1.9.2. An SLR to determine whether inadequate communication exists as a class-of-problems within software development projects indicated that general terms are used for *inadequate communication*, e.g., communication gaps, and communication problems, and that these general terms could be referring to vastly different problems which would require different solutions. The SLR therefore directed our analysis towards a search for existing taxonomies about communication problems in software development projects/environments. The researcher expected to find an article that synthesizes, in a taxonomy, typical causes for inadequate communication in software development projects, e.g., a categorisation of communication-related problems linked to standard phases of the software development cycle as well as solution pathways for the prevalent causes. This could however not be found in the search conducted, highlighting the need to create a vocabulary that could be used to describe communication problems in software development.

By presenting a rudimentary taxonomy of communication problems in software development in section 4.4 as a *secondary contribution*, which is based on the categories of human communication as presented by Littlejohn and Foss (2005), we were able to provide insights into the factors that impact effective communication in software development teams. The *taxonomy* assisted the researcher in comprehensively defining and understanding the specific communication problems experienced at CGIS.

In terms of generalisability, we believe that the rudimentary taxonomy could assist researchers and practitioners at other enterprises in the future by enabling a shared vocabulary, making it easier to describe communication problems in software development. We expect that the rudimentary taxonomy would be adjusted if used in different enterprises and contexts, as the participants would be different and their perception of problems and how to categorise the problems would there also be different. The taxonomy would therefore need to evolve as it is used in different enterprises and contexts.

The taxonomy of perceived communication problems, already presented in Fig. 15, is repeated in Fig. 29.

© University of Pretoria

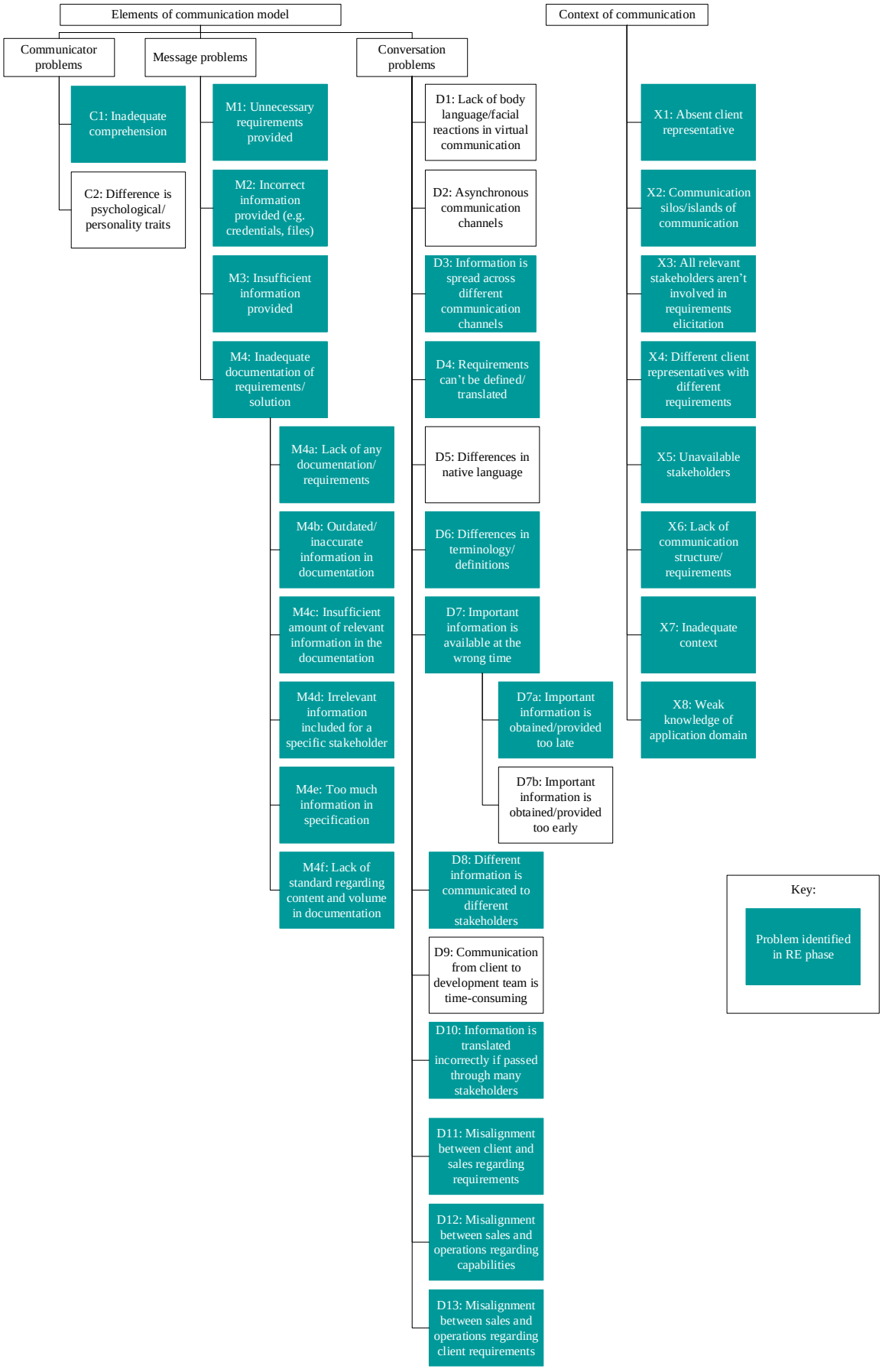


Fig. 29. Rudimentary taxonomy of perceived communication problems at CGIS

© University of Pretoria

8.2.2 Agile RE solution (ARES)

In section 1.9.3, it was determined that insight into stakeholder communication during RE specifically is required. The taxonomy presented in section 4.4 and again in section 8.2.1 highlights the communication problems during RE, providing insight into the problems that would need to be addressed to improve stakeholder communication during RE. Using the taxonomy to structure and focus attention on a particular category of problems, an *Agile RE solution* (ARES) was developed as the *main contribution*. The ARES focused on RE, providing empirical evidence of how communication during RE was addressed by a selected set of Agile frameworks and practices. The ARES, already presented in Fig. 22, is repeated in Fig. 30.

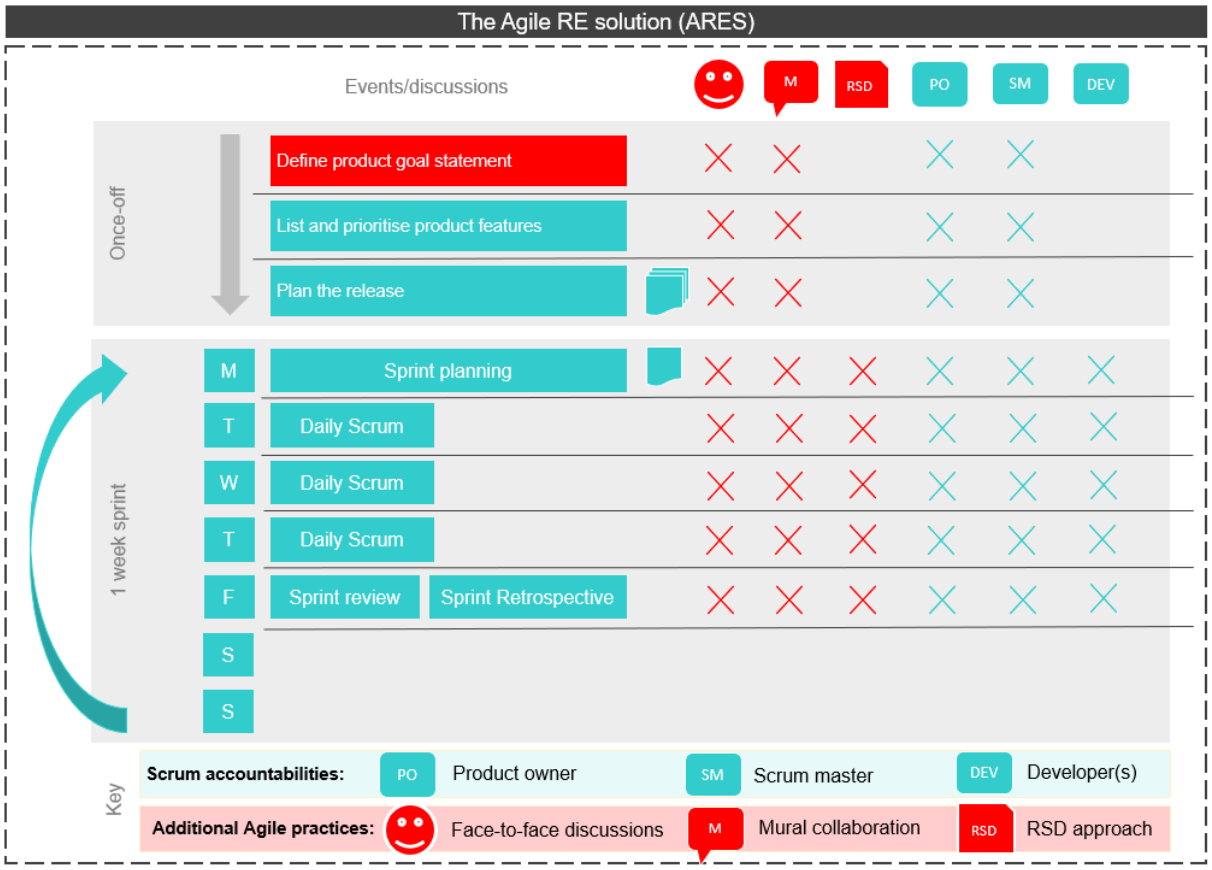


Fig. 30. Graphical representation of Agile RE solution (ARES)

A need was identified to reduce misalignment and the negative impact on project delivery (refer to section 1.9.4). The ARES was evaluated for two weeks, and the project was therefore still in progress when feedback was obtained from relevant stakeholders. We were able to determine that the ARES increased effective communication between stakeholders during RE at CGIS, but it was difficult to determine the impact on misalignment between stakeholders and the impact on project delivery.

The ARES improved stakeholder communication and addressed problems included in section 4.4, providing empirical evidence of how the ARES could improve communication between stakeholders on a software development project. A comparison of the expected and actual impact that the ARES had on communication problems, already included in Fig. 27, is included again in Fig. 31 . The Scrum events addressed the most problems, and also addressed more problems than initially expected. The face-to-face communication did not indicate any impact on the communication problems, and the role of the Scrum master and the RSD approach provided insufficient evidence, or only partially addressed the problems.

© University of Pretoria

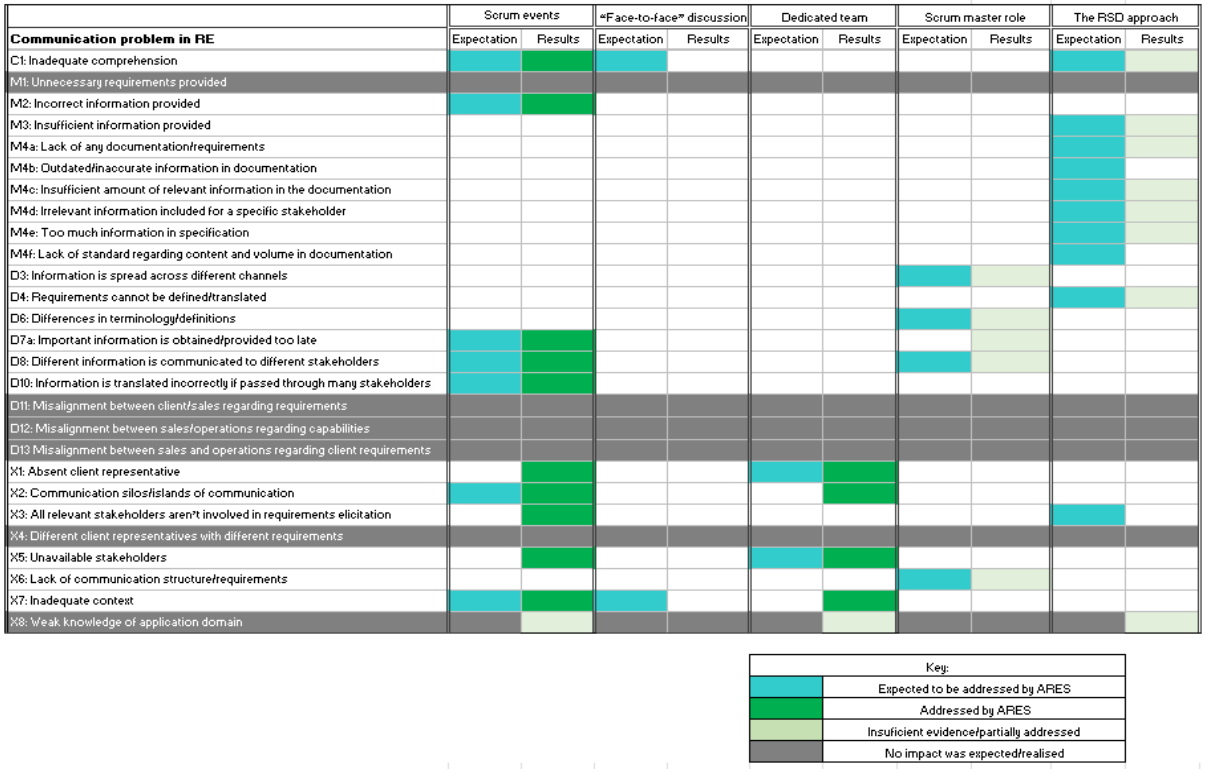


Fig. 31. Comparison of the ARES expectations and results on communication problems

By evaluating the ARES, through discussions with practitioners at CGIS, we confirmed that the artefact is useful and increased effective communication between stakeholders during RE at CGIS, i.e., the solution positively impacted problems identified in section 4.4. By demonstrating the solution within the CGIS context and confirming the positive impact the solution had on communication, this study could also motivate CGIS management to incorporate ARES into operations going forward, i.e., catalysing the migration to a formalised Agile framework.

Lawrence (2015) states that most qualitative research studies a specific issue in a certain population of a particular context, hence generalisability of qualitative research findings is not necessarily expected. In terms of the generalisability of the ARES, we believe that the ARES could be implemented/replicated in other enterprises and in other contexts, e.g., in an African country other than South Africa, as we have sufficiently documented the solution, the decisions made, the interview questions raised and the demographics of the interviewees, as well as the context of the enterprise in which these results were obtained. When implementing/incorporating the ARES at other enterprises/in other contexts, we expect the *results* to be different as the participants would be different. For example, cultural factors in other African countries could impact the feedback provided by participants to a researcher, and some participants may openly communicate and provide feedback whereas other participants might prefer to remain distant and not state their opinion as freely. This study's participants, i.e., test group, was also small. We only included three (3) CGIS practitioners in the design *stage* and four (4) CGIS practitioners in the *implementation* stage. These participants were also selected based on their involvement, and were therefore not randomly selected, impacting the expected generalisability.

Both the rudimentary *taxonomy* of perceived communication problems and the *Agile RE solution* (ARES) presented in this section could be improved and extended to obtain additional and/or different insights. Recommendations for future research are included in the following section.

8.3 Recommendations for future research

The main insight from the problem investigation at CGIS (Chapter 3) and the SLR conducted (Chapter 4), is that a standard taxonomy or classification structure is needed to highlight different kinds of communication problems within a software development context. A standard taxonomy should be extendable and based on enterprise-specific contextual variables. The taxonomy should also enable solution mapping, i.e., indicating which existing solutions, such as practices, techniques and/or tools, would be appropriate in addressing specific kinds of communication problems.

In this study, we already suggest a theory-ingrained taxonomy, applying the taxonomy at a real-world enterprise. For future work, the taxonomy should be validated at other enterprises that also experience

© University of Pretoria

communication problems during software development projects. One of the main categories of the taxonomy, the *context of communication*, should be further developed, based on other prominent clusters of communication-related problems. As an example, our research at CGIS indicated that communication problems are more prominent within the requirements elicitation phase of software development projects. We also extracted evidence from literature to support this claim (see Fernández et al. (2017) and Bjarnason et al. (2011)). Therefore, we suggest that a sub-category, called *requirements elicitation problems* should be added to the taxonomy's main category *context of communication*. The communication problems identified in the requirement elicitation/analysis phase at CGIS could also be compared to the findings in Fernández et al. (2017), validating the causes of communication flaws and the impact on the customer, product, design/implementation, and project (provided by Fernández et al. (2017)) within the South African context.

According to Mullarkey et al. (2019), an important direction for future research for their elaborated action design research (ADR) process is the application of the elaborated model to specific projects as well as incorporating the learnings into their model to build on their processes and methods. In this study, we gradually assembled multiple artefacts by using their model on a real-world project, but we did not provide insights in this study regarding potential elaborations and adaptions of their model based on our experience. Therefore, we recommend that the researcher documents the experience of using the elaborated ADR model on a real-world project to further grow the body of knowledge on ADR.

Only two weeks of the ARES implementation could be documented and included in this study due to time constraints. The practices that could address communication problems but were excluded from the study, i.e., swarming, planning poker, and automated testing, could be incorporated in the ARES in the future to determine the impact on the communication problems at CGIS.

After the study was concluded, the real-world project at CGIS continued to use the ARES. It is recommended that the Scrum team and relevant stakeholders are interviewed again once the project has been concluded, as the results obtained after a few more weeks of implementation could be different than the results included in this study, and additional and/or alternative insights could be obtained. In addition to this, it is recommended that the ARES is implemented on other project(s) at CGIS, e.g., a client project instead of a primarily internal project, so that the Agile RE solution can be compared in different contexts, e.g. the product owner and the client being different people.

Based on the feedback obtained on the deficiencies of the implemented solution (refer to section 6.2.3), the scalability of the Agile RE solution at CGIS should be investigated. Considering that CGIS has many different projects and products, it could be beneficial to determine how the Agile RE solution could practically be applied to the diverse environment at CGIS. Investigating whether DevOps could compliment the Scrum solution, especially with regards to software deployment and delivery, should also be investigated.

8.4 Concluding statement

As a closing remark for this study the theses statement is repeated:

An Agile RE solution can be designed to address a subset of classified communication challenges in software development companies (such as CGIS) to improve the information flow/communication between stakeholders during RE, that will assist management in reducing the misalignment between project stakeholders and/or the negative impact on project delivery.

This ADR study produced evidence that the ARES addressed a subset of classified communication challenges and improved communication between stakeholders during RE. This could assist management in reducing the misalignment between project stakeholders and/or the negative impact on project delivery. The results of this study therefore support the thesis statement.

© University of Pretoria

References

© Scaled Agile, I. (2021). Welcome to Scaled Agile Framework® 5! Retrieved from https://www.scaledagileframework.com/about/

Abbas, N., Gravell, A. M., & Wills, G. B. (2008). *Historical roots of agile methods: Where did "agile thinking" come from?* Paper presented at the International Conference on Agile Processes and Extreme Programming in Software Engineering.

Abelein, U., & Paech, B. (2011). State of practice of user-developer communication in large-scale IT projects. In *Requirements Engineering: Foundation for Software Quality* (pp. 95-111): Springer International Publishing.

Addison, T. (2003). E-commerce project development risks: Evidence from a Delphi survey. *International Journal of Information Management, 23*(1), 25-40. doi:10.1016/S0268-4012(02)00066-X

Agile Alliance. (n.d.). Agile 101. Retrieved from https://www.agilealliance.org/agile101/

Aldave, A., Vara, J. M., Granada, D., & Marcos, E. (2019). Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *The Journal of Systems & Software, 157*. doi:10.1016/j.jss.2019.110396

Alhazmi, A., & Huang, S. (2020). Survey on differences of requirements engineering for Traditional and Agile development processes. In *2020 SoutheastCon* (pp. 1-9): IEEE.

Alnuem, M. A., Ahmad, A., & Khan, H. (2012). Requirements understanding: A challenge in global software development, Industrial surveys in Kingdom of Saudi Arabia. In *2012 IEEE 36th Annual Computer Software and Applications Conference* (pp. 297-306): IEEE.

Alsaqaf, W., Daneva, M., & Wieringa, R. (2019). Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and Software Technology, 110*, 39-55.

Alzoubi, Y. I., Gill, A. Q., & Al-Ani, A. (2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information & Management, 53*(1), 22-37. doi:10.1016/j.im.2015.08.003

Atlassian Agile Coach. (2021a). Scrum of Scrums. Retrieved from https://www.atlassian.com/agile/scrum/scrum-of-scrums

Atlassian Agile Coach. (2021b). What is kanban? Retrieved from https://www.atlassian.com/agile/kanban

Aundhe, M. D., & Mathew, S. K. (2009). Risks in offshore IT outsourcing: A service provider perspective. *European Management Journal, 27*(6), 418-428. doi:10.1016/j.emj.2009.01.004

Baham, C., & Hirschheim, R. (2021). Issues, challenges, and a proposed theoretical core of agile software development research. *Information Systems Journal*. doi:10.1111/isj.12336

Bastow, J. (2015). The Product Vision Template. In: ProdPad.

Batra, D., Xia, W., & Zhang, M. (2017). Collaboration in agile software development: Concept and dimensions. *Communications of the Association for Information Systems, 41*(1), 429-449. doi:10.17705/1cais.04120

Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. (2001). The Agile Manifesto. Retrieved from https://www.agilealliance.org/agile101/the-agile-manifesto/

Bentley, L. D., Whitten, J. L., & Randolph, G. (2007). *Systems analysis and design for the global enterprise* (7th ed.). Boston: McGraw-Hill Irwin.

Bernard, H. R., Wutich, A., & Ryan, G. W. (2017). *Analyzing qualitative data: Systematic approaches* (2nd ed.). Thousand Oaks: SAGE.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011). Requirements are slipping through the gaps - A case study on causes and effects of communication gaps in large-scale software development. In *2011 IEEE 19th International Requirements Engineering Conference* (pp. 37-46).

Bowen, G. A. (2009). Document analysis as a qualitative research method. *Qualitative Research Journal, 9*(2), 27-40.

Buchan, J. (2014). An empirical cognitive model of the development of shared understanding of requirements. In *Requirements Engineering* (pp. 165-179): Springer Berlin Heidelberg.

Cerpa, N., & Verner, J. M. (2009). Why did your project fail? *Communications of the ACM, 52*(12).

Cervone, F. H. (2014). Effective communication for project success. *OCLC Systems and Services: International digital library perspectives, 30*(2), 74-77. doi:10.1108/OCLC-02-2014-0014

© University of Pretoria

Chatzoglou, P. D., Theriou, N. G., Dimitriadis, E., & Aggelides, V. (2007). Software project management and planning: The case of the Greek IT sector. *International Journal of Applied Systemic Studies, 1*(3), 305-316. doi:10.1504/IJASS.2007.017713

Cheng, B. H. C., & Atlee, J. M. (2007). Research directions in requirements engineering. In *Future of Software Engineering (FOSE '07)* (pp. 285-303): IEEE.

Collatto, D. C., Dresch, A., Lacerda, D. P., & Bentz, I. G. (2018). Is Action Design Research indeed necessary? Analysis and synergies between Action Research and Design Science Research. *Systemic Practice & Action Research, 31*(3).

Coughlan, J., Lycett, M., & Macredie, R. D. (2003). Communication issues in requirements elicitation: A content analysis of stakeholder experiences. *Information and Software Technology, 45*(8), 525-537. doi:10.1016/S0950-5849(03)00032-6

Crombie, I. K. (1996). *The pocket guide to critical appraisal : A handbook for health care professionals*. London: BMJ Pub.

Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *The Journal of Systems & Software, 139*, 32-50. doi:10.1016/j.jss.2018.01.036

Cypress.io. (2021). Why Cypress? Retrieved from https://docs.cypress.io/guides/overview/why-cypress#In-a-nutshell

Damian, D. E., & Zowghi, D. (2002). The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In *Proceedings IEEE Joint International Conference on Requirements Engineering* (pp. 319-328): IEEE.

Damian, D. E., & Zowghi, D. (2003). RE challenges in multi-site software development organisations. *Requirements Engineering, 8*(3), 149-160. doi:10.1007/s00766-003-0173-1

Davis, A., Dieste, O., Hickey, A., Juristo, N., & Moreno, A. M. (2006). Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In *14th IEEE International Requirements Engineering Conference (RE'06)* (pp. 179-188): IEEE.

DeFranco, J. F., & Laplante, P. A. (2017). Review and analysis of software development team communication research. *IEEE Transactions on Professional Communication, 60*(2). doi:10.1109/TPC.2017.2656626

Digital.ai. (2021). 15th State of Agile Report Retrieved from https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *The Journal of Systems & Software, 119*, 87-108. doi:10.1016/j.jss.2016.06.013

Doggett, A. M. (2005). Root cause analysis: A framework for tool selection. *Quality Management Journal, 12*(4), 34-45. doi:10.1080/10686967.2005.11919269

Fatoni, A., Adi, K., & Widodo, A. P. (2020). PIECES Framework and importance performance analysis method to evaluate the implementation of information systems. *E3S Web of Conferences, 202*, 15007. doi:10.1051/e3sconf/202020215007

Fernández, D. M. n., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., . . . Wieringa, R. (2017). Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical Software Engineering : An International Journal, 22*(5), 2298-2338. doi:10.1007/s10664-016-9451-7

Ferreira, C., & Cohen, J. (2008). Agile systems development and stakeholder satisfaction: A South African empirical study. *ACM International Conference Proceeding Series, 338*, 48-55. doi:10.1145/1456659.1456666

Flewelling, P. (2018). *The Agile developer's handbook: Get more value from your software development: get the best out of the Agile methodology*. Birmingham: Packt Publishing.

Flick, U. (2018). *Designing qualitative research* (Second edition. ed.). Los Angeles: Sage.

Ghobadi, S., & Mathiassen, L. (2017). Risks to effective knowledge sharing in Agile software teams: A model for assessing and mitigating risks. *Information Systems Journal, 27*(6), 699-731. doi:10.1111/isj.12117

GitLab B.V. (2021). The DevOps Platform has arrived. Retrieved from https://about.gitlab.com/

Gode, A. (2012). What is communication? - a naive approach. *International Journal of Translation, 10*(4), 145-148. Retrieved from doi:10.1075/babel.10.4.02god

Goldratt, E. M. (1990). *What is this thing called theory of constraints and how should it be implemented?* Croton-on-Hudson, N.Y.: North River Press.

Grech, T. (2015). The intersection of agile and waterfall. *Industrial Engineer: IE, 47*(8).

Guest, G., MacQueen, K. M., & Namey, E. E. (2012a). *Applied thematic analysis*. Los Angeles: Sage Publications.

Guest, G., MacQueen, K. M., & Namey, E. E. (2012b). Validity and Reliability (Credibility and Dependability) in Qualitative Research and Data Analysis. In *Applied Thematic Analysis* (pp. 79-106): SAGE Publications, Inc. .

© University of Pretoria

Hanisch, J., & Corbitt, B. (2007). Impediments to requirements engineering during global software development. *European Journal of Information Systems, 16*(6), 793-805. doi:10.1057/palgrave.ejis.3000723

Herbsleb, J. D., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering, 29*(6).

Herbsleb, J. D., Paulish, D. J., & Bass, M. (2005). Global software development at Siemens experience from nine projects. In *Proceedings of the 27th international conference on Software engineering* (pp. 524-533).

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hofstee, E. (2006). *Constructing a good dissertation: A practical guide to finishing a Master's, MBA or PhD on schedule*. Sandton, South Africa: EPE.

Holmstrom, H., Conchuir, E. O., Agerfalk, P. J., & Fitzgerald, B. (2006). Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)* (pp. 3-11): IEEE.

Iden, J., Tessem, B., & Päivärinta, T. (2011). Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts. *Information and Software Technology, 53*(4), 394-406. doi:10.1016/j.infsof.2010.12.002

IEEE Xplore. (2020). About IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-ieee-xplore

IGI Global. (2021). What is communication. Retrieved from https://www.igi-global.com/dictionary/communication/4619

Inayat, I., Salim, S. S., & Marczak, S. (2017). Survey of communication and awareness as the most relevant socio-technical aspects of requirements-driven collaboration among software development teams. *IET Software, 11*(6), 277-285. doi:10.1049/iet-sen.2016.0174

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior: Part B, 51*(Part B), 915-929. doi:10.1016/j.chb.2014.10.046

International DevOps Certification Academy™. (2020). What are the top 6 differences between DevOps and Scrum? (DevOps vs Scrum comparison). Retrieved from https://www.devops-certification.org/blog/What-Are-TOP-6-Differences-Between-DevOps-and-Scrum-DevOps-vs-Scrum-Comparison

Islam, S., Joarder, M. M. A., & Houmb, S. H. (2009). Goal and risk factors in offshore outsourced software development from vendor's viewpoint. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 347-352).

Jarzębowicz, A., & Poniatowska, K. (2020). Exploring impact of requirements engineering on other IT project areas – case study. *Computer Science, 21*(3). doi:10.7494/csci.2020.21.3.3618

Javed, I., Rodina, B. A., Muzafar, K., Fazal, E. A., Sultan, A., Mohd Hairul Nizam, N., . . . Muhammad, S. (2020). Requirements engineering issues causing software development outsourcing failure. *PLoS ONE, 15*(4). Retrieved from doi:10.1371/journal.pone.0229785

Jiménez, M., Piattini, M., & Vizcaíno, A. (2009). Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering, 2009*, 1-14. doi:10.1155/2009/710971

Junior, I. H. d. F., de Azevedo, R. R., de Moura, H. P., & da Silva, D. S. M. (2012). Elicitation of communication inherent risks in distributed software development. In *2012 Seventh IEEE International Conference on Global Software Engineering Workshop (ICGSEW 2012)* (pp. 37-42).

Karlsson, L., Dahlstedt, A. G., Regnell, B., Natt och Dag, J., & Persson, A. (2007). Requirements engineering challenges in market-driven software development - An interview study with practitioners. *Information and Software Technology, 49*(6), 588-604. doi:10.1016/j.infsof.2007.02.008

Kasauli, R., Liebel, G., Knauss, E., Gopakumar, S., & Kanagwa, B. (2017). Requirements engineering challenges in large-scale agile system development. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 352-361): IEEE.

Kerzner, H. (2014). *Project recovery: Case studies and techniques for overcoming project failure*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Kitto, S. C., Chesters, J., & Grbich, C. (2008). Quality in qualitative research. *The Medical journal of Australia, 188*(4), 243-246.

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM, 38*(3), 69-81. doi:10.1145/203330.203345

Krueger, R. A., & Casey, M. A. (2015). *Focus groups: A practical guide for applied research* (5th edition. ed.). Thousand Oaks, California: SAGE.

© University of Pretoria

Kunda, D., Mulenga, M., Sinyinda, M., & Chama, V. (2018). Challenges of Agile development and implementation in a developing country: A Zambia case study. *Journal of Computer Science, 14*(5), 585-600. doi:10.3844/jcssp.2018.585.600

Kvale, S., & Brinkmann, S. (2015). *InterViews: Learning the craft of qualitative research interviewing* (Third edition. ed.). Los Angeles: Sage Publications.

Larman, C., & Vodde, B. (2017). *Large-scale scrum: More with less*. In The Addison-Wesley signature series. Retrieved from http://ptgmedia.pearsoncmg.com/images/9780321985712/samplepages/9780321985712.pdf

Lawrence, L. (2015). Validity, reliability, and generalizability in qualitative research. *Journal of Family Medicine and Primary Care, 4*(3), 324-327. Retrieved from doi:10.4103/2249-4863.161306

Lehtinen, T. O. A., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures - An analysis of their relationships. *Information and Software Technology, 56*(6), 623-643. doi:10.1016/j.infsof.2014.01.015

Littlejohn, S. W., & Foss, K. A. (2005). *Theories of human communication* (8th ed. ed.). Belmont, Calif.: Thomson/Wadsworth.

Marshall, C., & Rossman, G. B. (2006). *Designing qualitative research* (4th ed. ed.). Thousands Oaks, Calif.: Sage Publications.

McCutcheon, G., & Jung, B. (1990). Alternative perspectives on action research. *Theory into Practice, 29*(3), 144-151.

Measey, P. (2015). *Agile foundation: Principles, practices and frameworks*. London: BCS.

Medeiros, J., Alves, D. C. P., Vasconcelos, A., Silva, C., & Wanderley, E. G. (2015). *Requirements engineering in agile projects: A systematic mapping based in evidences of industry*.

Medeiros, J., Vasconcelos, A., Goulão, M., Silva, C., & Araújo, J. (2017). An approach based on design practices to specify requirements in agile projects. In *Proceedings of the Symposium on Applied Computing* (pp. 1114-1121).

Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2018). Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *The Journal of Systems & Software, 142*, 171-194. doi:10.1016/j.jss.2018.04.064

Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2020). Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology, 117*. doi:10.1016/j.infsof.2019.106194

Mendez-Fernandez, D., Wagner, S., Kalinowski, N., Felderer, M., Mafra, P., Vetro, A., . . . Wieringa, R. J. (2017). Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical software engineering journal, 22*
(5).

Mircea, E. (2019). Project management using agile frameworks. *Economy Informatics, 19*(1), 34-44. Retrieved from doi:10.12948/ei2019.01.04

Mishra, D., & Mishra, A. (2009). Effective communication, collaboration, and coordination in eXtreme Programming: Human-centric perspective in a small organization. *Human Factors and Ergonomics in Manufacturing & Service Industries, 19*(5), 438-456. doi:10.1002/hfm.20164

Moore, G. A. (1999). *Crossing the chasm: Marketing and selling high-tech products to mainstream customers* (Rev. ed. ed.). New York: HarperBusiness.

Mountain Goat Software. (n.d.). Planning Poker. Retrieved from https://www.mountaingoatsoftware.com/agile/planning-poker

Mtsweni, E. S., & Mavetera, N. (2019). Soft issues that limit sharing of tacit knowledge within software development project teams. In *2019 IEEE AFRICON* (pp. 1-6): IEEE.

Mullarkey, M. T., Hevner, A. R., & Ågerfalk, P. (2019). An elaborated action design research process model. *European Journal of Information Systems, 28*(1), 6-20. doi:10.1080/0960085X.2018.1451811

Muszyńska, K. (2018). A concept for measuring effectiveness of communication in project teams. *Journal of Economics and Management, 33*(3), 63-79. Retrieved from doi:10.22367/jem.2018.33.04

Nakatsu, R. T., & Iacovou, C. L. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. *Information & Management, 46*(1), 57-68. doi:10.1016/j.im.2008.11.005

Nidhra, S., Yanamadala, M., Afzal, W., & Torkar, R. (2013). Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation. *International Journal of Information Management, 33*(2), 333-355. doi:10.1016/j.ijinfomgt.2012.11.004

Okoli, C. (2015). A Guide to conducting a standalone systematic literature review. *Communications of the Association for Information Systems, 37*(1).

Orb, A., Eisenhauer, L., & Wynaden, D. (2001). Ethics in Qualitative Research. *Journal of Nursing Scholarship, 33*(1), 93-96.

© University of Pretoria

Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2018). Large-scale agile transformation at Ericsson: A case study. *Empirical Software Engineering : An International Journal, 23*(5), 2550-2596. doi:10.1007/s10664-017-9555-8

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.* (pp. 308-313): IEEE.

Parviainen, P., & Tihinen, M. (2014). Knowledge-related challenges and solutions in GSD. *Expert Systems, 31*(3), 253-266. doi:10.1111/exsy.608

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45-77.

Pernstål, J., Feldt, R., Gorschek, T., & Florén, D. (2019). Communication problems in software development: A model and its industrial application. *International Journal of Software Engineering and Knowledge Engineering, 29*(10), 1497-1538. doi:10.1142/S0218194019500475

Phillippi, J., & Lauderdale, J. (2018). A Guide to field notes for qualitative research: Context and conversation. *Qualitative health research, 28*(3), 381-388. doi:10.1177/1049732317697102

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering : An International Journal, 13*(3), 303-337. doi:10.1007/s10664-008-9065-9

Platinum Edge Agile Experts (2021). [Certified Scrum Master].

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: An empirical study. *Information Systems Journal, 20*(5), 449-480. doi:10.1111/j.1365-2575.2007.00259.x

Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., . . . Wu, Z. (2021). Requirement engineering challenges in Agile software development. *Mathematical Problems in Engineering, 2021*, 1-18. doi:10.1155/2021/6696695

Reed, A. H., & Knight, L. V. (2010a). Effect of a virtual project team environment on communication-related project risk. *International Journal of Project Management, 28*(5), 422-427. doi:10.1016/j.ijproman.2009.08.002

Reed, A. H., & Knight, L. V. (2010b). Project risk differences between virtual and co-located teams. *Journal of Computer Information Systems, 51*(1), 19-30.

Reifer, D. J., Maurer, F., & Erdogmus, H. (2003). Scaling Agile Methods. *IEEE Software, 20*(4).

Ricca, F., Reggio, G., Astesiano, E., Scanniello, G., & Torchiano, M. (2014). Assessing the effect of screen mockups on the comprehension of functional requirements. *ACM Transactions on Software Engineering and Methodology, 24*(1). doi:10.1145/2629457

Rooney, J. J., & Heuvel, L. N. V. (2004). Root cause analysis for beginners. *Quality Progress, 37*(7).

Rubin, K. S. (2017). *Essential Scrum: A practical guide to the most popular agile process*. Upper Saddle River, NJ: Addison-Wesley.

Runeson, P. (2012). *Case study research in software engineering: Guidelines and examples*. In. Retrieved from https://ebookcentral-proquest-com.uplib.idm.oclc.org/lib/pretoria-ebooks/detail.action?docID=818522

Samarawickrama, S. S., & Perera, I. (2017). Continuous Scrum: A framework to enhance Scrum with DevOps. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 1-7): IEEE.

Sarker, S., & Sahay, S. (2004). Implications of space and time for distributed work: An interpretive study of US-Norwegian systems development teams. *European Journal of Information Systems, 13*(1), 3-20. Retrieved from doi:10.1057/palgrave.ejis.3000485

Schön, E., Thomaschewski, J., & JoEscalona, M. J. (2017). Agile requirements engineering: A systematic literature review. *Computer Standards & Interfaces, 49*, 79-91. doi:10.1016/j.csi.2016.08.011

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall.

Schwaber, K., & Sutherland, J. (2020). The 2020 Scrum guide. Retrieved from https://scrumguides.org/scrum-guide.html

Scopus. (2021). Scopus: access and use support center. Retrieved from https://service.elsevier.com/app/answers/detail/a_id/15100/supporthub/scopus/

Sebega, Y., & Mnkandla, E. (2017). Exploring issues in Agile requirements engineering in the South African software industry. *The Electronic Journal of Information Systems in Developing Countries, 81*(1), 1-18. doi:10.1002/j.1681-4835.2017.tb00597.x

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly, 35*(1), 37-56. doi:10.2307/23043488

Shah, U. S., Jinwala, D. C., & Patel, S. J. (2016). An excursion to software development life cycle models an old to ever-growing models. *ACM SIGSOFT Software Engineering Notes, 41*(1), 1-6. doi:10.1145/2853073.2853080

© University of Pretoria

Sharma, A., Sengupta, S., & Gupta, A. (2011). Exploring risk dimensions in the Indian software industry. *Project Management Journal, 42*(5), 78-91. doi:10.1002/pmj.20258

Shrivastava, S. V., & Rathod, U. (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology, 58*, 373-387. doi:10.1016/j.infsof.2014.07.007

Skelton, M., & Betteley, J. (2017). Merging Agile and DevOps. Retrieved from https://www.infoq.com/articles/merging-devops-agile/

Stott, W. (2003). Extreme programming: Turning the world upside down. *Computing and Control Engineering, 14*(3), 18-23. doi:10.1049/cce:20030303

Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed. ed.). Thousand Oaks: Sage Publications.

Sutherland, J. (2020). Swarming: How to instantly boost your Scrum team's productivity. Retrieved from https://www.scruminc.com/swarming-instantly-boost-scrum-team-productivity/

Tactivos Inc. dba MURAL. (2021). Let's transform teamwork. Retrieved from https://www.mural.co/

Tiwari, S., Rathore, S. S., & Gupta, A. (2012). *Selecting requirement elicitation techniques for software projects*. Paper presented at the 2012 CSI Sixth International Conference on Software Engineering (CONSEG).

Vlietland, J., & van Vliet, H. (2014). Alignment issues in chains of scrum teams. *Lecture Notes in Business Information Processing, 182 LNBIP*, 301-306. doi:10.1007/978-3-319-08738-2

vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to Design Science Research. In *Design Science Research. Cases* (pp. 1-13): Springer International Publishing.

Wells, D. (1999). When should Extreme Programming be used? Retrieved from http://www.extremeprogramming.org/when.html

Wieringa, R. (2014). *Design science methodology for information systems and software engineering*. Heidelberg: Springer.

Wiersinga, W. J., & Prescott, H. C. (2020). What is COVID-19? *JAMA, 324*(8).

Yin, R. K. (2014). *Case study research: Design and methods* (5 edition. ed.). Thousand Oaks, California: SAGE.

You, E. (2021). What is Vue.js? Retrieved from https://vuejs.org/

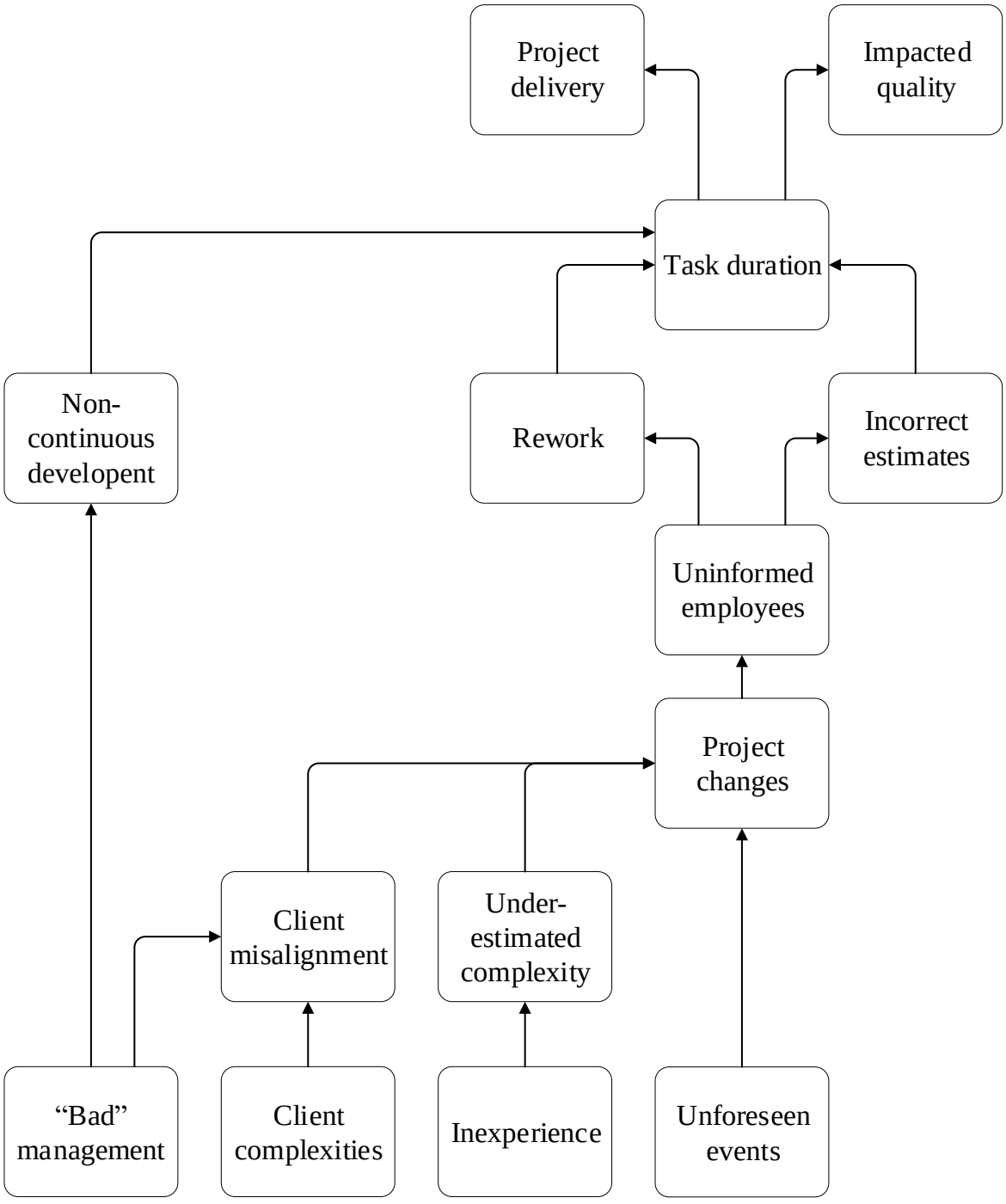
Zahedi, M., Shahin, M., & Ali Babar, M. (2016). A systematic review of knowledge sharing challenges and practices in global software development. *International Journal of Information Management, 36*(6 Part A), 995-1019. doi:10.1016/j.ijinfomgt.2016.06.007

© University of Pretoria

Appendices

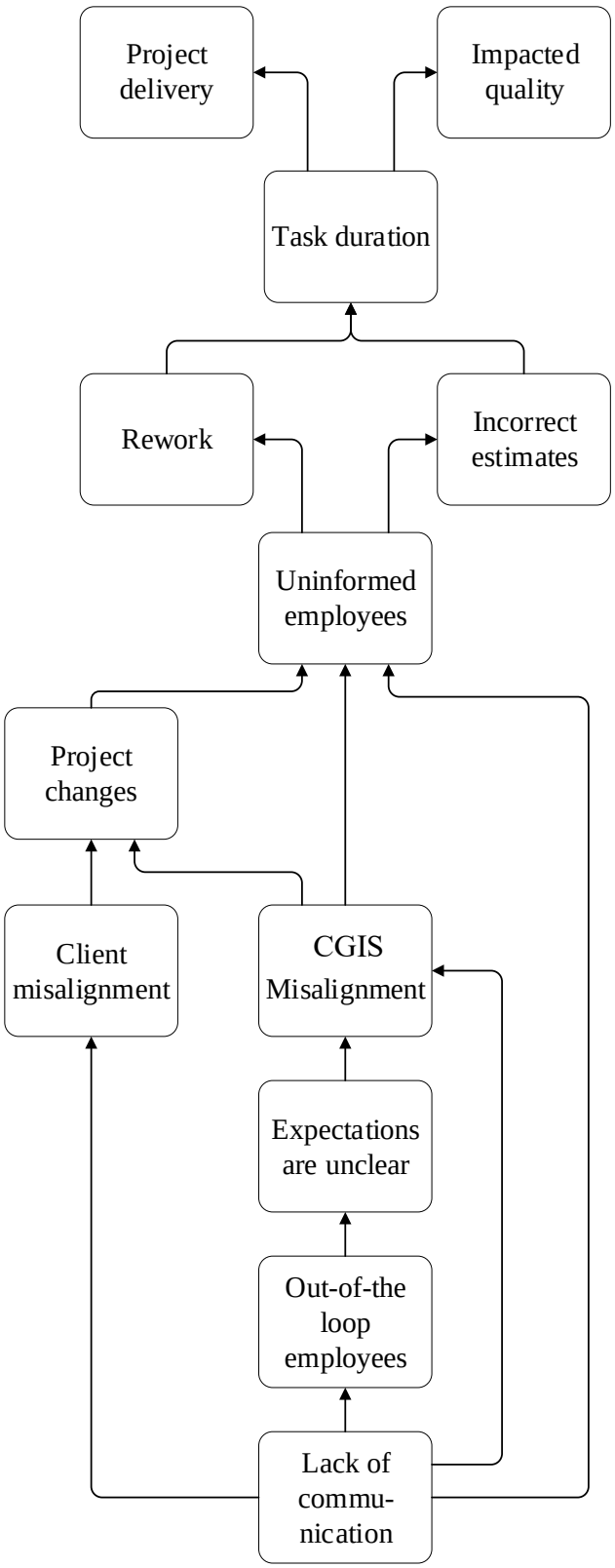
Appendix A

Root cause(s): Bad management, client complexities, inexperience, and unforeseen events



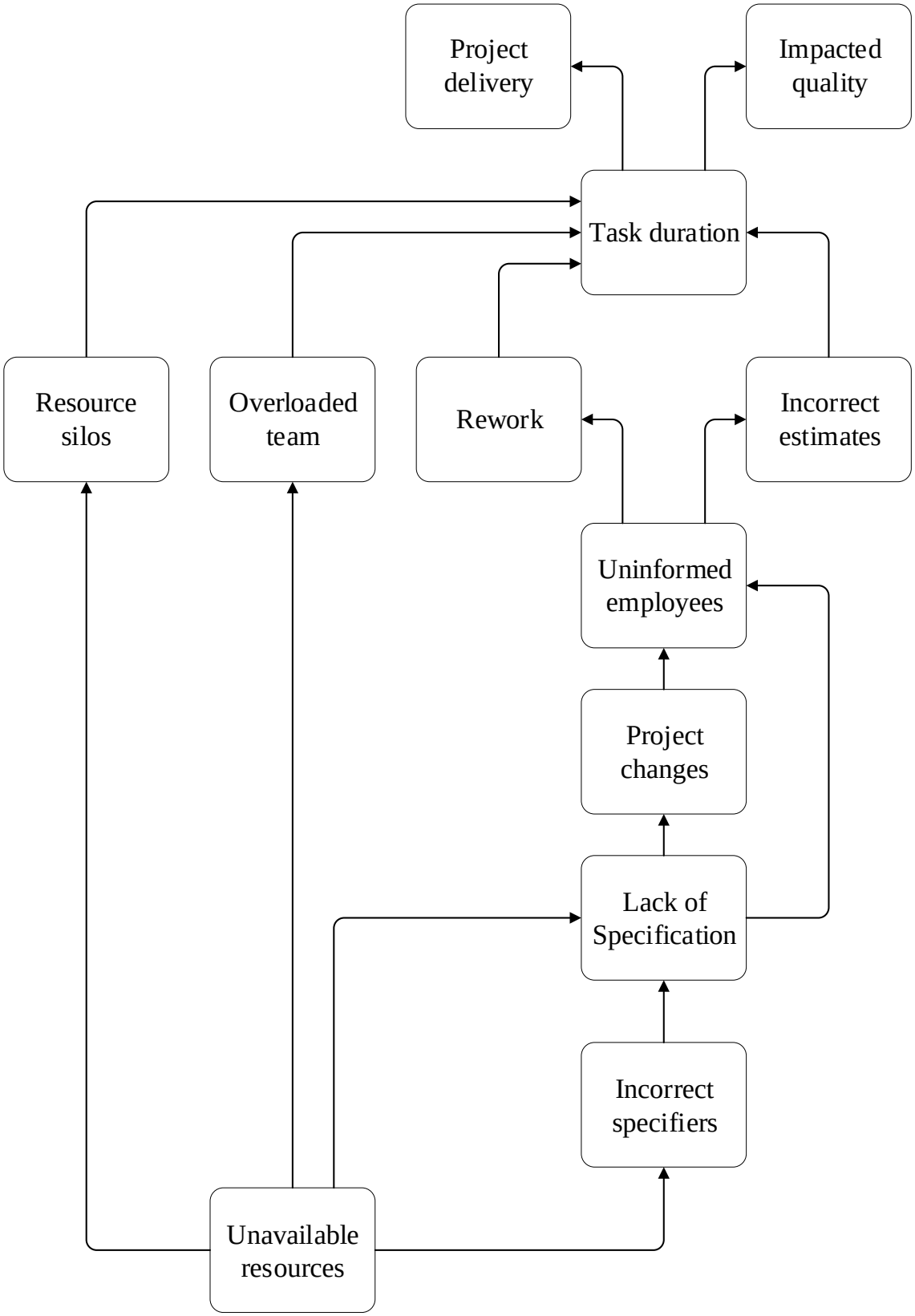
© University of Pretoria

Root cause(s): Lack of communication



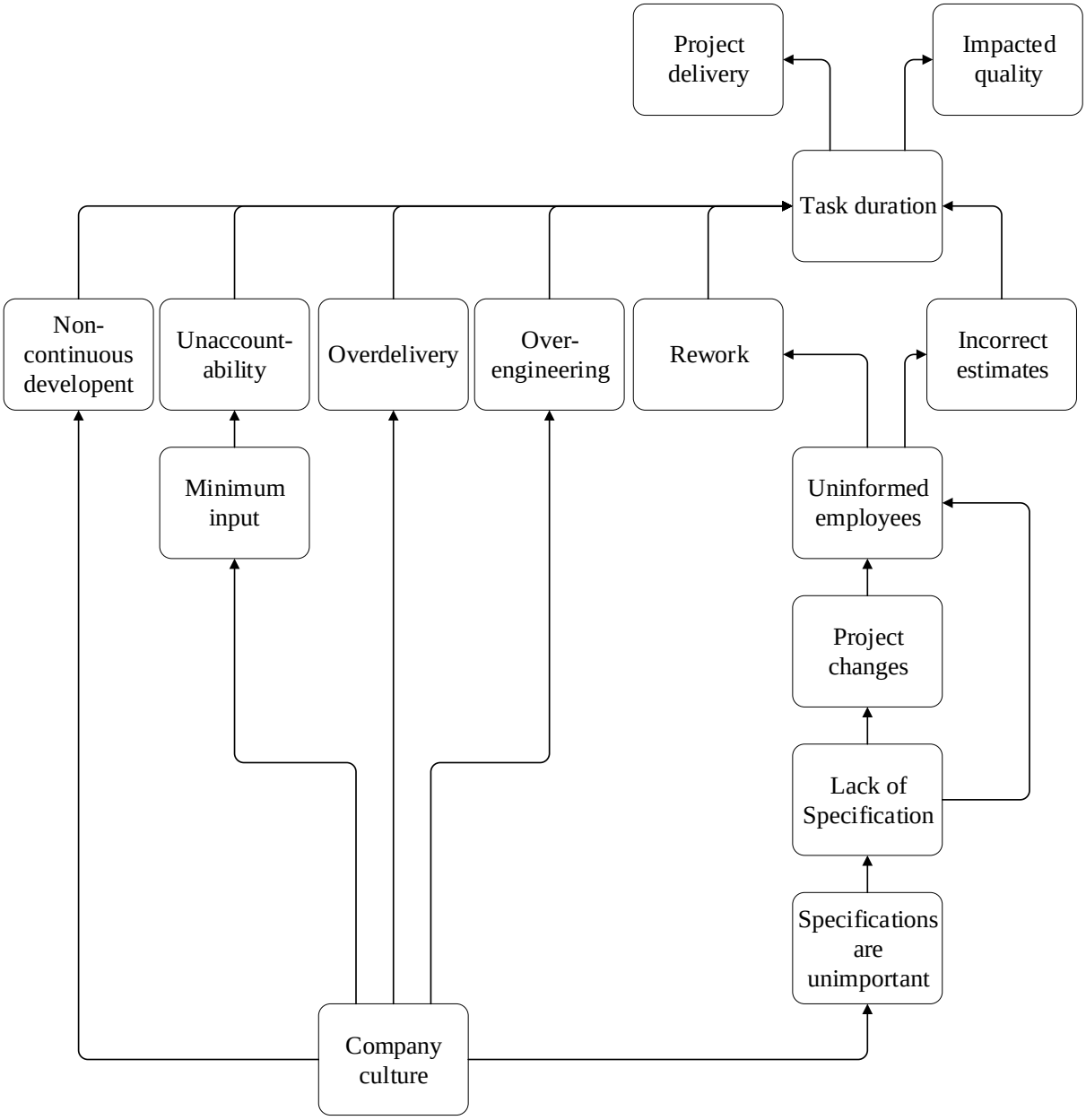
© University of Pretoria

Root cause(s): Unavailable resources



© University of Pretoria

Root cause(s): Company culture

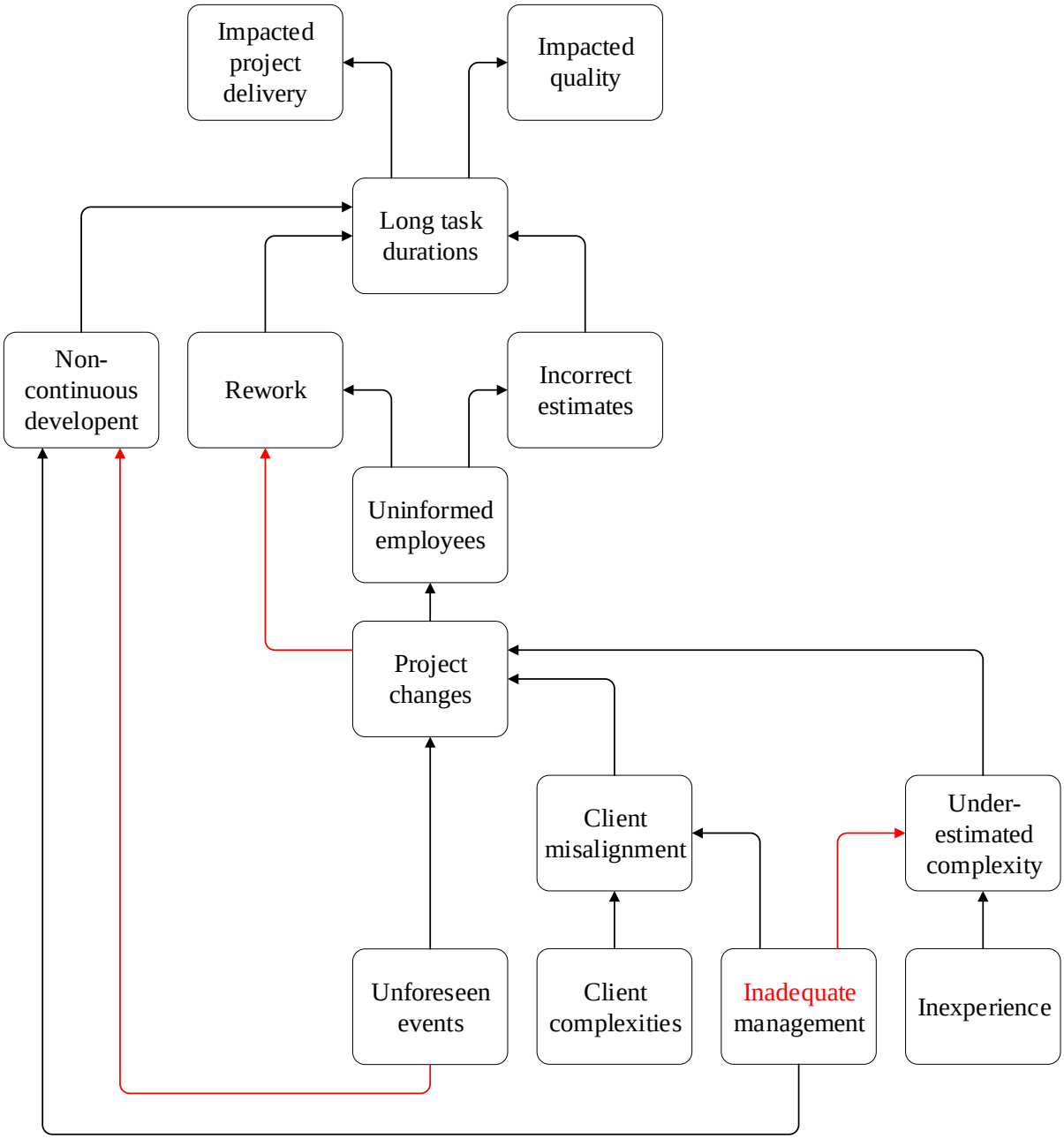


© University of Pretoria

Appendix B

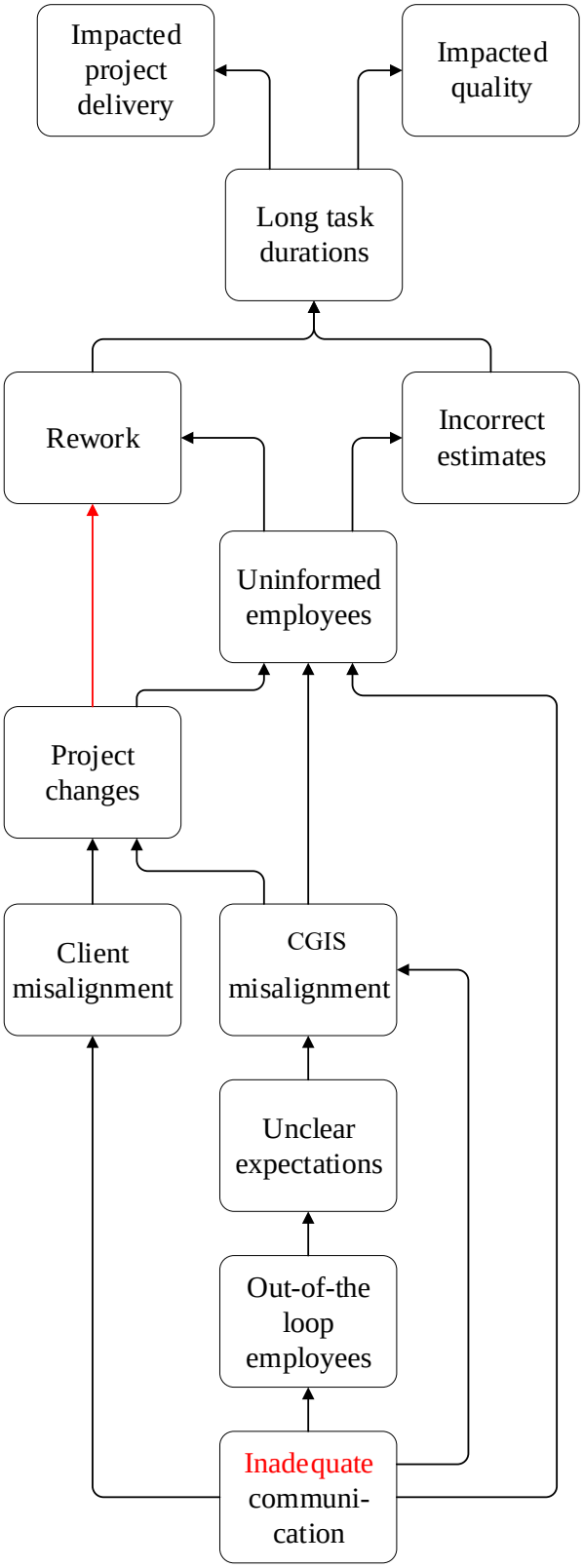
Alterations recommended by the focus group participants are depicted in *red*.

Root cause(s): Unforeseen events, client complexities, inadequate management, and inexperience



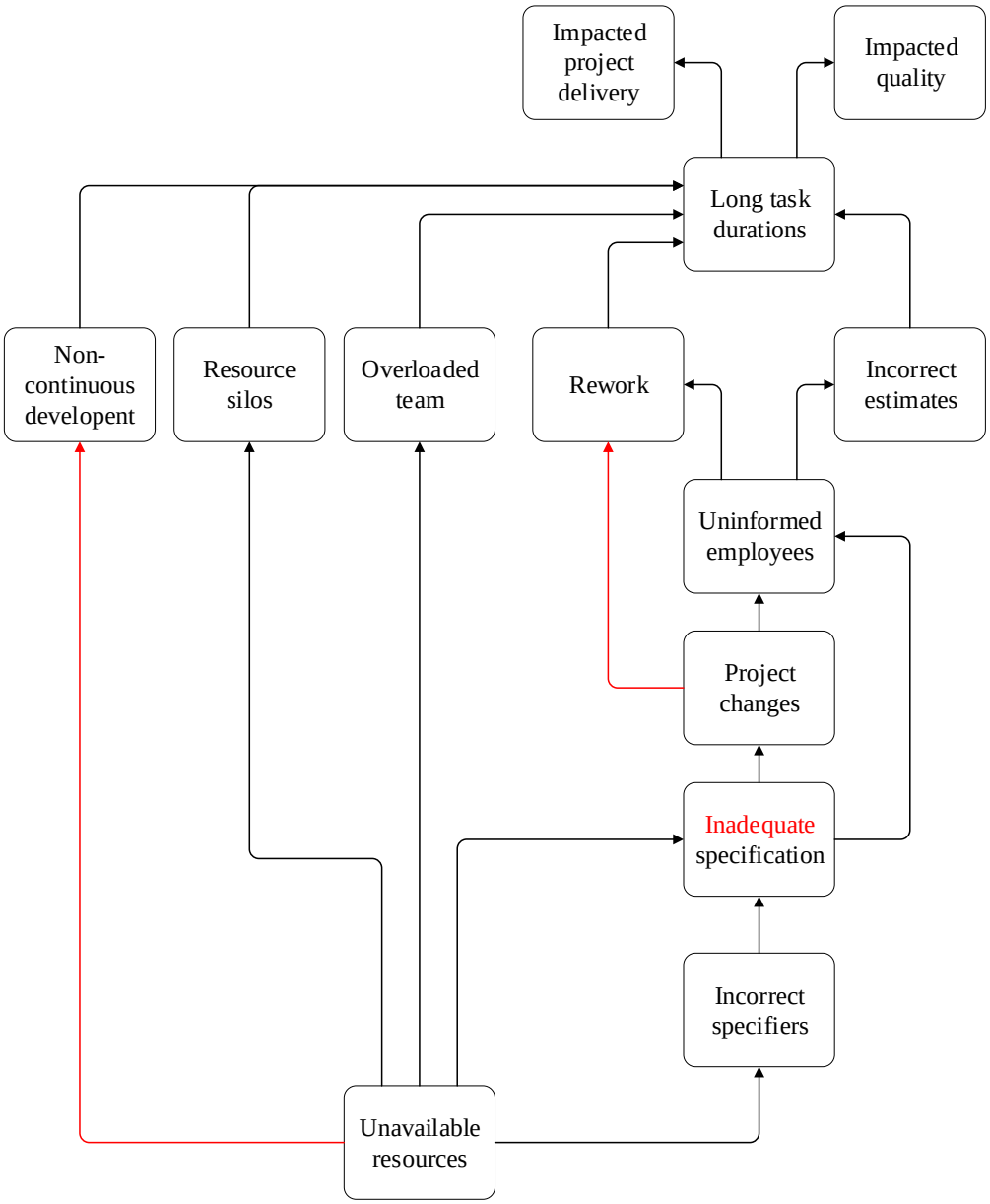
© University of Pretoria

Root cause(s): Inadequate communication



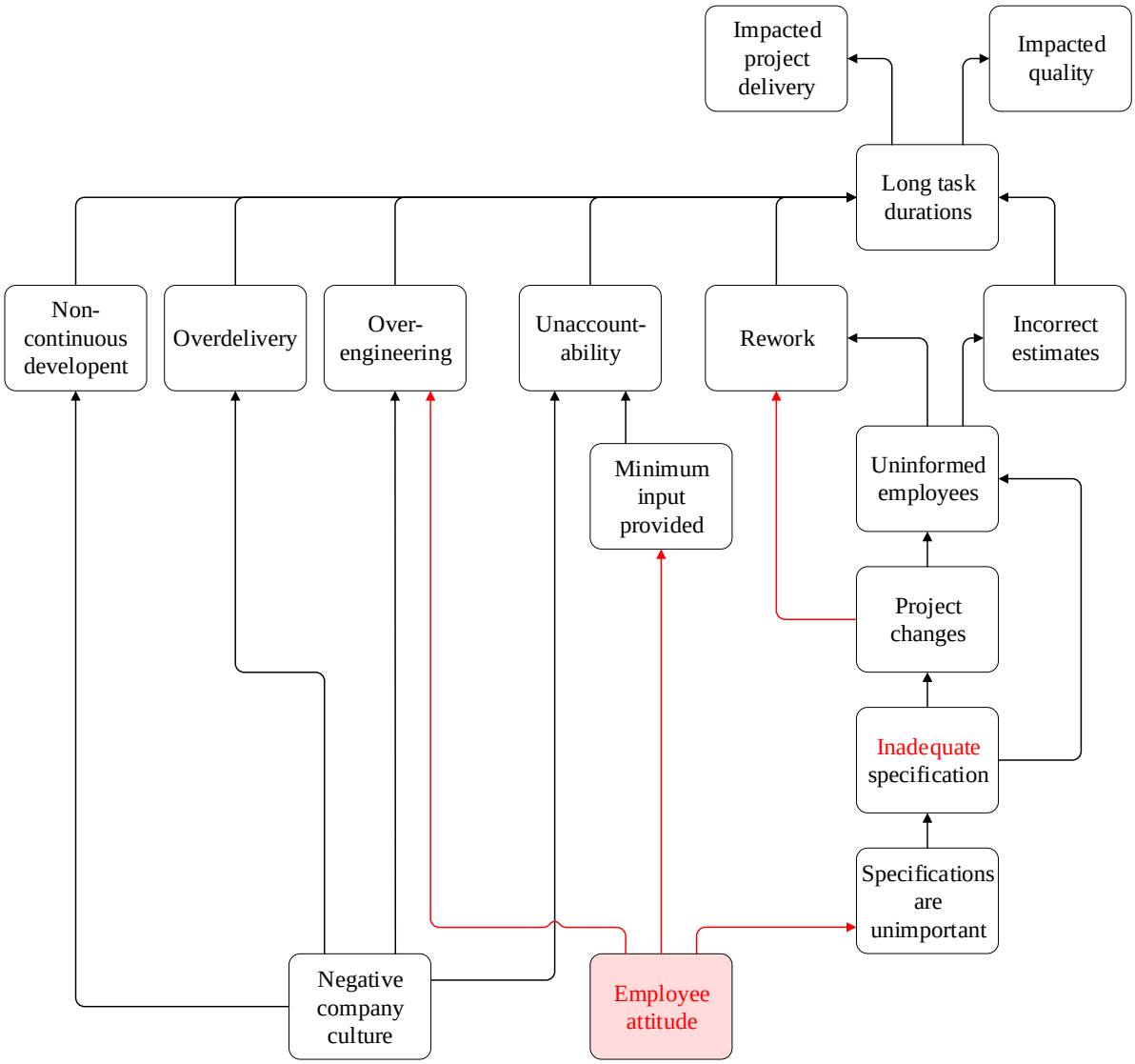
© University of Pretoria

Root cause(s): Unavailable resources



© University of Pretoria

Root cause(s): Negative company culture and employee attitude



© University of Pretoria

Appendix C

Author(s)	Source title	Year	Publisher	SLRs
C. L. Iacovou	A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study	2009	Elsevier	Principle
R. T. Nakatsu	A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study	2009	Elsevier	Principle
I. Inayat	A systematic literature review on Agile requirements engineering practices and challenges	2015	Wiley	SLR
M. Daneva	A systematic literature review on Agile requirements engineering practices and challenges	2015	Wiley	SLR
S. Marczak	A systematic literature review on Agile requirements engineering practices and challenges	2015	Wiley	SLR
S. S. Salim	A systematic literature review on Agile requirements engineering practices and challenges	2015	Wiley	SLR
S. Shamshirband	A systematic literature review on Agile requirements engineering practices and challenges	2015	Wiley	SLR
G. S. Walia	A systematic literature review to identify and classify software requirement errors	2009	Elsevier	SLR
J. C. Carver	A systematic literature review to identify and classify software requirement errors	2009	Elsevier	SLR
M. A. Babar	A systematic review of knowledge sharing challenges and practices in global software development	2016	Elsevier	SLR
M. Shahin	A systematic review of knowledge sharing challenges and practices in global software development	2016	Elsevier	SLR
M. Zahedi	A systematic review of knowledge sharing challenges and practices in global software development	2016	Elsevier	SLR
S. Huang	An empirical analysis of risk components and performance on software projects	2007	Elsevier	Principle
W. Han	An empirical analysis of risk components and performance on software projects	2007	Elsevier	Principle
A. Mockus	An empirical study of speed and communication in globally distributed software development	2003	IEEE	Principle
J. D. Herbsleb	An empirical study of speed and communication in globally distributed software development	2003	IEEE	Principle
R. J. Sweis	An investigation of failure in information systems projects: The case of Jordan	2015	SAGE	Principle
S. V. Shrivastava	Categorization of risk factors for distributed Agile projects	2015	Elsevier	Principle
U. Rathod	Categorization of risk factors for distributed Agile projects	2015	Elsevier	Principle
A. Vizcaıno	Challenges and improvements in distributed software development: A systematic review	2009	Elsevier	SLR
M. Jimenez	Challenges and improvements in distributed software development: A systematic review	2009	Elsevier	SLR
M. Piattini	Challenges and improvements in distributed software development: A systematic review	2009	Elsevier	SLR
A. César	Challenges and solutions in distributed software development project management: A systematic literature review	2010	IEEE	SLR
C. Costa	Challenges and solutions in distributed software development project management: A systematic literature review	2010	IEEE	SLR
C. França	Challenges and solutions in distributed software development project management: A systematic literature review	2010	IEEE	SLR
F. Q. B. da Silva	Challenges and solutions in distributed software development project management: A systematic literature review	2010	IEEE	SLR
R. Prikladinicki	Challenges and solutions in distributed software development project management: A systematic literature review	2010	IEEE	SLR
H. J. Rognerud	Challenges in enterprise software integration- An industrial study using repertory grids	2007	IEEE	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
J. E. Hannay	Challenges in enterprise software integration- An industrial study using repertory grids	2007	IEEE	Principle
I. Richardson	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
K. Faisal	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
M. Alshayeb	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
M. Niazi	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
M. R. Riaz	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
N. Cerpa	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
S. Mahmood	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
S. U. Khan	Challenges of project management in global software development- A client-vendor analysis	2016	Elsevier	SLR
A. Mohamed	Chaos issues on communication in Agile global software development	2012	IEEE	Principle
N. H. Arshad	Chaos issues on communication in Agile global software development	2012	IEEE	Principle
N. K. Kamaruddin	Chaos issues on communication in Agile global software development	2012	IEEE	Principle
J. Zhang	Comparing senior executive and project manager perceptions of IT project risk: A Chinese Delphi study	2010	Wiley	Principle
M. Keil	Comparing senior executive and project manager perceptions of IT project risk: A Chinese Delphi study	2010	Wiley	Principle
S. Liu	Comparing senior executive and project manager perceptions of IT project risk: A Chinese Delphi study	2010	Wiley	Principle
T. Chen	Comparing senior executive and project manager perceptions of IT project risk: A Chinese Delphi study	2010	Wiley	Principle
D. Vavpotič	Diagnosing organizational risks in software projects: Stakeholder resistance	2015	Elsevier	Principle
M. Krisper	Diagnosing organizational risks in software projects: Stakeholder resistance	2015	Elsevier	Principle
S. L.R. Vrhovec	Diagnosing organizational risks in software projects: Stakeholder resistance	2015	Elsevier	Principle
T. Hovelja	Diagnosing organizational risks in software projects: Stakeholder resistance	2015	Elsevier	Principle
D. E. Damian	Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring	2007	Elsevier	Principle
J. L. N. Audy	Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring	2007	Elsevier	Principle
R. Prikladinicki	Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring	2007	Elsevier	Principle
T. C. de Oliveira	Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring	2007	Elsevier	Principle
T. Addison	E-commerce project development risks: Evidence from a Delphi survey	2003	Elsevier	Principle
A. H. Reed	Effect of a virtual project team environment on communication-related project risk	2010	IEEE	Principle
L. V. Knight	Effect of a virtual project team environment on communication-related project risk	2010	IEEE	Principle
D. S. M da Silva	Elicitation of communication inherent risks in distributed software development	2012	IEEE	Principle
H. P. de Moura	Elicitation of communication inherent risks in distributed software development	2012	IEEE	Principle
I. H. de Farias Junior	Elicitation of communication inherent risks in distributed software development	2012	IEEE	Principle
R. R. de Azevedo	Elicitation of communication inherent risks in distributed software development	2012	IEEE	Principle
A. Al-Ani	Empirical studies of geographically distributed Agile development	2014	Springer	Principle
A. Q. Gill	Empirical studies of geographically distributed Agile development	2014	Springer	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
Y. I. Alzoubi	Empirical studies of geographically distributed Agile development	2014	Springer	Principle
A. Gupta	Exploring risk dimensions in the Indian software industry	2011	Wiley	Principle
A. Sharma	Exploring risk dimensions in the Indian software industry	2011	Wiley	Principle
S. Sengupta	Exploring risk dimensions in the Indian software industry	2011	Wiley	Principle
A. Y. Gheni	Factors affecting global virtual teams' performance in software projects	2016	Little Lion Scientific	SLR
M. A. Jabar	Factors affecting global virtual teams' performance in software projects	2016	Little Lion Scientific	SLR
N. M. Ali	Factors affecting global virtual teams' performance in software projects	2016	Little Lion Scientific	SLR
Y. Y. Jusoh	Factors affecting global virtual teams' performance in software projects	2016	Little Lion Scientific	SLR
D. J. Paulish	Global software development at Siemens: Experience from nine projects	2005	ACM	Principle
J. D. Herbsleb	Global software development at Siemens: Experience from nine projects	2005	ACM	Principle
M. Bass	Global software development at Siemens: Experience from nine projects	2005	ACM	Principle
B. Fitzgerald	Global software development challenges: A case study on temporal, geographical and socio-cultural distance	2006	Elsevier	Principle
E. O. Conchúir	Global software development challenges: A case study on temporal, geographical and socio-cultural distance	2006	Elsevier	Principle
H. Holmstrom	Global software development challenges: A case study on temporal, geographical and socio-cultural distance	2006	Elsevier	Principle
P. J. Ågerfalk	Global software development challenges: A case study on temporal, geographical and socio-cultural distance	2006	Elsevier	Principle
M. A. Joarder	Goal and risk factors in offshore outsourced software development from vendor's viewpoint	2009	IEEE	Principle
S. H. Houmb	Goal and risk factors in offshore outsourced software development from vendor's viewpoint	2009	IEEE	Principle
S. Islam	Goal and risk factors in offshore outsourced software development from vendor's viewpoint	2009	IEEE	Principle
L. G. Wallgren	Human factors related challenges in software engineering - An industrial perspective	2015	IEEE	Principle
P. Lenberg	Human factors related challenges in software engineering - An industrial perspective	2015	IEEE	Principle
R. Feldt	Human factors related challenges in software engineering - An industrial perspective	2015	IEEE	Principle
M. A. Khan	Identification of risks in Pakistani IT companies: A survey paper	2014	IEEE	Principle
M. Abbas	Identification of risks in Pakistani IT companies: A survey paper	2014	IEEE	Principle
M. F. Khan	Identification of risks in Pakistani IT companies: A survey paper	2014	IEEE	Principle
C. Caprano	Identifying and structuring challenges in large-scale Agile development based on a structured literature review	2018	IEEE	Principle
F. Matthes	Identifying and structuring challenges in large-scale Agile development based on a structured literature review	2018	IEEE	Principle
M. Kleehaus	Identifying and structuring challenges in large-scale Agile development based on a structured literature review	2018	IEEE	Principle
O. Uludag	Identifying and structuring challenges in large-scale Agile development based on a structured literature review	2018	IEEE	Principle
H. Hijazi	Identifying causality relation between software projects risk factors	2014	Science and Engineering Research Support Society	Principle
H. Muaidi	Identifying causality relation between software projects risk factors	2014	Science and Engineering Research Support Society	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
S. Alqrainy	Identifying causality relation between software projects risk factors	2014	Science and Engineering Research Support Society	Principle
T. Khdour	Identifying causality relation between software projects risk factors	2014	Science and Engineering Research Support Society	Principle
A. Idri	Identifying risks of software project management in global software development: An integrative framework	2016	IEEE	SLR
J. L. Fernandez-Aleman	Identifying risks of software project management in global software development: An integrative framework	2016	IEEE	SLR
J. Nicolas Ros	Identifying risks of software project management in global software development: An integrative framework	2016	IEEE	SLR
S. Y. Chadli	Identifying risks of software project management in global software development: An integrative framework	2016	IEEE	SLR
A. Mursu	Identifying software project risks in Nigeria: An international comparative study	2003	Springer	Principle
H. A. Soriyan	Identifying software project risks in Nigeria: An international comparative study	2003	Springer	Principle
K. Lyytinen	Identifying software project risks in Nigeria: An international comparative study	2003	Springer	Principle
M. Korpela	Identifying software project risks in Nigeria: An international comparative study	2003	Springer	Principle
K. Lyytinen	Identifying software project risks: An international Delphi study	2001	Taylor & Francis	Principle
M. Keil	Identifying software project risks: An international Delphi study	2001	Taylor & Francis	Principle
P. Cule	Identifying software project risks: An international Delphi study	2001	Taylor & Francis	Principle
R. Schmidt	Identifying software project risks: An international Delphi study	2001	Taylor & Francis	Principle
B. Corbitt	Impediments to requirements engineering during global software development	2007	Springer	Principle
J. Hanisch	Impediments to requirements engineering during global software development	2007	Springer	Principle
S. Sahay	Implications of space and time for distributed work: An interpretive study of US–Norwegian systems development teams	2004	Springer	Principle
S. Sarker	Implications of space and time for distributed work: An interpretive study of US–Norwegian systems development teams	2004	Springer	Principle
D. Tesch	IT project risk factors: The project management professionals' perspective	1985	IEEE	Principle
M. N. Frolick	IT project risk factors: The project management professionals' perspective	1985	IEEE	Principle
T. J. Kloppenborg	IT project risk factors: The project management professionals' perspective	1985	IEEE	Principle
A. E. Akgün	Knowledge sharing barriers in software development teams: A multiple case study in Turkey	2017	Emerald	Principle
H. Ayar	Knowledge sharing barriers in software development teams: A multiple case study in Turkey	2017	Emerald	Principle
H. Keskin	Knowledge sharing barriers in software development teams: A multiple case study in Turkey	2017	Emerald	Principle
Z. Okunakol	Knowledge sharing barriers in software development teams: A multiple case study in Turkey	2017	Emerald	Principle
M. Yanamadala	Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation	2013	Elsevier	SLR
R. Torkar	Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation	2013	Elsevier	SLR

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
S. Nidhra	Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation	2013	Elsevier	SLR
W. Afzal	Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation	2013	Elsevier	SLR
M. Tihinen	Knowledge-related challenges and solutions in GSD	2014	Wiley	Principle
P. Parviainen	Knowledge-related challenges and solutions in GSD	2014	Wiley	Principle
F. Steinson	Managing risks in distributed software projects: An integrative framework	2009	IEEE	SLR
J. Boeg	Managing risks in distributed software projects: An integrative framework	2009	IEEE	SLR
J. S. Persson	Managing risks in distributed software projects: An integrative framework	2009	IEEE	SLR
L. Mathiassen	Managing risks in distributed software projects: An integrative framework	2009	IEEE	SLR
T. S. Madsen	Managing risks in distributed software projects: An integrative framework	2009	IEEE	SLR
C. Lassenius	Perceived causes of software project failures – An analysis of their relationships	2014	Elsevier	Principle
J. Itkonen,	Perceived causes of software project failures – An analysis of their relationships	2014	Elsevier	Principle
J. Vanhanen	Perceived causes of software project failures – An analysis of their relationships	2014	Elsevier	Principle
M. V. Mäntylä	Perceived causes of software project failures – An analysis of their relationships	2014	Elsevier	Principle
T. O. A. Lehtinen	Perceived causes of software project failures – An analysis of their relationships	2014	Elsevier	Principle
B. Tessem	Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts	2011	Elsevier	Principle
J. Iden	Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts	2011	Elsevier	Principle
T. Päivärinta	Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts	2011	Elsevier	Principle
A. H. Reed	Project duration and risk factors on virtual projects	2013	Taylor & Francis	Principle
L. V. Knight	Project duration and risk factors on virtual projects	2013	Taylor & Francis	Principle
M. Daneva	Quality requirements challenges in the context of large-scale distributed Agile: An empirical study	2019	Elsevier	Principle
R. Wieringa	Quality requirements challenges in the context of large-scale distributed Agile: An empirical study	2019	Elsevier	Principle
W. Alsaqaf	Quality requirements challenges in the context of large-scale distributed Agile: An empirical study	2019	Elsevier	Principle
D. E. Damian	RE challenges in multi-site software development organisations	2003	Springer	Principle
D. Zowghi	RE challenges in multi-site software development organisations	2003	Springer	Principle
A. Bush	Reconciling user and project manager perceptions of IT project risk: A Delphi study	2002	Wiley	Principle
A. Tiwana	Reconciling user and project manager perceptions of IT project risk: A Delphi study	2002	Wiley	Principle
M. Keil	Reconciling user and project manager perceptions of IT project risk: A Delphi study	2002	Wiley	Principle
B. Regnell	Requirements are slipping through the gaps - A case study on causes & effects of communication gaps in large-scale software development	2011	IEEE	Principle
E. Bjarnason	Requirements are slipping through the gaps - A case study on causes & effects of communication gaps in large-scale software development	2011	IEEE	Principle
K. Wnuk	Requirements are slipping through the gaps - A case study on causes & effects of communication gaps in large-scale software development	2011	IEEE	Principle
B. Kanagwa	Requirements engineering challenges in large-scale Agile system development	2017	IEEE	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
E. Knauss	Requirements engineering challenges in large-scale Agile system development	2017	IEEE	Principle
G. Liebel	Requirements engineering challenges in large-scale Agile system development	2017	IEEE	Principle
R. Kasauli	Requirements engineering challenges in large-scale Agile system development	2017	IEEE	Principle
S. Gopakumar	Requirements engineering challenges in large-scale Agile system development	2017	IEEE	Principle
A. G. Dahlstedt	Requirements engineering challenges in market-driven software development – an interview study with practitioners	2007	Elsevier	Principle
A. Persson	Requirements engineering challenges in market-driven software development – an interview study with practitioners	2007	Elsevier	Principle
B. Regnell	Requirements engineering challenges in market-driven software development – an interview study with practitioners	2007	Elsevier	Principle
J. Natt och Dag	Requirements engineering challenges in market-driven software development – an interview study with practitioners	2007	Elsevier	Principle
L. Karlsson	Requirements engineering challenges in market-driven software development – an interview study with practitioners	2007	Elsevier	Principle
A. Akhunzada	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
Fazal-e-Amin	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
J. Iqbal	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
M. H. N. Nasir	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
M. Khan	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
M. Shoaib	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
R. B. Ahmad	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
S. Alyahya	Requirements engineering issues causing software development outsourcing failure	2020	Public Library of Science	Principle
M. A. Alnuem	Requirements understanding: A challenge in global software development	2012	IEEE	Principle
J. F. Defranco	Review and analysis of software development team communication research	2017	IEEE	SLR
P. A. Laplante	Review and analysis of software development team communication research	2017	IEEE	SLR
C. Gusmão	Risk factors in software development projects: A systematic literature review	2018	Springer	SLR
H. Moura	Risk factors in software development projects: A systematic literature review	2018	Springer	SLR
J. Menezes Jr	Risk factors in software development projects: A systematic literature review	2018	Springer	SLR
A. Alshehab	Risk factors taxonomy in software development projects: Study from Kuwait	2005	Little Lion Scientific	Principle
H. Gaderrab	Risk factors taxonomy in software development projects: Study from Kuwait	2005	Little Lion Scientific	Principle
T. Alfozan	Risk factors taxonomy in software development projects: Study from Kuwait	2005	Little Lion Scientific	Principle
A. F. da Silva	Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms	2014	Elsevier	Principle
B. E. P Sotomonte	Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms	2014	Elsevier	Principle
C. E. S. da Silva	Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms	2014	Elsevier	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
S. M. Neves	Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms	2014	Elsevier	Principle
V. A. P. Salomon	Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms	2014	Elsevier	Principle
B. A. Kitchenham	Risks and risk mitigation in global software development: A tertiary study	2014	Elsevier	SLR
J. M. Verner	Risks and risk mitigation in global software development: A tertiary study	2014	Elsevier	SLR
M. Niazi	Risks and risk mitigation in global software development: A tertiary study	2014	Elsevier	SLR
M. Turner	Risks and risk mitigation in global software development: A tertiary study	2014	Elsevier	SLR
O. P. Brereton	Risks and risk mitigation in global software development: A tertiary study	2014	Elsevier	SLR
A. Lopez	Risks and safeguards for the requirements engineering process in global software development	2009	IEEE	SLR
A. Toval	Risks and safeguards for the requirements engineering process in global software development	2009	IEEE	SLR
J. Nicolas	Risks and safeguards for the requirements engineering process in global software development	2009	IEEE	SLR
M. D. Aundhe	Risks in offshore IT outsourcing: A service provider perspective	2009	Elsevier	Principle
S.K. Mathew	Risks in offshore IT outsourcing: A service provider perspective	2009	Elsevier	Principle
L. Mathiassen	Risks to effective knowledge sharing in Agile software teams: A model for assessing and mitigating risks	2017	Wiley	Principle
S. Ghobadi	Risks to effective knowledge sharing in Agile software teams: A model for assessing and mitigating risks	2017	Wiley	Principle
E. S. Mtsweni	Soft issues that limit sharing of tacit knowledge within software development project teams	2019	IEEE	Principle
N. Mavetera	Soft issues that limit sharing of tacit knowledge within software development project teams	2019	IEEE	Principle
G. Klein	Software development risks to project effectiveness	2000	Elsevier	Principle
J. Jiang	Software development risks to project effectiveness	2000	Elsevier	Principle
B. Paech	State of practice of user-developer communication in large scale IT projects	2011	Springer	Principle
U. Abelein	State of practice of user-developer communication in large scale IT projects	2011	Springer	Principle
F. F. Silveira	Systematic review of risks in domestic and global IT projects	2018	IGI Publishing	SLR
I. G. Júnior	Systematic review of risks in domestic and global IT projects	2018	IGI Publishing	SLR
R. F. S. Macri Russo	Systematic review of risks in domestic and global IT projects	2018	IGI Publishing	SLR
R. Sbragia	Systematic review of risks in domestic and global IT projects	2018	IGI Publishing	SLR
J. Haikara	The impact of Agile practices on communication in software development	2008	Springer	Principle
J. Still	The impact of Agile practices on communication in software development	2008	Springer	Principle
M. Pikkarainen	The impact of Agile practices on communication in software development	2008	Springer	Principle
O. Salo	The impact of Agile practices on communication in software development	2008	Springer	Principle
P. Abrahamsson	The impact of Agile practices on communication in software development	2008	Springer	Principle
P. Sonchan	Top twenty risks in software projects: A content analysis and Delphi study	2014	IEEE	Principle
S. Ramingwong	Top twenty risks in software projects: A content analysis and Delphi study	2014	IEEE	Principle

© University of Pretoria

Author(s)	Source title	Year	Publisher	SLRs
L. Wang	Understanding the impact of risks on performance in internal and outsourced information technology projects the role of strategic importance	2014	Elsevier	Principle
S. Liu	Understanding the impact of risks on performance in internal and outsourced information technology projects the role of strategic importance	2014	Elsevier	Principle
E. Oz	Why information systems projects are abandoned: A leadership and communication theory and exploratory study	1985	Taylor & Francis	Principle
J. J. Sosik	Why information systems projects are abandoned: A leadership and communication theory and exploratory study	1985	Taylor & Francis	Principle

© University of Pretoria