AN INVESTIGATION INTO REQUIREMENTS VOLATILITY USING SYSTEMS DYNAMICS MODELLING – THE DESIGN OF A REQUIREMENTS ENGINEERING RESEARCH TOOL

BY

NAUDÉ PIETER SCRIBANTE

A thesis submitted in partial fulfilment of the requirements for the degree

PHILOSOPHIAE DOCTOR

In the

**FACULTY OF ENGINEERING BUILT ENVIRONMENT
AND INFORMATION TECHNOLOGY**

UNIVERSITY OF PRETORIA

**SUPERVISOR: PROF. LEON PRETORIUS
CO-SUPERVISOR: DR SIEBERT BENADE**

June 2019

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

THESIS ABSTRACT

AN INVESTIGATION INTO REQUIREMENTS VOLATILITY USING SYSTEMS DYNAMICS MODELLING – THE DESIGN OF A REQUIREMENTS ENGINEERING RESEARCH TOOL

BY

NAUDÉ PIETER SCRIBANTE

Supervisor	:	Prof. Leon Pretorius
Co-Supervisor	:	Dr Siebert Benade
Department	:	Department of Engineering and Technology Management

UNIVERSITY OF PRETORIA

A project aims to deliver a system, product or service that satisfies the needs and desires of an end user. A large body of research exists in the field of requirements engineering, which reviews the problem of eliciting and implementing the actual requirements for a given situation. Despite all the research conducted to date, requirements remain a key challenge within the delivery of a project, product or service. The challenges in the requirements engineering domain can be traced to several factors, including that the requirements engineering domain is a complex socio-technical system. This complexity makes research in this field difficult since it is not possible to set up research experiments if the researcher is not involved and if the experiment cannot repeatedly produce the same outcome. A research approach is thus required that evaluates and researches the requirements engineering process from within the context it occurs. This approach allows the researcher to gain an increased understanding of the requirements engineering process with the aim of improving it. This thesis proposes the design of a requirements engineering research tool by following a design science research methodology. The tool can be used by researchers and requirements engineering practitioners as an investigative artefact for researching the requirements engineering domain. This thesis makes the following unique, novel and consequential contributions to the scientific and academic body of knowledge:

1. Unique, primary research contribution
    a. The development and the application of an elicitation-diffusion model for use in requirements engineering.
    b. The research methodology and method presented is not only applicable to the requirements engineering domain but also the larger

complex research domain as well as the complex socio-technical research domain.

2. Novel research contribution
    a. The application of a design science research methodology in the field of requirements engineering.
    b. The enhancement of the design cycle as used in the design science research methodology with relevant elements from the International Council on Systems Engineering (INCOSE) Systems Engineering Handbook.
    c. The definition of the object-of-study when designing a research tool to include the original artefact and problem context as the new problem context.
    d. The application of Soft Systems Methodology to the design of a research tool.
3. Consequential research contribution
    a. The use of the structure of a causal loop diagram to identify the contributing factors to observed phenomena in the requirements engineering domain.

ACKNOWLEDGEMENTS AND DEDICATION

I wish to acknowledge the following people:

My wife, Sonja, and my children, Pieter, Willie and Danie, for their continued support.

My study leaders, Professor Leon Pretorius and Doctor Siebert Benade, for guiding me without prescribing.

The subject matter experts who consented to interviews, and to my colleagues for acting as a sounding board.

"The research for a thesis is never completed; it just gets submitted."
(Pieter Scribante, 2018)

Research is like the act of untangling a ball of wool. Results are only achieved by patiently worrying each knot and loop. There are no shortcuts, and only by perseverance will the full simplicity of the answer be revealed.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

CONTENTS

FIGURES

TABLES

ABBREVIATIONS

Abbreviation	Description
COTS	Commercial off-the-shelf
FFBD	Functional flow block diagram
INCOSE	International Council on Systems Engineering
JAD	Joint application development
OoS	Object-of-study
RAD	Rapid application development
SME	Subject matter expert

GLOSSARY

Term	Description
Deductive inference	The conclusions of the inference are guaranteed to be true when its premises are true (Wieringa, 2014).
Epistemology	What the nature of knowledge is (Gregg, Kulkarni and Vinzé, 2001).
Inductive inference	The conclusion of the inference is based on the combination of the results of a single test and the results of knowledge (Wieringa, 2014).
Ontology	What the nature of reality is (Gregg *et al.*, 2001).
Research design typology	Research design classification according to general types (Lewis, 2019).
Research domain	The subject matter under study (Nunamaker, 1990).
Research method	A way of doing something, especially a systematic way; implies an orderly logical arrangement (usually in steps) (Lewis, 2019). The way of conducting and implementing research. (Adams, Khan, Raeside and White, 2007).

Term	Description
Research methodology	The branch of philosophy that analyses the principles and procedures of inquiry in a particular discipline. (Lewis, 2019). What is the approach to obtaining the desired knowledge and understanding? (Gregg *et al.*, 2001). The science and philosophy behind the research (Adams *et al.*, 2007). The combination of the process, methods, and tools that are used for conducting research within a specific domain (Nunamaker Jr., Chen and Purdin, 1990).
Research paradigm	Assumptions about knowledge – Defined by ontology, epistemology and methodology (Gregg *et al.*, 2001). How to acquire knowledge and about the physical and social world (Gregg *et al.*, 2001) – positivist, interpretivist/constructivist.

CHAPTER 1. INTRODUCTION AND RESEARCH METHODOLOGY

“The greater danger for most of us is not that our aim is too high and we miss it, but rather that it is too low and we reach it.” – Michelangelo

1.1 Introduction and problem scenario

Simon (1996) postulated in his book *The Sciences of the Artificial* that we live in a world that contains both natural elements and created artefacts. These artefacts can range from the buildings we live in and the vehicles we use for transportation through to the letter and number constructs we use to convey information. An artefact can thus be described as something that is created with a specific purpose, and which will only be of value if it provides the utility that it was designed for (Simon, 1996).

An artefact stands in contrast to a naturally occurring item such as the sun or a tree. Specific artefacts may be created to imitate items that occur naturally. Examples of these items includes product such as artificial diamonds or synthetic rubber. These items can be created using similar materials and processes as are found in nature, or from entirely different materials and using different processes. Irrespective of the process by which they are created, they are still expected to provide a specific utility or to exhibit desired properties to be of value. These desired properties or needs describe how the artefact ought to be and how it should function to achieve its goals (Simon, 1996).

When setting out to design such an artefact, the desired utility must be articulated in the form of a shared mental vision among the different stakeholders. This shared mental vision should clearly define what is required based on known and yet undiscovered needs. The need referred to in this context can be defined as a capability desired by a user or a customer to solve a problem or achieve a specific objective (INCOSE, 2017; Young, 2004).

Turning this shared mental vision into reality requires the collaboration of a multi-disciplinary team, which includes representatives from the different stakeholders who come together within the context of a project (Forsberg, Mooz and Cotterman, 2005; PMBoK, 2008). A project in this context can be defined as a temporary endeavour that is established to create a system, product, service or process (Burgelman and Maidique, 1988; Hamid, Chew and Halim, 2012; PMBoK, 2008). The project will be staffed by a project team with the primary objective of applying knowledge, skills, tools, and techniques in managing the available resources within the boundaries of cost, schedule and technical performance to meet the project requirements. In instances that the project is executed for an external customer, an additional constraint of executing the project within good customer relations comes into play (Forsberg *et al.*, 2005; Kerzner, 2013). This concept of the four constraints on project execution is shown in Figure 1.

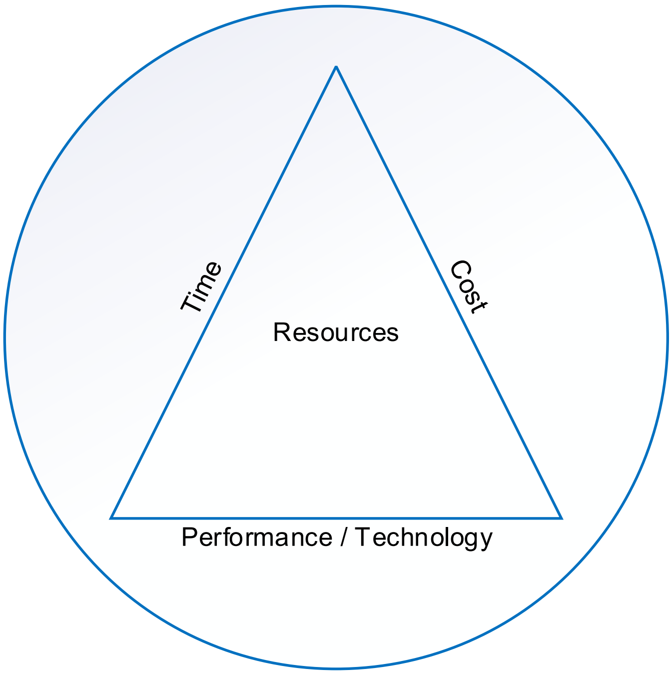


Figure 1: The four constraints of project execution [redrawn from Kerzner (2013)]

One of the first stages of the project execution process is refining and transforming the mental vision and needs of the different stakeholders into measurable requirements (Forsberg *et al.*, 2005). These requirements are the golden thread that runs through the project from the beginning to the end. At the start of the project, the requirements represent the embodiment of the shared mental vision in the form of measurable criteria. Since a project is a temporary endeavour, it must have a definite completion point somewhere in the future (PMBoK, 2008). To determine when a project has arrived at the completion point, the requirements will act as the benchmark against which the project will be validated to determine compliance with the shared mental vision. The project requirements form the bridge between the problem domain (the “what”) and the solution or implementation domain (the “how”).

Requirements engineering forms part of the systems engineering process and is concerned with eliciting or discovering, documenting and managing the requirements that embody the desired utility. Requirements can be elicited from three primary sources that include stakeholders, documentation, and existing or legacy systems (Anwar and Razali, 2012; Burnay, 2016; Honour, 2018; Zowghi and Coulin, 2005). A variety of elicitation techniques can be used, including among others interviews, questionnaires, task analysis, brainstorming, joint application development (JAD), and rapid application development (RAD) (Goguen and Linde, 1993; Zowghi *et al.*, 2005). These elicited requirements form the point of departure for the project, whether it is an internal development process or a contracted-out project.

The requirements engineering process can broadly be defined to consist of the following iterative steps: Discovery and elicitation, analysis, validation and negotiation, and triage. The documentation and specification, as well as the management activities, are ongoing activities that occur in parallel with the iterative steps (Hickey and Davis, 2004; Sommerville, 2011; Wiegers, 2000).

While the requirements engineering process may seem to be straightforward, it is one of the most complex and critical phases in the life cycle of a project. Results from research estimated that between 60% and 70% of the life cycle cost of a project has already been committed by the end of the conceptual design stages (Blanchard and Fabrycky, 1990; INCOSE, 2015). The later a change is made in the design and implementation life cycle of a project; the more severe the impact will be on the cost and the schedule of the project; potentially resulting in a failed or challenged project. Figure 2 shows the typical committed life cycle cost against time.

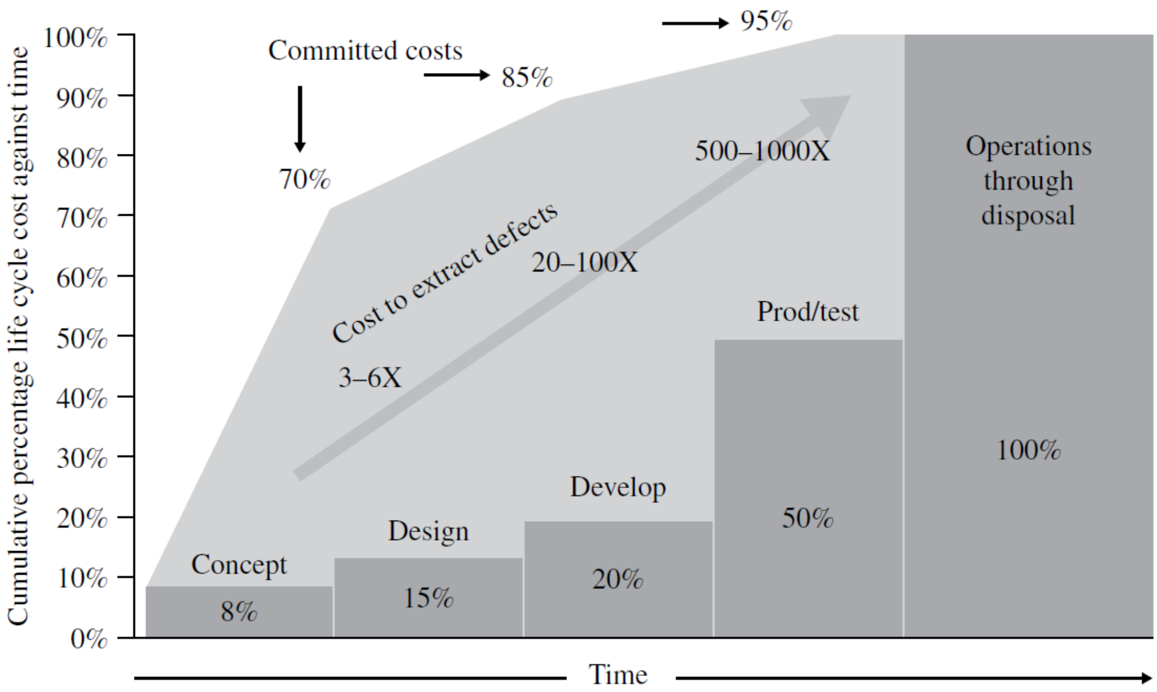


Figure 2: Committed life cycle cost vs time (INCOSE, 2015)

Executing projects successfully in the modern-day high-technology environment remains a massive challenge. In a survey of more than 8000 projects, Van Lamsweerde (2000) reported that a third of the projects were never completed and half of the projects only succeeded partially. In his review of the various projects, Van Lamsweerde (2000) identified the following contributing factors leading to failed or partially successful projects: a lack of user involvement (13%), requirements incompleteness (12%), changing requirements (11%), unrealistic expectations (6%), and unclear objectives (5%).

There has been no significant improvement over the past 15 years since Van Lamsweerde's research. In a more detailed review of software development projects, Hastie and Wojewoda (2015) analysed the Standish Group Chaos Report. They reported that over the preceding five years, the percentage of successful projects ranged between 27% and 31%, the percentage of challenged projects ranged between 49% and 56%, and the percentage of failed projects ranged between 17% and 22%. Their analysis identified that the main factors that cause projects to be challenged or failed are incomplete requirements and specifications, as well as changing requirements and specifications. These failures account for between 21% and 25% of the cases.

The complexity found in requirements engineering arises from different causes, including the maturity of the organisation, the number of stakeholders, and the type of stakeholder involvement in the process (Sheard, 2013). Different literature sources suggested that there may be as many as 32 different types of complexity spread over 12 disciplines and domains (Young, Farr and Valerdi, 2010). The effect of the complexity encountered within the requirements elicitation process is that it creates volatility in the requirements process that could result in an incorrect set of requirements being elicited. This incorrect set of requirements will be carried forward in subsequent project phases, which could lead to a failed or challenged project (Scribante, Pretorius and Benade, 2016b) .

The project requirements may also change throughout the life cycle of the project due to various other factors including errors in the original requirements, evolving customer needs, changes in the business environment or company organisational policies, incorrect stakeholder selection, the use of inappropriate elicitation techniques, stakeholder characteristics, and stakeholders changing their understanding of the need (Nurmuliani, Zowghi and Williams, 2006; Scribante, Pretorius and Benade, 2016c). In this sense, requirements elicitation is often considered to be more of a "learning" process than a "gathering" process (Reifer, 2000). What is to be expected is that the requirements elicitation process tends to be more unstable or volatile during the initial phases of the project, but should stabilise as the project progresses (Scribante *et al.*, 2016c).

1.2 Research problem statement

During the initial phases of the project life cycle, the problem space consists of the yet undiscovered needs of the stakeholders that must be converted into project requirements using the requirements engineering process. The purpose of the requirements engineering process is to bridge the gap between the problem space, which is represented by the undiscovered needs, and the solution space, which is represented by the final requirements that can be used to realise the project.

It has been argued in paragraph 1.1 that the requirements engineering process is vital to the success of a project concerning the elicitation of the correct set of requirements. It has also been established that the success rate of the requirements engineering process to deliver the correct requirements from the problem domain to the solution domain remains unsatisfactory.

One source of challenges in the requirements engineering process results from the interaction between the social element (the different stakeholders) and the technology present. This interaction creates a complex socio-technical system, which not only occurs on the creating side of the process (establishing the requirements) but also on the created side of the process (the resulting system interacting with the stakeholders). Taking these factors into account, the research problem statement for this thesis can be formulated as follows:

> *"It is difficult to identify the correct project requirements within a complex socio-technical environment."*

1.3 Research methodology

1.3.1 Introduction

Adams *et al.* (2007:19) defined a research activity as "*… a diligent search, studious enquiry, investigation or experimentation aimed at the discovery of new facts and findings*". The purpose of research is to enhance an organisation's knowledge and understanding of known phenomena, less-known phenomena and unknown phenomena, and to increase its ability to meet the demands of the future (Adams *et al.*, 2007; Vaishnavi and Kuechler, 2015). In order to achieve this purpose, a research study needs a solid design before collecting and analysing data. The purpose of this research design is to ensure that any experimental evidence obtained during the research process serves its purpose by making the arguments presented more robust (Strang, 2015). Research can further be defined as a process of systematic inquiry that is conducted according to a theoretical framework. The purpose of the theoretical framework is to provide a distinction between research and other similar activities such as evaluation (Gregg *et al.*, 2001).

The objective of research conducted in the natural, business, management, and social sciences paradigm is to explain and predict the behaviour of specific observed phenomena and to find new truths or proofs as in, "*Why do things work in the way that they do?*" (Adams *et al.*, 2007; Geerts, 2011; March and Smith, 1995; Strang, 2015; Venable, 2006).

This type of research is fundamentally reactive as it tries to explain an event that has already occurred. Furthermore, it tends to follow the more conventional research approach of defining the problem, doing a literature review, stating a hypothesis, collecting data via some form of experimentation and analysis, documenting results, and coming to a conclusion (Geerts, 2011; Hevner, March, Park and Ram, 2004). The goal of this type of research is to identify and codify emergent properties and to discover and formulate laws or theories that explain the observed organisational and human behaviour (Hevner *et al.*, 2004).

The project execution domain, including the system engineering domain and within it, the requirements engineering domain, is a complex socio-technical system. Such a socio-technical system is created in which two jointly independent systems – the social and the technical – interact in a correlative way to produce a single outcome (Bostrom and Heinen, 1977a; Emery and Trist, 1960; Heydari and Pennock, 2018; Whitworth, 2012). Within this socio-technical system, the technical part is the result of the various processes, tasks and technologies involved. The social part relates to the different attributes of the various stakeholders involved in the process, the relationship between these stakeholders, the reward system present in the organisation, and the reporting and authority structure present (Bostrom and Heinen, 1977b).

This socio-technical system can also be viewed as a practical system in which naturally occurring objects, as well as human-made objects, are present and in which practical problems occur. Such practical problems can be seen as an unwanted or undesired state of affairs in which there is a gap between the current state and the desired state, as is perceived by the social element involved (Johannesson and Perjons, 2014). Many of these practical problems can also be considered to be wicked problems that can be difficult or near impossible to solve due to incomplete information, contradictory or changing requirements, and the complex interaction between the different elements present in the problem situation (Johannesson *et al.*, 2014).

1.3.2 Researching within a complex socio-technical system

When working with a complex system, an intuitive approach for a researcher is to try and break the problem into smaller and more manageable parts. This method is referred to as a reductionist approach. While this approach may provide suitable results when performing research in the natural, business or social sciences, it does not always yield the desired results when applied to complex socio-technical systems. One of the reasons identified for this failure is that complex or wicked systems, including complex socio-technical systems, contain many interconnected parts with the resulting relationship between the interconnected parts being more significant than the individual parts themselves (Arnold and Wade, 2017; Jackson, 2002).

One problem facing a researcher who needs to perform research within a complex socio-technical system is that there is no one single or unique way of defining complex or wicked problems, but many depending on the viewpoint of the researcher. Therefore, there are also no clear criteria that can be used to define when a problem has been solved, or a specific research objective has been reached. Any added effort can only improve the situation. When attempting to solve or improve such complex or wicked problems by using traditional research processes, as was referred to previously, it is often found that these research approaches do not always yield the desired results (Bostrom *et al.*, 1977a; Johannesson *et al.*, 2014). These shortcomings can be attributed to the sometimes-unpredictable interaction of the technical environment with the social nature of the problem domain, and the related inability to define and execute repeatable experiments.

An alternative approach is to view the stated research problem, firstly, as an investigation activity and, secondly, as an improvement activity. In both the investigation and the improvement activities, the first step is to establish a current baseline within a specific problem scenario, identify areas of improvement, make changes, and apply these improvements to the problem scenario. Once these improvement changes have been made, a new baseline can be determined, which is compared with the original baseline to see if any improvements have been realised.

Several areas of risk can be identified with such an approach. The first risk is that when dealing with a complex socio-technical system, making changes within such a system can have several detrimental effects, including increasing resistance to change and conflict within the stakeholder group. While making a once-off change can probably be managed, making some changes or tweaks over an extended period is bound to fail.

A second risk that must be considered is that within the complex socio-technical system, it is often found (and will also be demonstrated later in the thesis) that there are various mechanisms at play that influence one another either directly or indirectly. Tweaking these mechanisms within a live system poses a high risk of failure.

This thesis proposes the design of a requirements engineering research tool (also referred to as a research tool) using a design science research methodology. The research tool can be used to investigate the complex socio-technical system to determine a current performance baseline. Specific improvements can then be designed and evaluated on a simulation level. Only when the researcher or requirements engineering practitioner feels comfortable with the predicted improvement, can the improvement be implemented in the 'live' system or environment. This concept is shown in Figure 3.

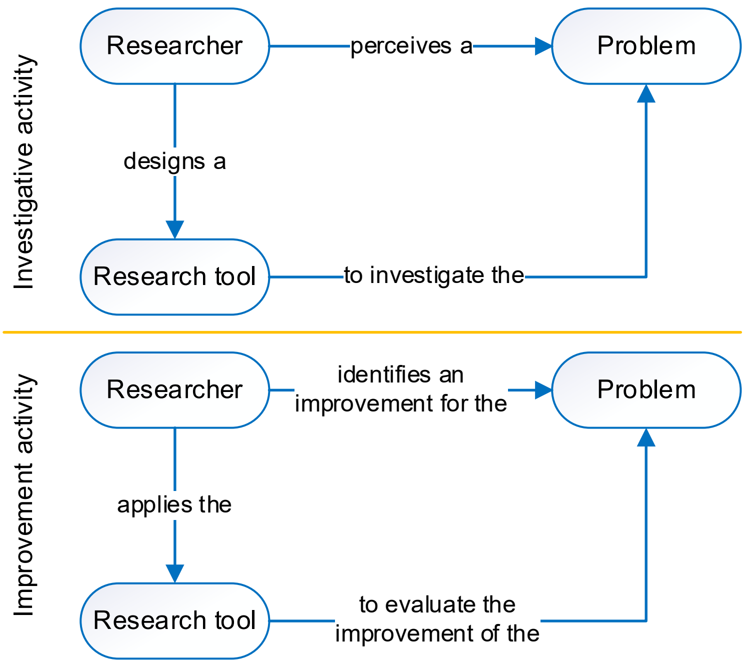


Figure 3: Application of design science research in addressing a complex socio-technical problem situation [own contribution]

This approach to problem-solving or problem-improving research using a design science research perspective will allow the focus of the research to shift from "*what was*" to "*how something should be to provide a specific utility at some point in the future*" by designing and applying an artefact to the problem context (Hevner *et al.*, 2004; Simon, 1996; Van_Aken, Chandrasekaran and Halman, 2016; Wieringa, 2014). In contrast to the reactive nature of alternative traditional research methods, this approach is a more proactive approach.

The research process produces Mode 2 knowledge while incorporating Mode 1 knowledge production as part of the empirical research cycle that is inherently part of the design science research methodology (Frost and Osterloh, 2003).

1.3.3 Design science research methodology

Design science and design science research as a science are concerned with the scientific study of design and the application of the design process in the systematic and scientific creation of knowledge about design and using design (Baskerville, 2008; Cash, 2018; Hevner and Chatterjee, 2010; Peffers, Tuunanen and Niehaves, 2018). Aljafari and Khazanchi (2013) described design science as the act of "*exploring while building*". Design science is a paradigmatic approach to research that is focused on solving a specific identified problem. This type of research creates an artefact and positions it within the problem setting (Baskerville, 2008; Wieringa, 2014). In doing so, the researcher answers questions relevant to human problems whereby the answers are obtained via the creation and use of innovative artefacts. These answers contribute new knowledge to the scientific body of knowledge. The artefact that is designed and the process followed in designing and implementing the artefact are fundamental to the understanding of the problem being solved (Baskerville, Baiyere, Gergor, Hevner and Rossi, 2018; Hevner *et al.*, 2010).

Design science research aims to create new and innovative artefacts that can be used to address significant and essential problems, demonstrate the capabilities of the artefact, and predict the future benefits and risks of these artefacts (Baiyere, Hevner, Rossi, Gregor and Baskerville, 2015; Baskerville, Pries-Heje and Venable, 2009; Hevner *et al.*, 2004; Vaishnavi and Kuechler, 2004).

The roots of design science and design science research can be traced back to the work of Simon (1996). He identified the concept of focusing research on the creation of artefacts from the viewpoint of "*how things ought to be to attain goals, and to function*" (Simon, 1996).

Wieringa (2014) introduced the concept of the problem context as part of the definition of design science. In this definition, design science is defined as the design of artefacts and the investigation of these artefacts within the problem context that they were explicitly designed for. The objective of the artefact is to improve something within the problem context.

The artefacts constructed as part of the research process, as well as the problem context, can take several different forms. Table 1 gives some examples.

Table 1: Examples of design science artefact and context elements [(March *et al.*, 1995; Wieringa, 2014)*]*

Artefacts	Problem context elements
Software, hardware, components and systems	People
Organisations	Values, desires, fears

Artefacts	Problem context elements
Constructs	Goals
Business processes	Norms, budgets
Methods	Software
Techniques	Hardware
Concepts	Services, methods
Models	Conceptual structures
Algorithms	Techniques

The artefacts and context elements shown in Table 1 are but some examples. The most significant distinguishing factor between artefacts and context elements is that individual elements such as people, values, and fears, among others, cannot be designed and thus cannot be artefacts (Wieringa, 2014).

Different design science artefacts have two common characteristics, namely, (a) relevance, and (b) novelty (Geerts, 2011). These shared characteristics are used to distinguish design science research from conventional design and to ensure that the artefacts are relevant to solving essential or current problems (Geerts, 2011). Hevner *et al.* (2004) suggested that a design science research problem should either focus on unsolved problems that are unique or find a better solution to an already solved problem. Design science should be distinguished from conventional design or “best practices” type of design.

Another critical aspect to consider in design science research is that it is not the artefact itself that solves the problem, but rather the interaction between the artefact and the problem context. Applying the same artefact to a different problem context can potentially yield a vastly different result (Wieringa, 2014).

The outcome of design science research can be evaluated in terms of new and improved theories and methods that find their way back to routine design activities and best practices design guidelines (Hevner *et al.*, 2004). In practice, it is found that the types of problems solved via design science research tend to be more of a socio-technical nature (Baskerville *et al.*, 2009).

Various researchers in the field of design science research, including Hevner *et al.* (2004); and Hevner (2007); and Peffers *et al.* (2007); and Wieringa (2014); and Iivari (2015); and Vaishnavi and Kuechler (2015); and Venable, Pries-Heje and Baskerville (2016) have published articles and books as to the aspects that should be included in a design science research methodology.

Hevner (2007) identified three cycles that form a core part of the design science research cycle. These cycles are: (a) the relevance cycle; (b) the design cycle; and (c) the rigour cycle, which are shown in Figure 4. In this model, the relevance cycle triggers the research process with an environment that not only identifies the requirements for the research but also defines the acceptance criteria for the evaluation of the results of the research process. The output of the research process must then again be returned to the environment from which it originated to be studied and evaluated.

The design cycle is central to the design science research project. Within the design cycle, the research activity iterates between the design and realisation of an artefact, its evaluation, and the resulting feedback to refine the design (Hevner, 2007; Simon, 1996).

The purpose of the rigour cycle is to identify past knowledge relevant to the research project to ensure its innovation. It is the researchers' responsibility to ensure that the designs produced are research contributions and not conventional designs based on the application of established processes (Hevner *et al.*, 2004). The research rigour is dependent on the skill of the researcher to select and apply the appropriate theories and methods for constructing and evaluating the artefact (Hevner, 2007).

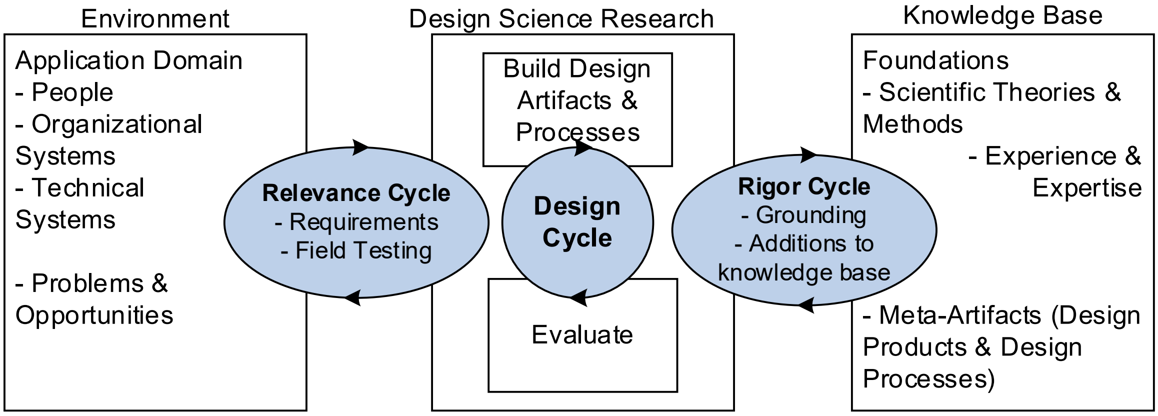


Figure 4: Design science research cycles [redrawn from Hevner (2007)]

Peffers *et al.* (2007) included the following elements in their design science research methodology: (a) problem identification; (b) objectives of the solution; (c) design and development; (d) demonstration; (e) evaluation; and (f) communication. Their view of the research methodology can be seen to approach a process model structure as shown in Figure 5.

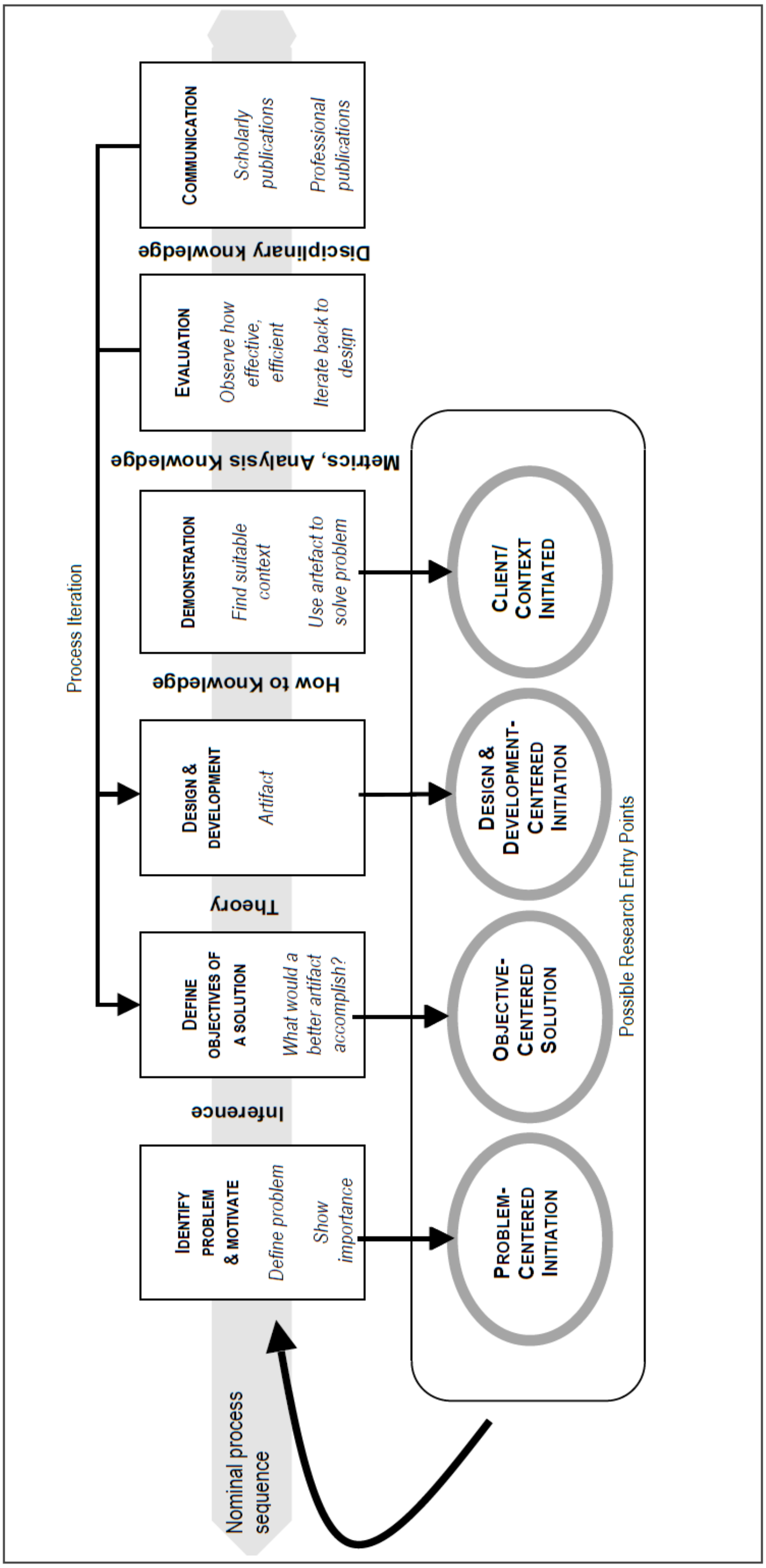


Figure 5: A process model for a design science research methodology according to Peffers et al. (2007)

Wieringa's (2014) approach to a design science research methodology is to define a design cycle and an empirical research cycle. The design science research process iterates between these activities of designing an artefact to fulfil the desired utility or need, and the empirical investigation of this designed artefact within the problem context it was designed for (Wieringa, 2014). This design science research methodology approach is shown conceptually in Figure 6.

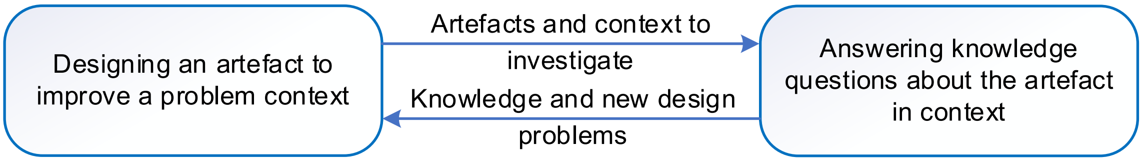


Figure 6: Design science research iterates between designing and answering knowledge questions [redrawn from Wieringa (2014)]

This approach results in a two-cycle approach that splits the design science research methodology into two parts functioning under the umbrella of the problem context. The first part is the design cycle that is used as a rational problem-solving process. During its first iteration, this design cycle addresses the problem investigation, treatment design, treatment validation, and treatment implementation activities. Any subsequent iterations address the evaluation and improvement of the implementation. The design cycle is shown in Figure 7.

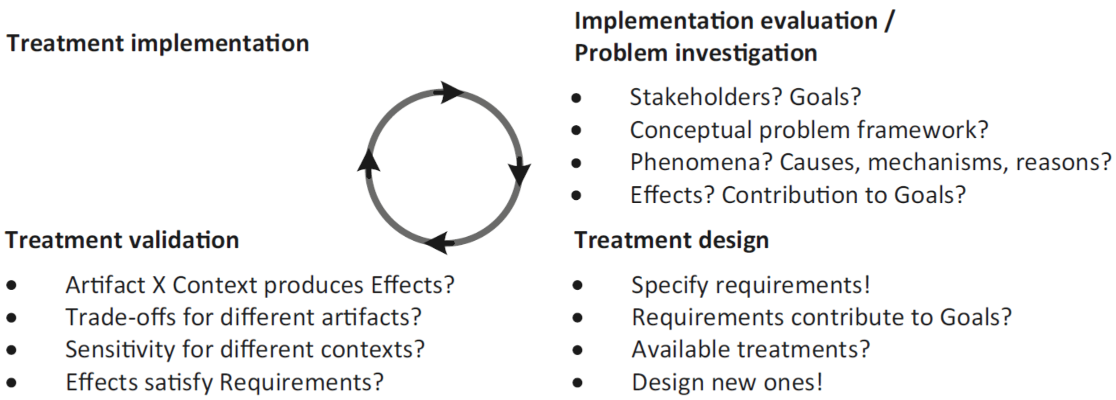


Figure 7: Design cycle according to Wieringa (2014)

The second part of the research methodology is the empirical research cycle that Wieringa defined as a rational way of answering scientific knowledge questions (Wieringa, 2014). This methodology addresses the research problem analysis, research and inference design, validation, and research execution. The empirical research cycle is shown in Figure 8.

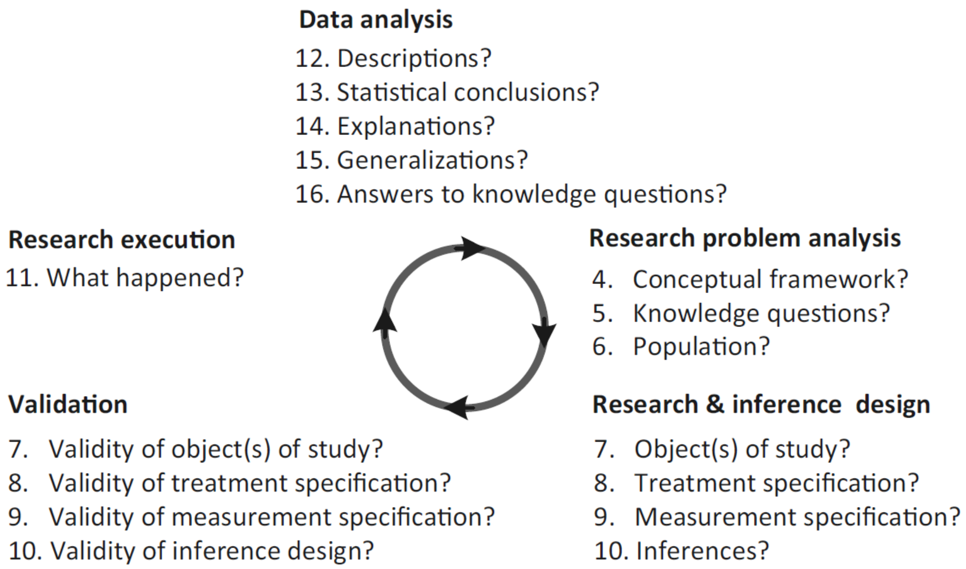


Figure 8: Empirical research cycle according to Wieringa (2014)

Vaishnavi *et al.* (2015) proposed a fourth alternative view of a design science research process model, which is shown in Figure 9. In their model, the process moves from (a) an awareness of the problem (or proposal); (b) a suggestion (or a tentative design); (c) the development of the artefact; (d) the evaluation of the artefact; and (e) the conclusion that feeds back in the form of design science knowledge to the awareness of the problem.

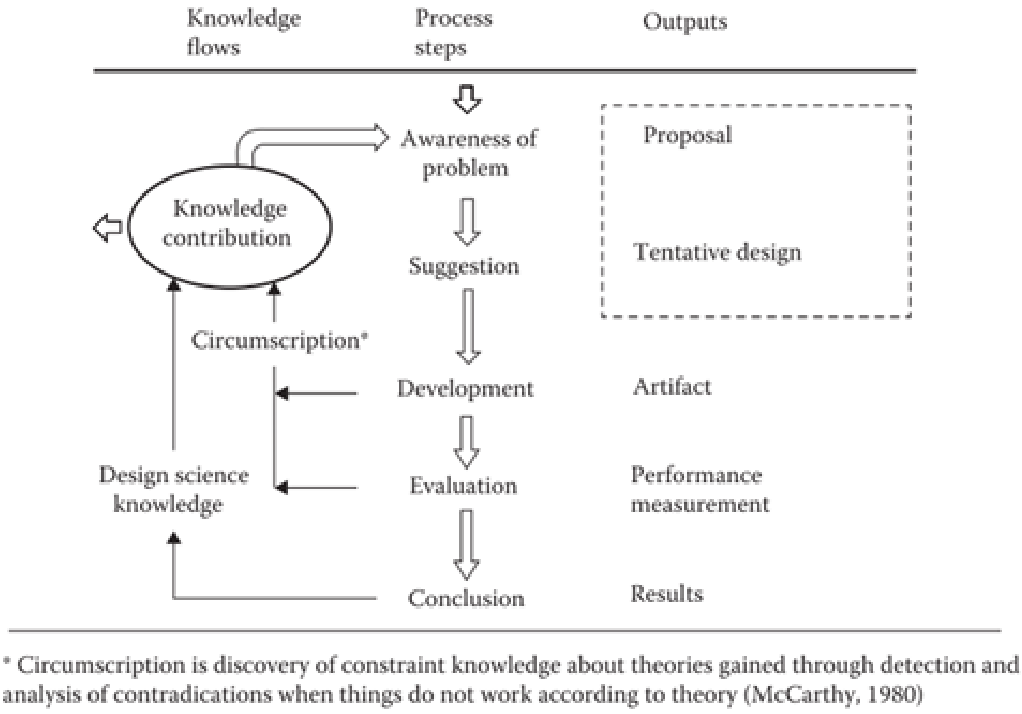


Figure 9: Vaishnavi *et al.* (2015) *design science research process model*

When comparing the various methodologies presented, three areas of commonality can be identified between the original three-cycle model proposed by Hevner *et al.* (2004), and the subsequent work done by Peffers *et al.* (2007), Wieringa (2014) and Vaishnavi *et al.* (2015).

The first common area relates to the relevance of the research, which Hevner *et al.* (2004) addressed with his relevance cycle. Peffers *et al.* (2007) incorporated this concept in their "identify problem and motivate" heading, which Wieringa (2014) referred to "as the acquisition and validity of the objects-of-study" in his methodology. Vaishnavi *et al.* (2015) addressed the topic under the heading of "awareness of the problem".

The second common area relates to the research rigour cycle identified by Hevner *et al.* (2004). Wieringa (2014) addressed this directly by specifying the use of empirical research methods.

The third common element is the design cycle identified by Hevner *et al.* (2004). Wieringa (2014) addressed this by a reasonably detailed design cycle that covers the main activities required to design an artefact. Peffers *et al.* (2007) and Vaishnavi *et al.* (2015) addressed the design activities as part of the topics of design, development and evaluation.

Of the various design science research methodologies proposed, the methodology proposed by Wieringa (2014) is the most comprehensive except for the detail activities and processes identified for the design cycle. An alternative approach for enhancing the design cycle was presented by Scribante, Pretorius and Benade (2018a) that expanded the design cycle proposed by Wieringa (2014) to include elements of the technical process as detailed in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

Table 2 compares the design cycle as described by Wieringa (2014) with the design process as defined in the Systems Engineering Handbook (INCOSE, 2015) and the expanded design cycle that includes additional steps to ensure the completeness of the design and implementation of the artefact being realised as was proposed in Scribante, Pretorius and Benade (2018a).

Table 2: Comparison of the engineering and design cycle of Wieringa (2014) *and the INCOSE Systems Engineering Handbook* (INCOSE, 2015) *and* Scribante *et al.* (2018a)

Engineering and design cycle as per Wieringa (2014)	Engineering and design cycle as per INCOSE (2015)	Scribante et al. (2018a)
Research context	Business or mission analysis	Research context definition phase
Problem investigation	Stakeholder need and requirements definition	Research problem analysis phase
Treatment design	System requirements definition	Research design process
Treatment implementation	Architecture definition	Research instrument implementation process
	Design definition	
	System analysis	
	Implementation	
	Integration	
	Verification	Verification process
	Transitioning	Transitioning process
Treatment validation	Validation	Validation process
		Research execution and data analysis process
		Communication and dissemination

1.4 Research design

The purpose of this section is to define the research process as it will be applied in this thesis. Design science is the design and investigation of artefacts interacting with a problem context in order to solve a specific problem within the context (Wieringa, 2014). During this problem-solving process, the researcher iterates between the two main activities of designing an artefact, which aims to improve a concern of the identified stakeholders, and investigating the performance of the artefact in the context of using empirical research methods (Wieringa, 2014).

The research design that will be used in this thesis is based on the core design science research methodology as proposed by Wieringa (2014). The design cycle of the research methodology will, however, apply the expanded design cycle as proposed by Scribante *et al.* (2018a). Table 3 provides a mapping of the different research phases as proposed by Scribante *et al.* (2018a) to the different chapters in this thesis.

Table 3: Comparison of the different research phases as proposed by Scribante et al. *(2018a) and the chapters in this thesis*

Research phases as proposed by Scribante *et al.* (2018a)	Chapters in this thesis
Research context definition phase	Chapter 2: Research problem identification phase
Research problem analysis phase	Chapter 3: Research problem definition phase
Research design process	Chapter 4: Requirements engineering research tool specification phase
Research instrument implementation process	Chapter 5: Requirements engineering research tool design, implementation and verification phase
Verification process	
Transitioning process	
Validation process	Chapter 6: Requirement engineering research tool validation phase
Research execution and data analysis process	Chapter 7: Conclusion
Communication and dissemination	

Figure 10 shows the design science research process that will be used in this thesis as a functional flow block diagram (FFBD).

1.4.1 Research problem identification phase

1.4.1.1 Overview

The research problem is identified by defining the research problem context in terms of a problem or opportunity that must be addressed, including the problem space within which the problem or opportunity resides as well as the object-of-study. The research context definition can be equated to the business or mission analysis process as is defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015). The activities in the sub-sections that follow form part of the research problem identification phase.

1.4.1.2 Research problem identification

The identification and selection of the research problem that is to be studied using a design science research methodology can be evaluated against the criteria of: (a) the relevance of the problem in terms of current or relevant problems; (b) the novelty of the problem in terms of its unique nature; and (c) the significance of the problem (Baskerville *et al.*, 2009; Geerts, 2011; Hevner *et al.*, 2004; Vaishnavi *et al.*, 2004).

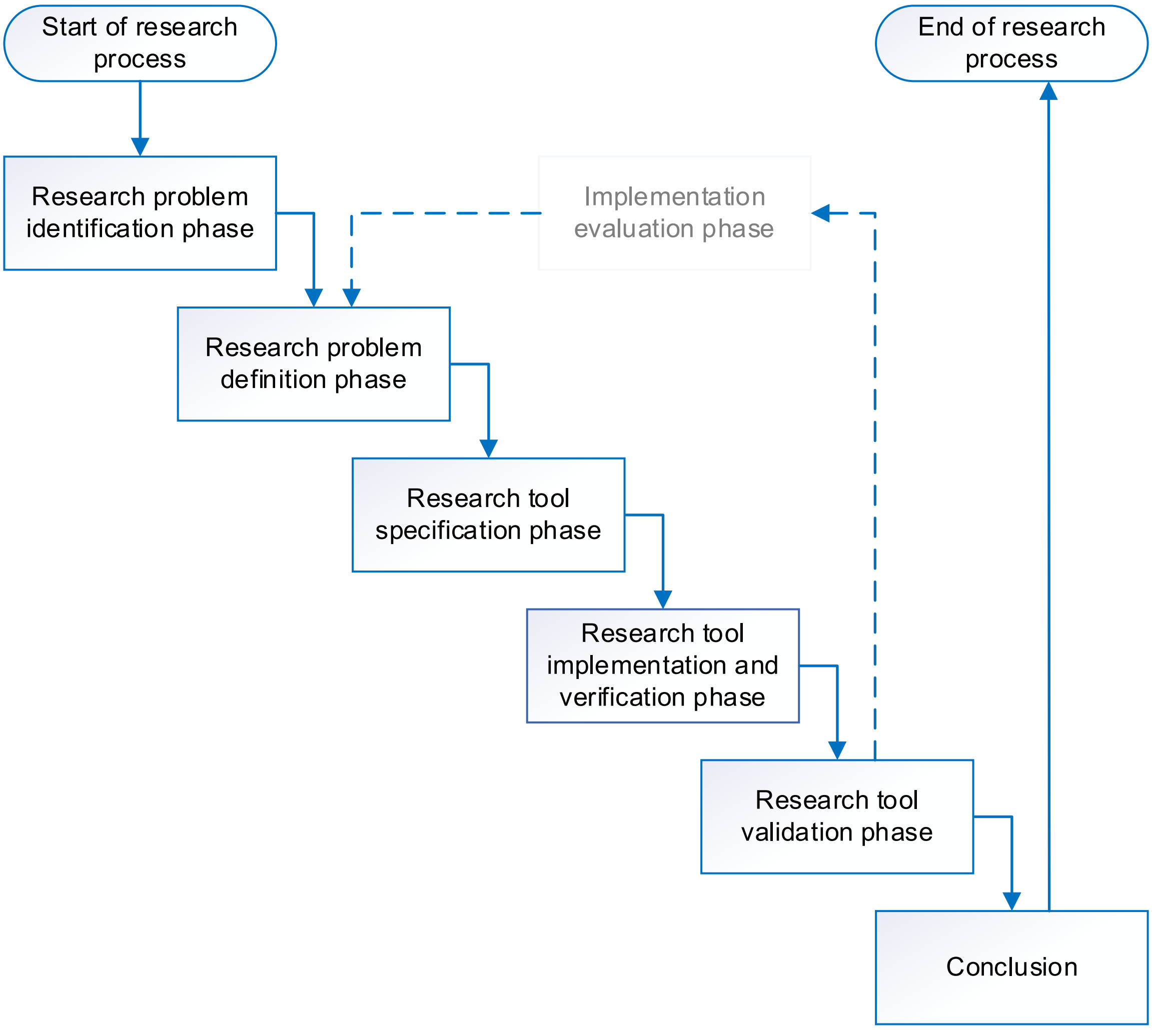


Figure 10: Research FFBD

1.4.1.3 Research goals

The research goals associated with the research problem can be specified in terms of design goals, knowledge goals, prediction goals and improvement goals, or a combination of these goals.

The knowledge goals can further be defined in terms of implementation evaluation goals, problem investigation goals, a survey of existing solutions, or new technology validation goals (Wieringa, 2014). The relationship between the different types of goals is shown schematically in Figure 11. At some point in the research process, research instruments or research tools may be required to achieve some of the identified knowledge goals. The goals associated with these research tools are defined as instrument design goals and are highlighted in the bottom part of Figure 11.

The research goals are a combination of the goals of the external stakeholders and the goals of the researcher. The goals of the researcher may include aspects such as curiosity or a desire to improve something within society. The external stakeholder may have similar goals, but these will typically include more concrete goals to arrive at an actual solution to a problem situation. This will especially be true if the external stakeholder is also the sponsor of the research project (Wieringa, 2014).

Typical research methods that the researcher can employ to determine the research goals include observational techniques, self-reflection, interviews, and focus groups. The research goals for this study are presented and discussed in paragraph 2.3.1.

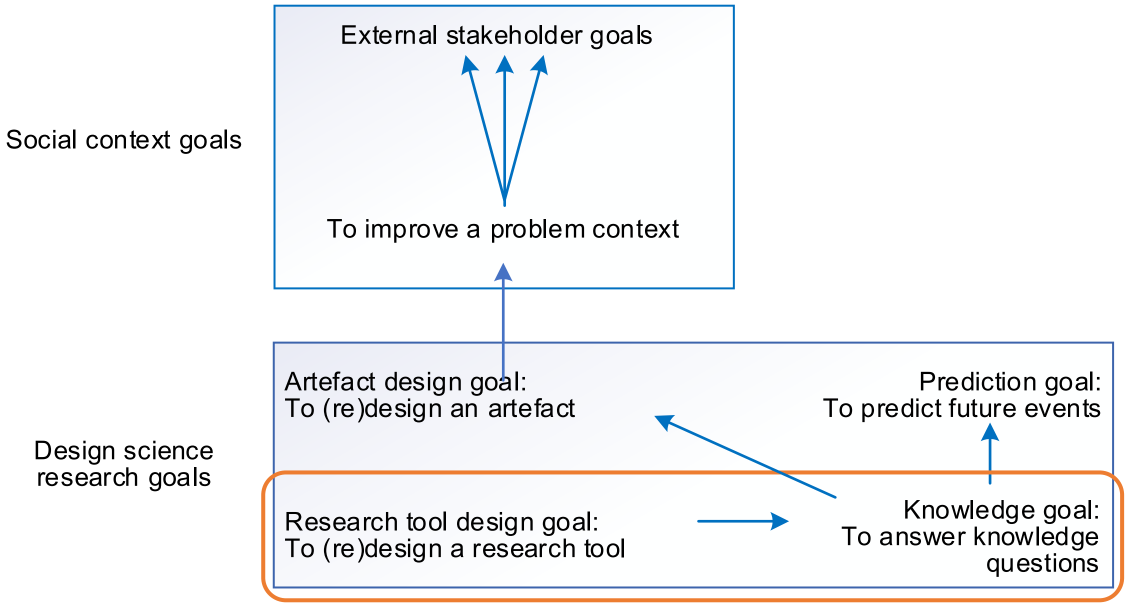


Figure 11: Relationship between the various knowledge and design goals [redrawn and adapted from Wieringa (2014)]

1.4.1.4 Current knowledge

The current knowledge available on the research problem or subject must be established. The function of the current knowledge is to serve as background support for the description of the observed phenomena, which helps to define the research problem. This knowledge can be obtained from a variety of sources, including professional literature such as published scientific, technical and trade literature; professional literature; or subject matter experts (SMEs) in the field (INCOSE, 2015; Peffers, Tuunanen, Gengler, Rossi, Hui, Virtanen and Bragge, 2006; Wieringa, 2014).

The search for and the description of the current knowledge provide a point of departure for the researcher. In some instances, it may turn out that after the current knowledge available has been established, enough data already exists that either answers the knowledge goals or may reduce the scope of the study. Typical research methods that the researcher can employ to establish the current knowledge include literature surveys and expert interviews. The current knowledge can be equated to the literature survey found in the classical research methodology. The current knowledge for this study is presented in paragraph 2.3.2.

1.4.1.5 Research conceptual framework

The conceptual research framework provides the basis for the reasoning why the research topic matters (is relevant) and why the research process is appropriate and rigorous. To support the arguments for relevance and rigour, the research conceptual framework should (a) map the research questions as an extension of the problem statement; (b) trace the research design through to the research goals, research questions and context; (c) demonstrate that the data collected supports the analysis of the research questions; and (d) that the inference and analytical process selected supports the answering of the research questions (Ravitch and Riggan, 2017). These items are summarised in Figure 12.

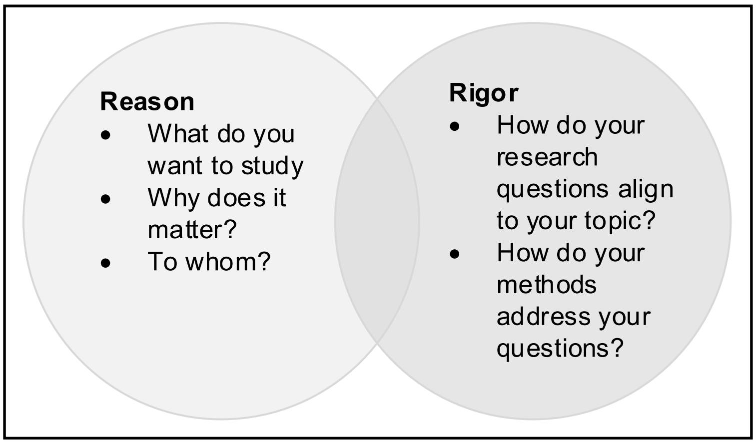


Figure 12: Relevance (reason) and rigour of the research process [redrawn from Ravitch *et al.* (2017)*]*

Gregor and Jones (2007) provide a different viewpoint of the relevance and reason argument in terms of eight elements as shown in Table 4.

Table 4: Eight components of design theory framework (Gregor *et al.*, 2007)

No.	Component	Description
1	Purpose and scope (*causa finalis*)	"What the system is for", the set of meta-requirements or goals that specifies the type of artefact to which the theory applies, and in conjunction also defines the scope, or boundaries, of the theory.
2	Constructs (*causa materialis*)	Representations of the entities of interest in the theory.
3	The principle of form and function (*causa formalis*)	The abstract blueprint or architecture that describes an artefact, either product or method/intervention.
4	Artefact mutability	The changes in the state of the artefact anticipated in theory, that is, what degree of artefact change is encompassed by the theory.
5	Testable propositions	Truth statements about the design theory.
6	Justificatory knowledge	The underlying knowledge or theory from the natural, social or design sciences that gives a basis and explanation for the design (kernel theories).
7	Principles of implementation (the *causa efficiens*)	A description of processes for implementing the theory (either product or method) in specific contexts.

No.	Component	Description
8	Expository instantiation	A physical implementation of the artefact that can assist in representing the theory – both as an expository device and for purposes of testing.

The detail of the conceptual frameworks is presented in paragraph 2.3.3.

1.4.1.6 The object-of-study

The object-of-study in a design science research project is an artefact in context. It can be defined as the entity in which the observed phenomena occur from which measurements are to be made (Wieringa, 2014). The artefact is purposefully designed to interact with the problem context to improve an element within that context (Wieringa, 2014). It is neither the artefact nor the context that provides research output, but rather the interaction between the artefact and the context. This interaction is shown in Figure 13.

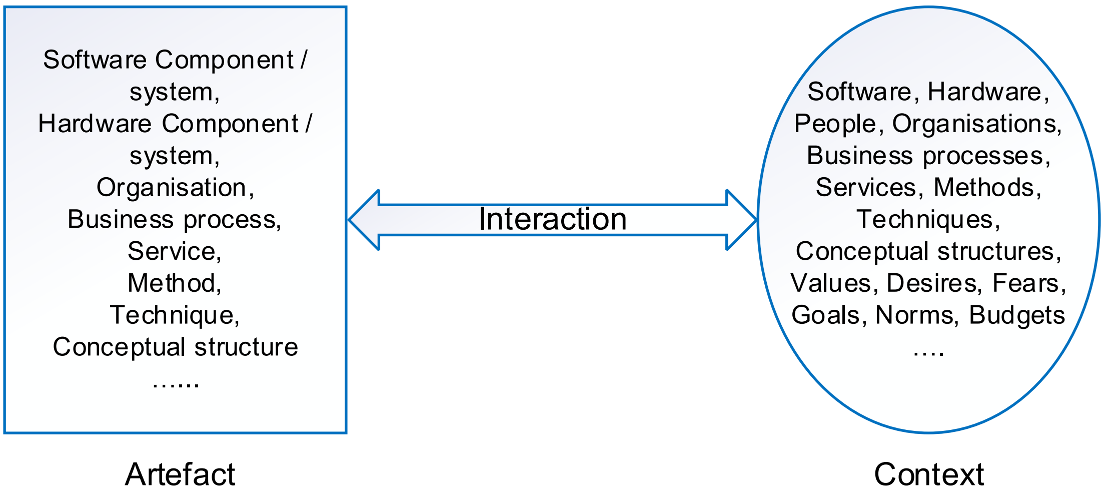


Figure 13: The object-of-study in design science research – the artefact in context [redrawn from Wieringa (2014)*]*

The artefact can consist of various items or combination of items such as software or hardware components, organisations, business processes, and methods. The context can comprise similar items such as hardware, software, business processes, and methods; the main difference between the artefact and the context is that the context includes people and their characteristics such as values, desires, fears, and norms (Wieringa, 2014).

When designing a research tool, such as is the focus in this thesis, the object-of-study changes. Instead of just studying the artefact interacting with the context, a research tool artefact must be designed that investigates a new problem context that now consists of the original problem artefact interacting with the original problem context (Scribante *et al.*, 2018a). The research tool artefact must thus be able to investigate the new problem context without influencing its operation. This will initially allow the researcher to make observations of the new problem context and establish a performance baseline. This performance baseline can be re-evaluated based on specific changes that were made to the problem context (e.g. such as an improved process). This new object-of-study is shown in Figure 14.

The object-of-study is the part of the world that the researcher interfaces with in order to learn something based on a sample taken from the population of the problem context or a model of the population elements (Wieringa, 2014). This is conceptually shown in Figure 15. Identifying the population and selecting the sample object-of-study to be studied depend on the nature of the research to be done. When doing case-based research, the aim is to study individual objects with the objective being to generalise to similar objects. In sample-based research, the aim is to study samples with the objective being to generalise to the whole population from which the samples were taken (Wieringa, 2014).

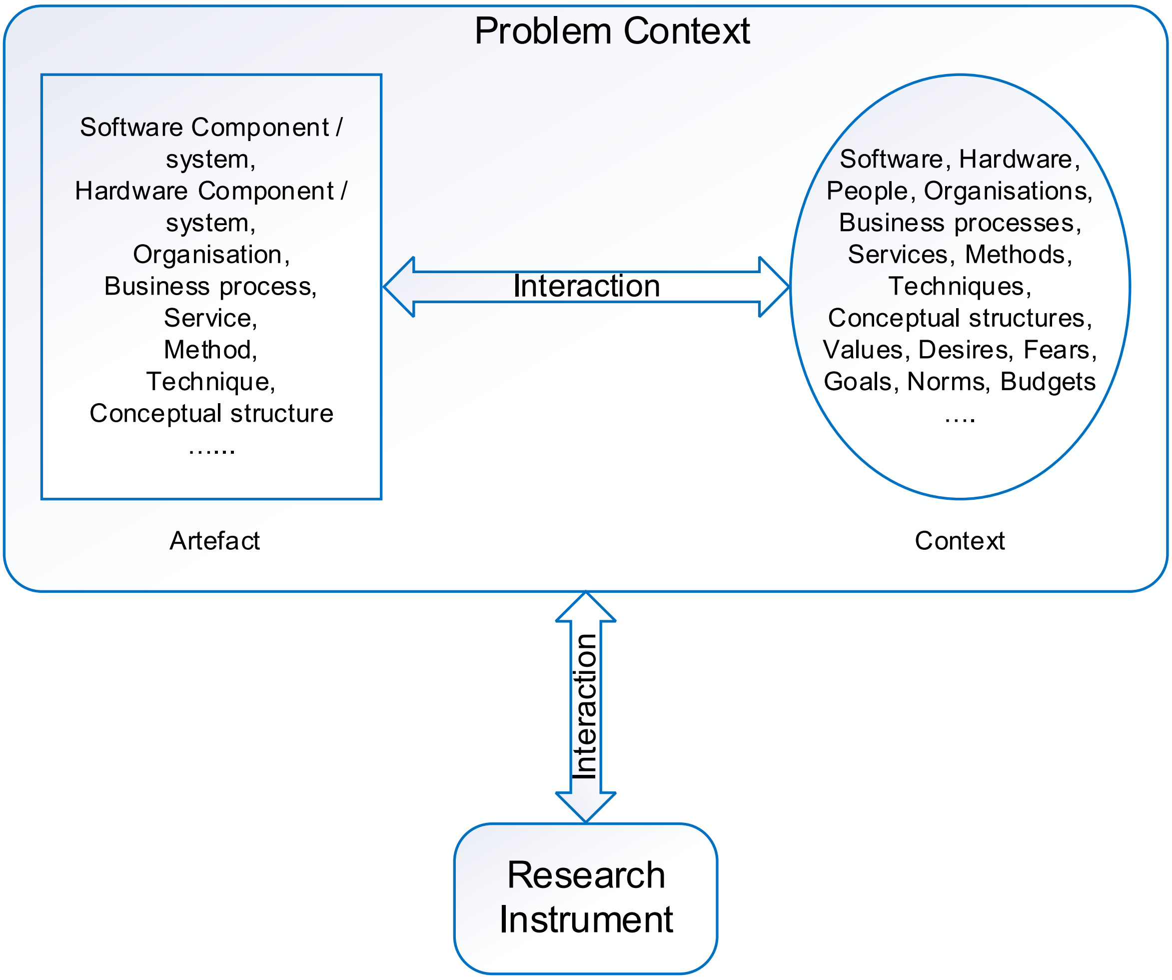


Figure 14: The object-of-study for the design of a research tool artefact (Scribante *et al.*, 2018a)

As part of the selection process of the object-of-study, the following aspects must be borne in mind (Wieringa, 2014):

- The object-of-study must be a valid sample from the population of elements. This is illustrated in Figure 15.
- The object-of-study must support the specific inference that is selected to analyse the data obtained.
- Other researchers performing similar sampling should come to a similar result.
- The ethical considerations that may be applicable during the sampling process must be considered.

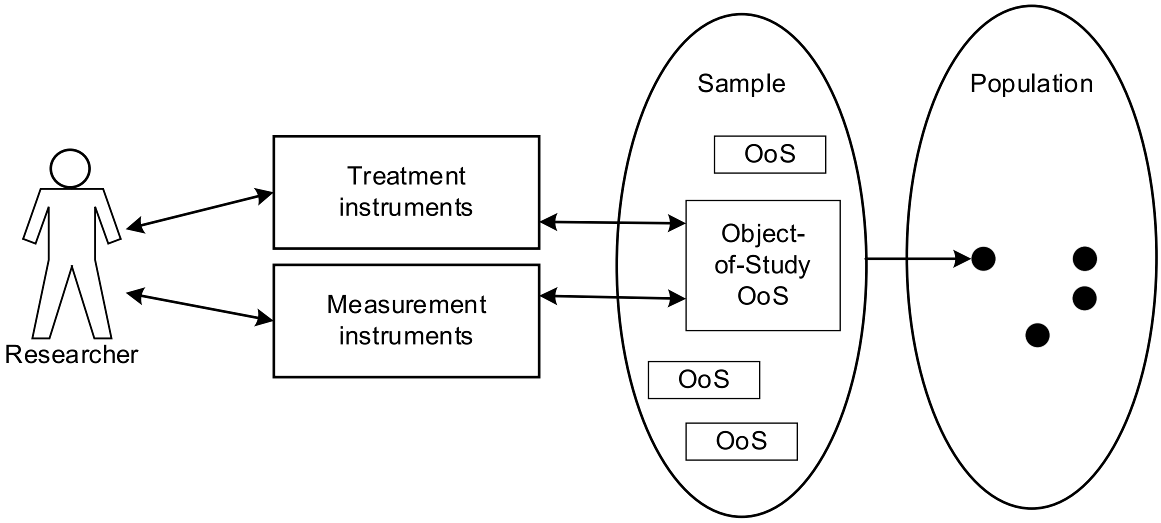


Figure 15: Relationship between the researcher, the object-of-study sample and the population [redrawn from Wieringa (2014)]

The object-of-study for this research is presented in paragraph 2.4.3.

1.4.1.7 Identify alternative solution classes

Alternative solution classes need to be identified. The function of identifying these alternative solution classes is to ensure that the researcher is not stuck in a rut by only using one tool no matter what the problem is. In this thesis, the approach presented in the book *Creative Problem-solving: Total Systems Intervention* by Flood and Jackson (1991) will be used to identify alternative solution classes.

1.4.2 Research problem definition phase

Once the research problem has been identified, the research problem must be analysed in detail. During this process, the objectives of the solution are inferred rationally from the identified research problem. These objectives can be either quantitative or qualitative.

The research problem definition can be equated to the stakeholder needs and requirements definition process as is defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

The following aspects must be considered in the context of defining the research problem:

1. The process of identifying the stakeholders as well as their goals, desires and conflicts. The purpose of this activity is to define the stakeholder requirements for the research that can provide the results needed in the defined environment. This activity can include the following:

   a. Identifying stakeholders. For a research project, the stakeholders can include the researchers performing the research, the research subjects, and the organisation requesting the research.
   b. Discovering and analysing the different needs, goals, desires and conflicts of each of the identified stakeholders.
   c. Establishing how (in) and why this project is different to them.
2. What are the phenomena? Why do they happen? (Causes, mechanisms, reasons.) What are their effects if nothing would be done about them? Do they contribute or detract from research goals?
3. The conceptual problem framework referred to paragraph 1.4.1.5 is expanded by defining the architectural or statistical structures in greater detail. These structures are defined by the causes and mechanism of the observed phenomena, and what models of the random variables can be identified (Hevner *et al.*, 2004; Vaishnavi *et al.*, 2015; Wieringa, 2014).
4. Knowledge and research questions must be formulated. The knowledge or research questions cover aspects such as what the observed phenomena are, what causes and effects could be identified, and the contribution of these phenomena to the stakeholder goals. Various types of questions can be asked such as open (exploratory), closed (hypothesis testing) or descriptive questions. Statistical questions can either be in the effect, satisfaction, trade-off, or sensitivity questions (Wieringa, 2014).
5. If statistical structures are present, the statistical population must be identified regarding the architecture of the elements of the population and how the population elements are alike and dissimilar to other elements.
6. Population – Population predicate? What is the architecture of the elements of the population? In which ways are all population elements alike and dissimilar to other elements? – Chance models of random variables: The assumptions that can be made concerning the distribution of variables.
7. The operational concept of how the research will be executed must be developed.
8. The identified needs must be transformed from their user-orientated view into requirements and confirmed with the stakeholders[1] to ensure they are expressed correctly. The following methods can be considered to support the process of identifying the correct requirements for the research tool artefact:
   a. Systems thinking – Hard systems thinking vs soft systems thinking – operation research, systems engineering or situational awareness (Jackson, 2003).

[1] The researcher is considered to be the stakeholder for the purpose of the discussion in the thesis.

   b. Organisational cybernetics.
   c. Complexity theory.
   d. Soft systems methodology (approach or viewpoint).
   e. Rich pictures (method).
   f. Total systems intervention (these are all systems thinking methodologies).
   g. Critical systems practice.
9. The identified requirements must be analysed to ensure that there is no conflicting, duplicated or possible missing requirements and that the requirements form a coherent set (INCOSE, 2015; Wieringa, 2014).
10. The research ethics must be established and included as part of the requirements set (Mouton, 2001).

1.4.3 Solution specification phase

The previous section identified the stakeholder requirements for the research tool artefact. In this section, the specification for the research tool artefact is established, including the research methodology that will be followed. The solution specification phase can be equated to the system requirements definition process and the architecture definition process as is defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

The following aspects must be considered:

1. Artefacts can constitute many different types of items and include among others, constructs, models, methods, instantiations algorithms, methods, notations, techniques, and even conceptual frameworks (Hevner *et al.*, 2004; Wieringa, 2014). The design activity includes determining the artefact's desired functionality and architecture.
2. Different alternatives for artefacts should be considered and evaluated before selecting one or more alternative architectures or designs that meet the requirements.
3. The design of the specific implementation of the artefact should be consistent with the architectural entities as defined in models and views of the system architecture (INCOSE, 2015).

An essential aspect that must be considered when designing a research tool is that the research tool artefact must support the specific research and inference design. In doing so, the object-of-study, sampling of the research population, and the specific measurements that will be performed must be defined. For the design of the inference process and method of the data collected, it must be decided whether descriptive inference, statistical inference, abductive inference, analogical inference, or a combination of these will be used to analyse the results (Wieringa, 2014).

1.4.4 Solution design, implementation and verification phase

The purpose of the solution implementation and verification phase is to define the architecture and the design of the artefact or research tool according to the specifications established previously. The solution implementation phase covers the design definition process, system analysis process, implementation processes, integration process, and the verification process as defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

The research tool artefact can consist of one or more building blocks or elements that will be integrated to create the realised system (product or service) that satisfies the identified requirements, architecture, and design. The artefact realised at the end of this phase can range anywhere from being a software component, hardware component, business process, service, method, a technique, to a simulation model (Refer Table 1).

Before the solution can be transferred to a research environment for validation, its basic operation must first be verified. The purpose of the verification process is to provide proof that a system or system element fulfils its specified requirements and characteristics (Haskins, 2016; INCOSE, 2015; NASA, 2017). Activities such as inspection, testing, demonstration and analysis can be used to verify the system. The primary purpose is to establish the effectiveness of the artefact to solve the problem by using experimentation, simulation, a case study, or other suitable activity as indicated in Table 5.

Table 5: Design evaluation methods (Hevner *et al.*, 2004)

Evaluation method	Specific implementation
Observational methods	*Case Study* – Study artefact in-depth within the environment
	Field Study – Monitor the use of the artefact in multiple instances

Evaluation method	Specific implementation
Analytical evaluation methods	*Statistical analysis* – Examine the structure of artefact for static qualities (e.g. complexity)
	Architecture analysis – Study fit of artefact in technical structure
	Dynamic analysis – Study artefact in use for dynamic qualities (e.g. performance)
Experimental	*Controlled experiment* – Study artefact in a controlled environment for qualities (e.g. usability)
	Simulation – Execute artefact with artificial data
Testing	*Functional (black box) testing* – Execute artefact interfaces to discover failures and identify defects
	Structural (white box) testing – Perform coverage testing of all execution paths in the artefact

1.4.5 Solution validation phase

The purpose of the solution validation process is to provide proof that the research tool artefact meets its intended purpose when used within the research context as identified previously (INCOSE, 2015; NASA, 2017). The solution validation phase covers the validation process as defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

This validation can be done by performing several iterations of the intended research and data analysis cycles. Based on the results obtained, a conclusion can be drawn on how well the artefact will support the actual research process. This activity will involve comparing the objectives of the solution with the actual observed results from the use of the artefact in the demonstration. In order to achieve this, knowledge of the relevant metrics and analysis techniques identified as part of the inference design will be required (Hevner *et al.*, 2004).

The research data obtained during the solution validation process has to be checked for consistency to identify and correct data transformations, missing values, and removal of outliers (Wieringa, 2014). Various questions can be asked during the solution validation phase. These questions can include the following (Wieringa, 2014): (a) Did the selected cases have the architecture that was planned during research design? (b) Did any unexpected events occur during the study? (c) What happened during the analytical induction (i.e. sampling) and did it support the original design?

The results generated must be analysed and explained using causal, architectural, or rational reasoning methods. The general validity of the results must furthermore be examined to determine if the methods used and the results obtained are transferable to similar cases or populations (Wieringa, 2014). In the end, the results obtained should answer the knowledge questions posed during the research design process and include a summary of the conclusions and the limitations of the conclusions (Wieringa, 2014).

The solution validation phase can potentially also be used to evaluate the performance of different research tool artefacts when more than one artefact has been validated. Furthermore, the solution validation phase can be used to evaluate the sensitivity of the research tool and to quantify the contribution of the research tool to the knowledge goals and improvement goals that were identified at the start of the study.

In the end, the purpose of the validation process is to evaluate the performance of the artefact in context, look at the trade-offs for the different types of artefacts when more than one have been validated, establish the sensitivity of the research tool, and quantify the contribution of the research tool to the knowledge goals and improvement goals identified at the start of the study.

The relationship among the various requirements stages, design stages and the verification and validation activities are shown in Figure 16.

1.4.6 Research communication or dissemination

The outcome of the overall research process is the knowledge that contributes to the understanding of a phenomenon (Vaishnavi *et al.*, 2015). The results of the research process must be shared and communicated to the broader academic and technical community to be added to the general knowledge base and to ensure rigour in the research process. The research can be shared by communicating the problem and its importance, the artefact produced, its utility and novelty of design, and its effectiveness in answering the knowledge questions (Hevner *et al.*, 2004). In terms of the research presented in this thesis, the contribution is in the form of the thesis itself and the scholarly articles presented at conferences and published in the proceedings. The articles are detailed in Table 7.

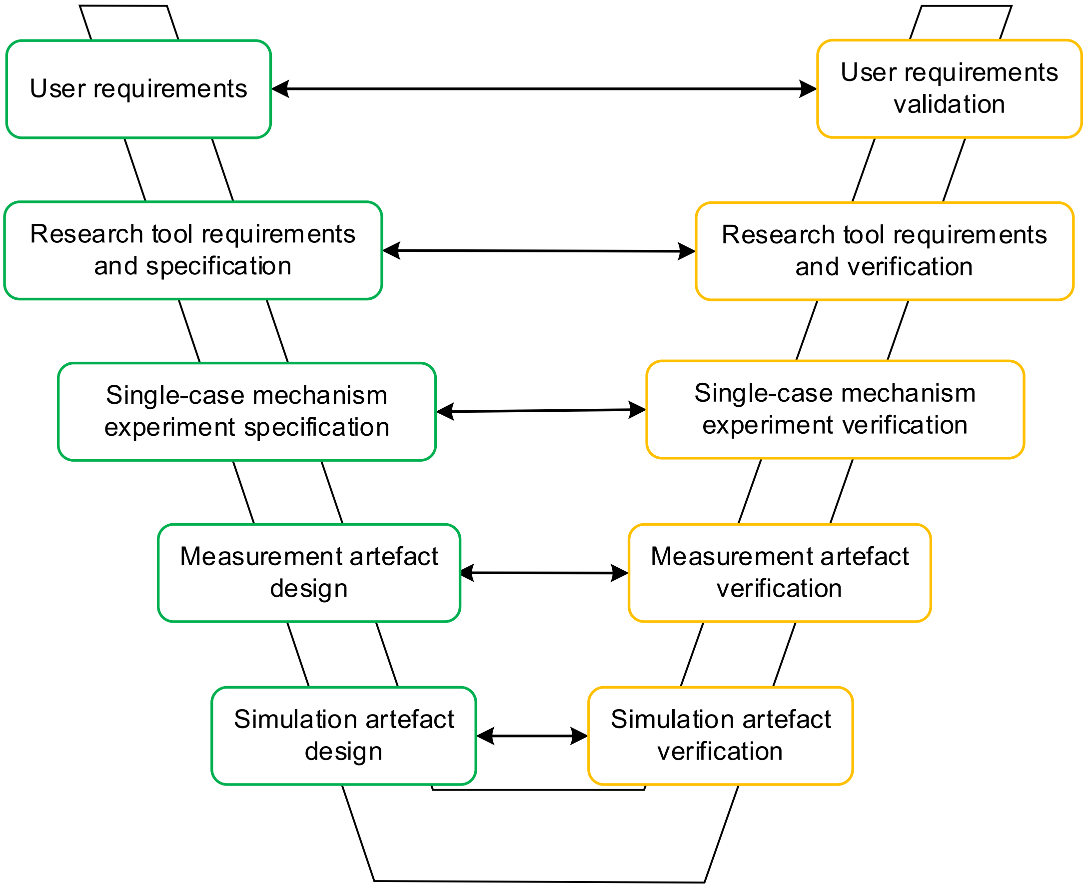


Figure 16: Relationship between the various requirements stages, design stages and the verification and validation activities (own contribution)

1.5 Design science research topology

Research is not an activity that can be done in isolation as the knowledge and insight gained from research must be shared with other researchers and stakeholders for it to contribute to the general body of knowledge. It is essential that the design of the investigative research tool artefact uses recognised ontologies, boundaries, guidelines, and deliverables. This will ensure that the overall research design approach, and the communication of results, is understood by other design science researchers specifically, as well as other academic researchers in general (Hevner, 2007; Rahi, 2017; Strang, 2015).

To support the research process and to make the information sharable, Strang (2015) defined a conceptual research model that addressed the research process as a four-layer, top-down topology. The four layers are shown in Figure 17.

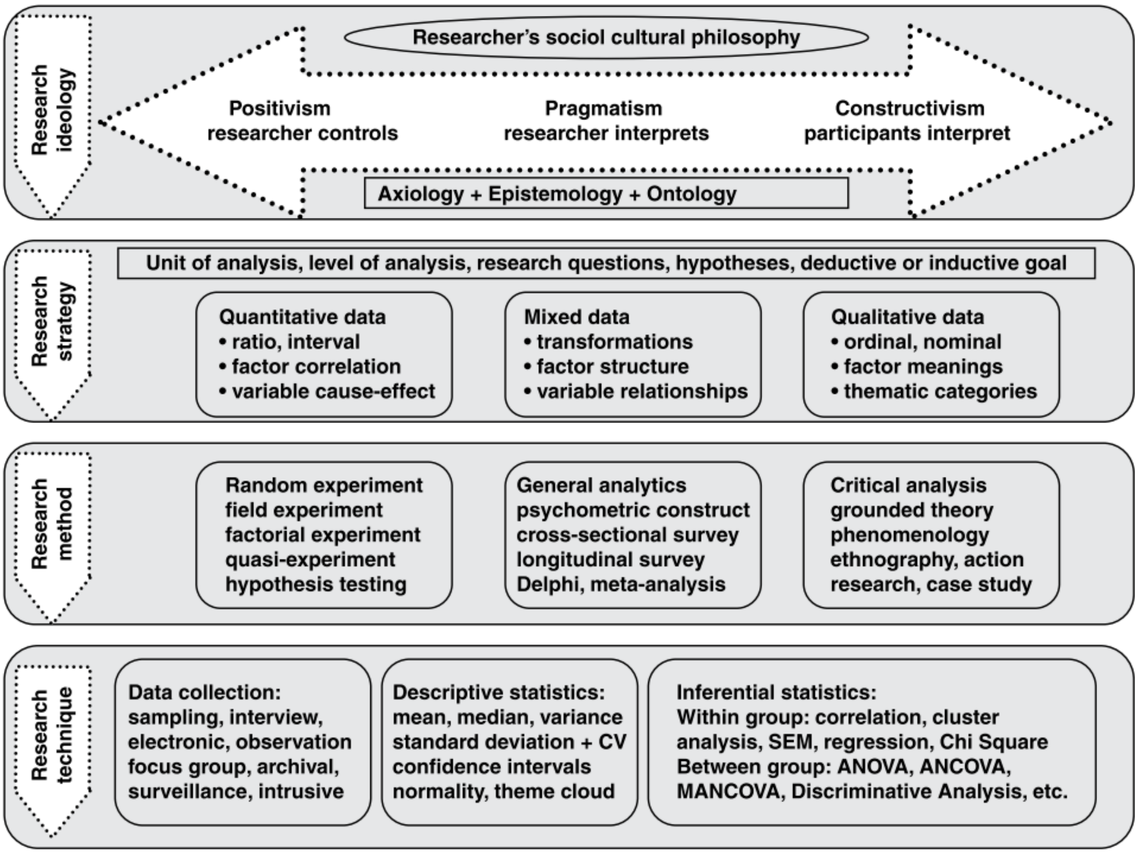


Figure 17: Four-layer, top-down research topology (Strang, 2015)

(SEM: Structural equation modelling, ANOVA: Analysis of variance, ANCOVA: Analysis of covariance, MANCOVA: Multivariate analysis of covariance)

Strang (2015) defines the elements of the research topology as following-:

a. The research ideology provides an indication of how the researcher thinks about knowledge claims that are made. The ideology is defined in terms of the research paradigm, which can range from explicit evidence derived from theories (positivism) to qualitative evaluations provided by the participants (constructivism), and the metaphysical assumptions in terms of the socio-cultural and philosophical attitude of the researcher. These categories define the axiology, epistemology and ontology used as reference basis during the research. The purpose of these categories is only used for relative comparison among researchers so that they may understand the terminology and priorities that played a role in defining the research (Gregg *et al.*, 2001; Maxwell, 2013; Strang, 2015; Vaishnavi *et al.*, 2015).
   i. The axiology can be defined as the theory of beliefs that the researcher subscribes to, and is described by the priority of values such as morals, religious influences and ethics (Strang, 2015).

ii. The epistemology can be defined as the theory of knowledge, which includes the specific terminology used in the field of study for communication between researchers (Strang, 2015).

b. The ontology can be defined as the theory of being and defines the degree to which the researcher believes something is real (Strang, 2015). The research strategy is defined by the unit of analysis, level of analysis, research questions, hypotheses, and deductive or inductive goals (Strang, 2015).
c. The research method is the way of conducting and implementing the research (Adams *et al.*, 2007). Different research methods include experiments in various forms, hypothesis testing, general analytics, surveys, critical analysis, grounded theory, action research, and case studies, among others. The selection of a specific research method usually depends on the research ideology being followed in the specific research exercise (Strang, 2015).
d. The specific research technique selected depends on the research ideology, research strategy and research method used. Research techniques can draw on a range of specific techniques for data collection, descriptive statistics and inferential statistics.

Iivari (2007) and Aljafari and Khazanchi (2013) and Vaishnavi and Kuechler (2015) expanded on the research topology definition of Strang (2015) by defining a new research ideology *design/design science* that stands at the same level as the positivist, pragmatist and constructivist/interpretivist ideologies. Within this *design/design science* ideology, they provided descriptions for the ontology, epistemology, methodology and axiology. The combined perspective is shown in Table 6. The examples shown in the table are not exhaustive but are only indicative of the type of the parameter.

Table 6: Philosophical assumption for the design/design science research ideology compared with the positivist and interpretivist ideologies (Aljafari *et al.*, 2013; Iivari, 2007; Strang, 2015; Vaishnavi *et al.*, 2015)

	Research Ideology		
Basic Belief	Positivist	Constructivist/ Interpretivist	Design/Design Science
Axiology (what is of value)	Truth: universal and beautiful; prediction.	Understanding: situated and description.	Control; problem-solving; progress (i.e., improvement); understanding the real world by creating artefacts.

	Research Ideology		
Basic Belief	Positivist	Constructivist/ Interpretivist	Design/Design Science
Epistemology	Objective; dispassionate. Detached observer of truth.	Subjective, i.e., values and knowledge emerge from the researcher-participant interaction.	*Knowing through making*: objectively constrained construction within a context. Iterative circumspection reveals the meaning.
Ontology	A single reality. Knowable, probabilistic.	Multiple realities, socially constructed.	Multiple, contextually situated alternative world states. Reality is socio-technically created and enabled.
Research strategy	Quantitative data: ratios, intervals; factor correlation.	Qualitative data: ordinal, nominal; factor meanings; thematic categories.	Development; unit of analysis: an organisational problem for deductive-inductive theory building.
Research method	Random experiment, field experiment, factorial experiment, quasi-experiment, hypothesis testing. Observation: quantitative, statistical.	Participation; hermeneutical, dialectical, critical analysis, grounded theory, phenomenology ethnography, action research, case study.	Mixed methods; measure artefactual impacts on the composite system, action research, case study.
Research technique	Data collection: sampling, interview, electronic, observation focus group, archival, surveillance, intrusive.	Inferential statistics: Within-group: correlation, cluster analysis, SEM,[2] regression, chi-squared between group: ANOVA,[3] ANCOVA,[4] MANCOVA.[5]	Using a combination of surveys and single-case mechanism experiments in an action research setting.
The relationship between theory and practice	The theory is used to produce the desired state of affairs in the physical world.	Theory cannot wholly be used to predict future situations.	Design theory is used to build predictably functioning artefacts.

[2] SEM: Structural equation modelling
[3] ANOVA: Analysis of variance
[4] ANCOVA: Analysis of covariance
[5] MANCOVA: Multivariate analysis of covariance

	Research Ideology		
Basic Belief	Positivist	Constructivist/ Interpretivist	Design/Design Science
Role of the researcher	Passive/value neutral observer.	Participant observer who initiates change in social relation.	Participant observer at an early stage, then more neutral observer later.

1.6 Research contribution

Hevner (2007) identified the three activities or cycles that form part of the design science research process, namely, the relevance cycle, the design cycle, and the rigour cycle. One of the motivational factors in a design science research project is the desire to improve something (either physically or in terms of understanding) by introducing new and innovative artefacts (Simon, 1996).

The research contribution for this study can thus be evaluated in terms of (a) the specific problem identified; (b) the improvement achieved in terms of the better understanding of the problem; and (c) the potential improvements to be made.

This study will contribute to the system engineering body of knowledge in the following fields:

- Revisiting the requirements engineering domain by taking a fresh and novel look at the different processes and problem areas.
- Designing a novel research tool that can be used by researchers and requirements engineering practitioners to investigate the requirements engineering process and improve it.
- Providing a practical application of the design science research process in the designing of a research tool that can be extrapolated by other researchers in future.

The aim of the research presented in this thesis is to propose the design and implementation of a research tool that will consist of a measurement artefact and a simulation artefact. The measurement artefact will comprise a research questionnaire whose aim is to establish the current performance of the object-of-study in terms of specific parameters. The simulation artefact will consist of a system dynamic simulation model that models the environment and processes found in the requirements engineering environment. The aim of this research tool is to enable requirements engineers or researchers to perform research within the requirements engineering environment to identify possible areas of improvement, design such improvements, and evaluate the improvements prior to implementing them in practice.

A complete review of the research contribution of this thesis is also provided in Chapter 7.

1.7 Thesis layout

The following chapters are presented as part of the study:

- **Chapter 1 – Introduction:** This chapter provides an introductory explanation of the background leading to the research, a detailed definition of the research problem that is being studied in the thesis, and the research design that will be used in this thesis.
- **Chapter 2 – Research problem identification phase:** This chapter identifies and describes the research problem and defines the research problem context in terms of the research goals, the current knowledge, and the applicable theoretical and conceptual frameworks. The chapter further describes the research environment in terms of the stakeholder groups, the operational research concept, and the object-of-study.
- **Chapter 3 – Research problem definition phase:** This chapter defines the research problem by establishing the individual stakeholders, describing the observed phenomena, and identifying how these phenomena are caused and by which mechanisms they are produced. It further evaluates the observed phenomena and whether they detract from the stakeholders' goals or not.
- **Chapter 4 – Requirements engineering research tool specification phase:** This chapter identifies the sample population for the research effort and identifies the stakeholders with the sample population to be interviewed. The requirements engineering research tool specification will then be developed using these inputs as well as the inputs from the previous chapters.
- **Chapter 5 – Requirements engineering research tool implementation and verification phase:** This chapter describes the implementation and verification of the measurement artefact and the simulation artefact of research tool according to the specification that was developed in the previous chapter.
- **Chapter 6 – Requirements engineering research tool validation phase:** This chapter describes the validation process of the research tool by performing research activities using actual data that has been collected by interviewing SMEs in various industry sectors. The purpose of the validation

process is to determine if the research tool could be used to answer the research questions stated earlier.

- **Chapter 7 – Thesis summary, conclusion and recommendations:** This chapter summarises the results of the research and provides the concluding remarks.

1.8 Thesis constraints and boundaries

The work presented in this thesis only covers the research process up to the stage at which the theoretical improvements are evaluated in the research tool as part of the validation process. The theoretical improvements will not be returned to the real-world environment and be implemented in practice.

1.9 Ethics

Ethical clearance for the research was obtained in line with the requirements of the University of Pretoria. The ethical clearance letter is included in Appendix A.

1.10 Peer-reviewed research articles published

The following peer-reviewed articles were published as part of the research effort for this thesis.

Table 7: Research articles published as part of the study

No.	Article
1	In the beginning... – Challenges in requirements elicitation (Scribante, Pretorius and Benade, 2015)
2	A conceptual system dynamics model for requirements engineering (Scribante, Pretorius and Benade, 2016a)
3	Conflict in the requirements engineering process (Scribante *et al.*, 2016b)
4	Elements of a system dynamics model for requirements elicitation (Scribante *et al.*, 2016c)
5	A single-stage system dynamics simulation model for requirements engineering (Scribante, Pretorius and Benade, 2017a)
6	Requirements engineering principles applicable to technology and innovation management (Scribante, Pretorius and Benade, 2017b)
7	Applying a design science research methodology to the design of a research instrument (Scribante *et al.*, 2018a)
8	Elements of a systems dynamics simulation model for requirements engineering (Scribante, Pretorius and Benade, 2018b)

1.11 Other PhD studies conducted in related fields

The PhD theses listed in Table 8 have been published in related research fields.

Table 8: PhD theses published in related research fields

No.	Thesis	Main research fields
1	Modelling methodology for assessing the impact of new technology on complex socio-technical systems (Oosthuizen, 2014)	• Complex socio-technical systems • Design science research • System dynamics modelling
2	The world according to MARP (Ter_Mors, 2009)	• Design science research
3	Sustainability if quality improvement programmes in a heavy engineering manufacturing environment: A systems dynamics approach (Van_Dyk, 2013)	• System dynamics modelling
4	The development of complex systems: An integrated approach to design influencing (Wessels, 2012)	• Complex systems
5	A situational approach and intelligent tool for collaborative requirements elicitation (Coulin, 2007)	• Requirements engineering • Requirements elicitation
6	A socio-technical view of the requirements engineering process (Marnewick, 2013)	• Requirements engineering • Socio-technical systems
7	Software evolution: A requirements engineering approach (Ernst, 2012)	• Requirements engineering
8	A framework to improve communication during the requirements elicitation process in GSD projects (Aranda, 2007)	• Requirements engineering • Requirements elicitation
9	Resolving requirements conflicts with computer-supported negotiation (Easterbrook, 1991a)	• Requirements engineering • Requirements elicitation
10	Tracing requirements and source code during software development (Delater, 2013)	• Requirements engineering • Requirements elicitation
11	Social networks and collaborative filtering for large-scale requirements elicitation (Lim, 2010)	• Requirements engineering • Requirements elicitation

No.	Thesis	Main research fields
12	The requirements dilemma (Harris, 2006)	• Requirements engineering • Requirements elicitation
13	Requirements engineering with interrelated conceptual models and real world scenes (Haumer, 2000)	• Requirements engineering
14	The derivation of a pragmatic requirements framework for web development (Jeary, 2010)	• Requirements engineering

1.12 Chapter summary and conclusion

We live in an artificial world where the utility that the artefacts must provide is described by the needs and requirements of the user of that artefact. The process for the discovery and elicitation of these requirements forms part of the requirements engineering domain. The requirements engineering domain can be classified as a complex socio-technical system in which social and technical elements interact. This complexity can result in an incorrect set of requirements being elicited that may eventually lead to a failed or challenged implementation project. A better understanding of the complex interaction between the social and technical elements and how this may impact the requirements engineering process will allow researchers and requirements engineering practitioners to identify and implement improvements to the requirements engineering process.

This thesis proposes the design of a requirements engineering research tool by applying a design science research methodology that can be used to investigate the requirements engineering domain. A research design framework was developed that implemented the various elements required to arrive at a validated research tool. This tool can be used to perform research within the requirements engineering domain with the aim of identifying and implementing incremental improvements.

The grounding of the research design was established by identifying and mapping the salient elements of the design research topology in paragraph 1.5.

The application of the elements of the framework will be defined in the following chapter, starting with the research problem identification phase in Chapter 2.

CHAPTER 2. RESEARCH PROBLEM IDENTIFICATION PHASE

"It's really hard to design products by focus groups. A lot of time, people don't know what they want until you show it to them." – Steve Jobs

2.1 Introduction

The purpose of the research problem identification phase is to articulate the research problem, identify the context within which the research problem occurs, and define the research environment within which the research problem will be studied.

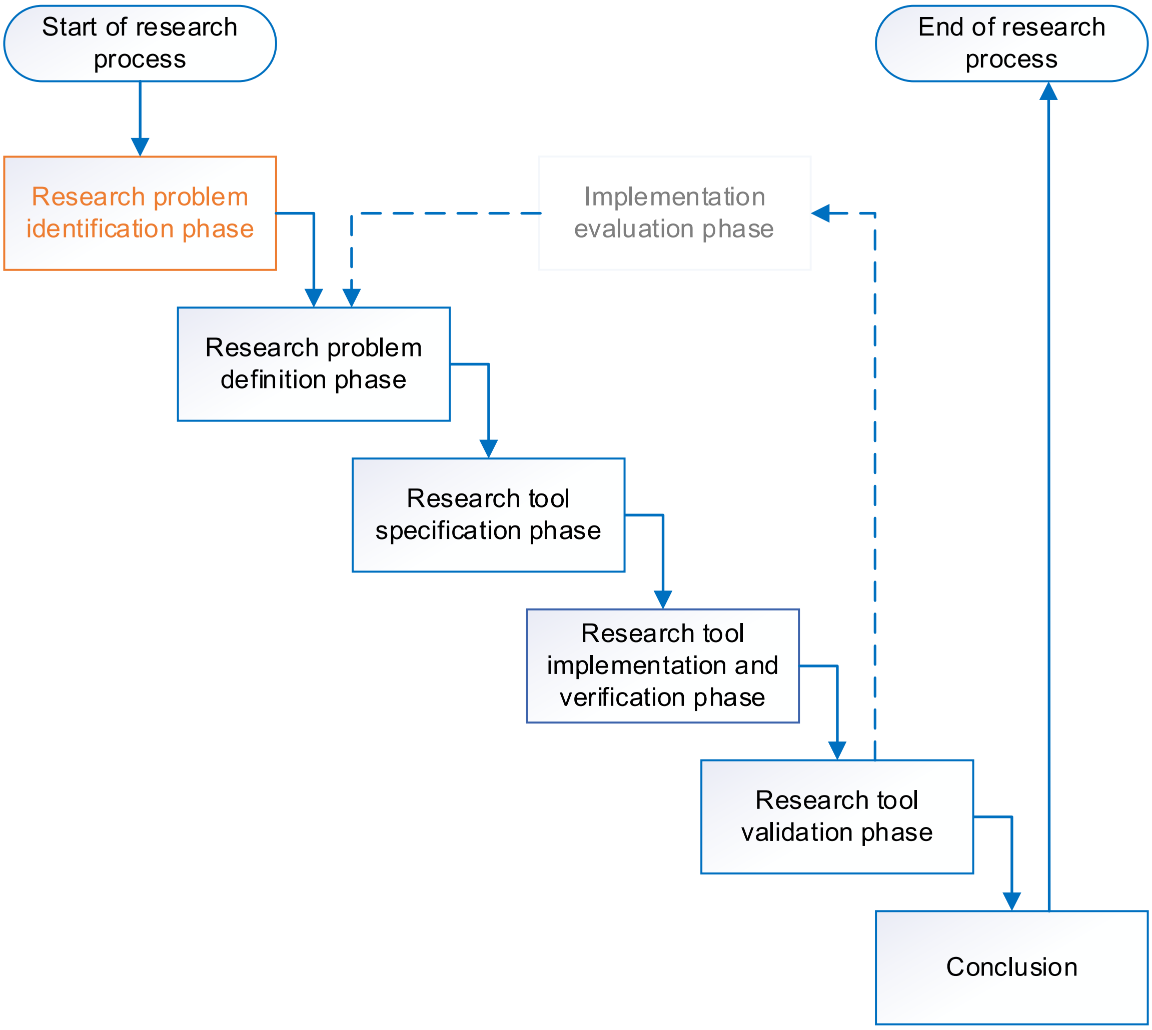


Figure 18: Relative position of the research problem identification phase in the overall research process

Figure 18 shows the FFBD as was previously presented in Figure 10, but with the relative position of the research problem identification phase highlighted. The detail flow of the discussion for the rest of the chapter is presented in Figure 19.

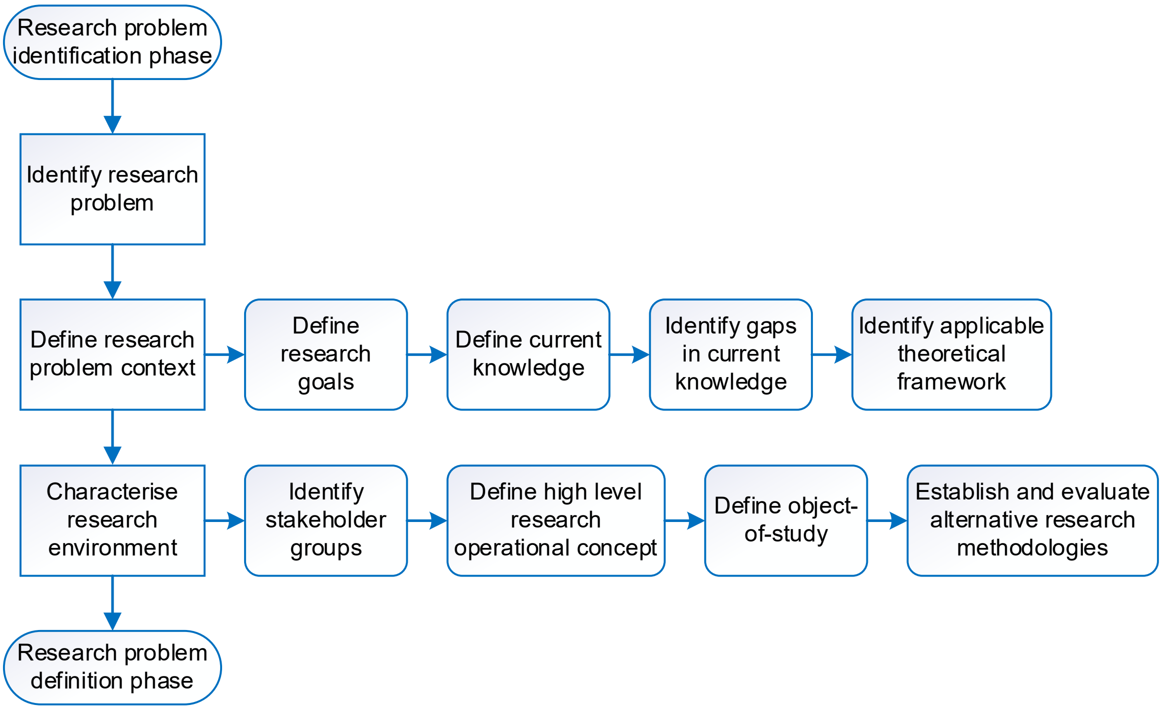


Figure 19: Research problem identification phase FFBD

2.2 Research problem identification

A brief description of the problem scenario was given in paragraph 1.1, and the research problem statement was given in paragraph 1.2. The research problem identified in Chapter 1 is linked to the difficulty of identifying the correct project requirements within a complex socio-technical system. Based on the identified research problem, a design science research methodology was selected in paragraph 1.3 as the dominant research process.

Specific guidelines are identified in the literature regarding the suitability of research problems to qualify to be approached using a design science research methodology. These guidelines include the relevance of the problem identified, the novelty of the problem, and the significance of the problem to the stakeholders should it not be resolved (Baskerville *et al.*, 2009; Geerts, 2011; Hevner *et al.*, 2004; Vaishnavi *et al.*, 2015). In evaluating the identified problem for this thesis against these criteria, the following results were obtained:

1. **Problem relevance:** Evidence from the literature was presented in Chapter 1 regarding the effect on the success of a project when working with an incorrect or incomplete set of requirements. Even though it is a well-known phenomenon, it was also argued that there had been no significant improvement in the requirements engineering process relating to the elicitation of incorrect requirements over the last decade with similar errors still occurring today as during the execution of projects ten years ago.

2. **Problem novelty:** The problem is not a novel problem but approaching the problem from an improvement point of view is unique and has not yet been reported on explicitly in literature. Many approaches have been discussed in literature ranging from various elicitation methods to the use of various agile project execution methods. These methods focus on specific elements of the problem but do not view the problem from a complex socio-technical perspective.
3. **Problem significance:** The significance of the problem in terms of the financial effect that it has on organisations is enormous. Some cases have even been reported in which a failed project has resulted in companies closing down (Charette, 2005).

2.3 Research problem context

2.3.1 Research goals

The research goals are aligned with the research problem statement identified in paragraph 1.2, which are achieved by solving research problems. The primary research goal of this study can be stated as:

"To explain the cause-effect behaviour observed within the requirements engineering domain that results in requirements volatility."

By understanding the various factors in play and how they may combine and interact will increase the understanding of the requirements engineering domain in general and the problem context specifically. In turn, this increased understanding will ease the elicitation of requirements in a complex socio-technical system.

This type of research can be classified as problem investigation research taking place within a real-world environment. This type of research is ideally suited to generate more Mode 2 knowledge (Frost *et al.*, 2003). The research method that will be used in this study is a single-case mechanism experiment.

The research goal is in turn divided into a design goal and a knowledge goal. In this study, the design goal is identified as the design and implementation of a research tool. This type of design goal can also be classified as an instrument design goal. Instrument design goals are achieved by solving design problems surrounding the design of the research tool. In turn, the knowledge goal is achieved by answering the posed knowledge questions by applying the research tool to the problem context

(Wieringa, 2014). The relationship between the instrument design goal and the knowledge goal is shown in Figure 20.

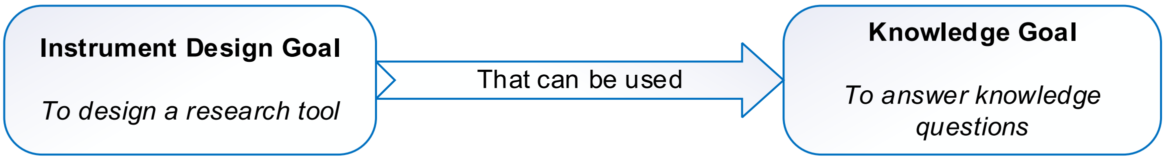


Figure 20: Relationship between knowledge goals and design goals [adapted from Wieringa, (2014)]

The instrument design goal for this study can be stated as follows:

> *"To improve the understanding of the requirements engineering process by designing and implementing a requirements engineering research tool that can be used to investigate and explain the phenomena observed in the requirements engineering process."*

The identified knowledge goal for this study can be stated as the following:

> *"To identify and describe the observed phenomena, their interactions, and the effect of these interactions as observed in the complex requirements engineering domain that may result in the incorrect set of requirements being used in the project implementation."*

2.3.2 Current knowledge

The purpose of detailing the current knowledge available is to support the research process in three areas. The first area is related to the relevance arguments being made as to why the research matters. The second area is related to the rigour arguments of the research process in considering what has been done and reported on previously in academic literature and technical literature among other sources. The third area is to support the definition of the object-of-study. (Also refer to the discussion in paragraph 1.3.3 regarding the relevance and rigour cycles inherently present in the design science research methodology.)

The prevalence of requirements engineering problems occurring in projects was detailed in Chapter 1. These failures manifest in terms of project budget overruns, projects exceeding set timescales, and/or projects underperforming in terms of stakeholder expectations. The effect of a failed project can be catastrophic to all involved and, in some instances, even result in the closure of companies with a resulting loss of jobs (Charette, 2005).

The reasons for the failures are dependent on the role of the person being interviewed. From the perspective of senior managers, these range from estimation and planning not being done accurately enough to project status reporting often being wrong or misleading. The quality and reliability of delivered projects are unacceptably low (Jones, 2006).

The view of project managers in the same study reflects an opposing view. Their comments included stakeholders not accepting accurate and conservative estimates, and project managers experiencing unrealistic time pressure in terms of the required delivery schedule, dealing with scope changes made during the development phase, coping with political interference, doing risk management, and managing the size and volatility of the project (Jones, 2006; Ratsiepe and Yazdanifard, 2011; Rost, 2004; Sauer, Gemino and Reich, 2007). Some project managers cite the inherent complexity of projects as the main reason for project failure (Daniels and Lamarsh, 2007). Others cite human factors as one of the leading causes of project failure (Mohamadi and Ranjbaran, 2013).

Verner, Sampson and Cerpa (2008) identified a range of factors including a broad spectrum of issues as being key to project failure. These issues include organisational structure; unrealistic or unarticulated goals; software failing to meet the real business needs; poorly defined system requirements, user requirements and requirements specification; the project management process; poor project management; software development methodologies; sloppy development practices; scheduling and project budget; inaccurate estimates of needed resources; poor reporting of project status; inability to handle project complexity; unmanaged risks; poor communication among customers, developers and users; use of immature technology; stakeholder politics; commercial pressures; customer satisfaction; product quality; leadership; upper management support; personality conflicts; business processes and resources; and weak or no tracking tools.

What is clear from the literature cited above is that there is no one single reason that can be pinpointed as being responsible for project failure. As a result, there is no single solution in either technology or management technique that can be identified that will bring a solution or even a significant improvement to the table (Brooks, 1986; Serna, Bachiller and Serna, 2017).

One of the reasons identified in the literature that plays a significant role in the success or failure of a project is the problematic nature of requirements (Attarzadeh and Ow, 2008; Cui and Loch, 2014; Dorsey, 2005; Frese and Sauter, 2006; Geneca LLC., 2011; Jones, 2006; Kappelman, McKeeman and Zhang, 2006; Méndez Fernández, Wagner, Kalinowski, Felderer, Mafra, Vetrò, Conte, Christiansson, Greer, Lassenius, *et al.*, 2017; Morisio, Egorova and Torchiano, 2007; Pinto and Mantel, 1990; Rost, 2004; Verner *et al.*, 2008).

Christel and Kang (1992) grouped the problems associated with requirements engineering into three categories: (a) problems of scope in which the requirements address too much or too little information; (b) problems of understanding within the different groups and users within a project; and (c) problems of volatility in which the requirements change frequently within a project.

Bubenko (1995) identified several challenges in the requirements engineering process. These challenges included (a) management and organisational challenges; (b) user-stakeholder and developer challenges; (c) methodology challenges, (d) support tools challenges; and (e) research and education challenges. These categories of challenges serve to confirm the complexity of the requirements engineering environment.

Many of the challenges relate to the elicitation of requirements. In this regard, various requirements elicitation methods have been defined and described in the literature. These methods include, among many others, tools and techniques such as interviews, questionnaires, task analysis methods, domain analysis, introspection, card sorting, group work, and brainstorming (Zowghi *et al.*, 2005).

Sommerville and Sawyer (2003) further identified five potential risks that a requirements engineer must address during the requirements elicitation phase. These risks include that (a) the stakeholder often does not really know what they need; (b) the stakeholders express their needs in their own terms and with implicit knowledge of their environment; (c) different stakeholders may have different requirements and may express these requirements in different ways; (d) organisational issues and political factors may play a significant role in influencing the requirements elicited; and (e) the economic and business environment that the organisation finds itself in is dynamic and may change during the elicitation process.

The number of tools and techniques available do not necessarily hold the promise of success. Hickey *et al.* (2004) observed that even with all the various methods available, the requirements engineer may only use a small fraction of them in each situation. The requirements engineer may select a specific technique since (a) it is the requirements engineer's favourite technique; (b) a specific methodology is being followed; (c) it is the only technique known to the requirements engineer; and (d) the requirements engineer intuitively feels that it is the correct technique.

What becomes apparent from the literature is that there is no universal answer that can solve all the requirements engineering related problems in a single stroke due to many reasons, some of which have already been discussed (Brooks, 1986). Therefore, one of the aims of this thesis is to view the problems and challenges experienced in the requirements engineering domain as an improvement problem.

2.3.3 Applicable theoretical and conceptual problem frameworks

The function of the conceptual framework is to define what is being studied, why it is being studied (relevance), and how it will be studied (methodology and rigour) (also refer to paragraph 1.3.3)(Dickson, Hussein and Agyem, 2018; Ravitch *et al.*, 2017). This will be done by answering the following questions:

1. What is being studied and why?
   The purpose of this question is to establish the relevance of the research being conducted.
2. How will this research topic be studied?
   Where will the data for the study be obtained from?
   What theoretical framework will support the research?
   How will the data be analysed?
   The purpose of these questions is to establish the research methodology and research rigour of the study.

The references to the relevance and rigour of the research effort also echo the thoughts of Hevner (2007) as encapsulated in the three-cycle view of design science research.

2.3.3.1 Research relevance

The problem statement as defined in paragraph 1.2 focuses on the difficulty of establishing the correct requirements within a complex socio-technical system. This problem statement reflects the experience of the researcher in this thesis over a period of more than 30 years in the field of system engineering and programme management within a high-technology environment. During this period, it was observed that identifying the actual needs and requirements of the customer proved to be extremely difficult. The overall impression was that the customer did not always seem to know what is needed. This idea is shown conceptually in Figure 21.

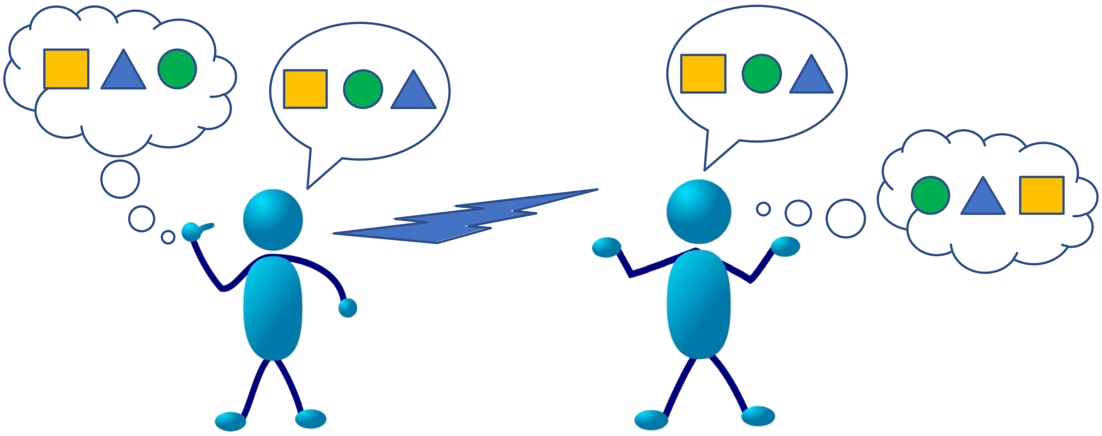


Figure 21: Rich picture of the perceived problem situation

The researcher's experiences were discussed with other professional people in similar roles within various industry sectors. From the discussions, it became clear that it was not an isolated problem that was observed but seemed to be a broader problem present in various industry sectors. It further became clear from studying the available literature on the requirements engineering environment and process that there is no single solution available to solve all the experienced problems [based on the number of tools and methods discussed in literature, for example Achimugu, Selamat, Ibrahim and Mahrin (2014); Al-Zawahreh and Almakadmeh (2015); Alspaugh and Antón (2008); Amber, Bajwa, Naweed and Bashir (2011); Boulila, Hoffmann and Herrmann (2011); Broll, Hussmann, Rukzio and Wimmer (2007); Burnay (2016); Chua, Bernardo and Verner (2010); Geisser and Hildenbrand (2006); Hickey and Davis (2003a); Lempia, Schindel, Mcgill and Graber (2016); and Sendall and Strohmeier (2002)]

The motivation for the study is thus to enable the researcher, other researchers, and requirements engineering practitioners to increase their understanding of the problem context with the aim of improving the requirements engineering process for future projects. This will be done by the provisioning of the requirements engineering research tool that forms one of the objectives of this thesis.

2.3.3.2 Research rigour

Figure 22 shows the overall research approach followed in this thesis.

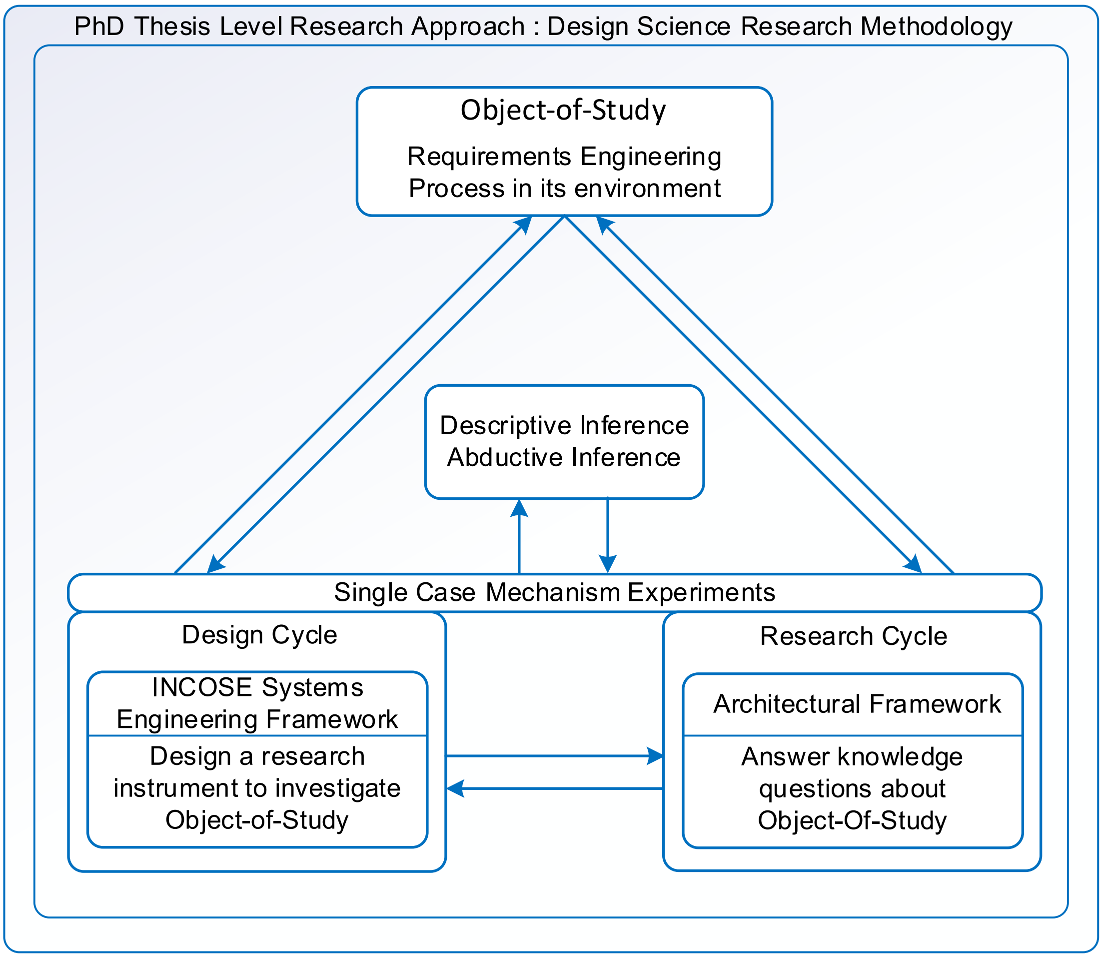


Figure 22: Overall research approach (own contribution)

The overall research methodology that will be applied to the research reported on in this thesis will be a design science research methodology. The main focus of the design science research methodology is to investigate an artefact interacting with its intended problem context (Wieringa, 2014). The knowledge and insight gained from this research methodology are neither from the artefact only nor from the problem context only, but rather from the interaction between the artefact and the problem context.

One of the main arguments for selecting this research approach is that the requirements engineering environment is a complex socio-technical system. Within such a system, a complete solution to a problem is rarely found. An alternative approach within this environment is to treat a problem instead as an improvement problem – something to which the design science research methodology is ideally suited.

Various research instruments will be used in the study. These instruments will include techniques such as interviews, literature surveys and questionnaires

(Berenbach, Paulish, Kazmeier and Rudorfer, 2009; Sommerville *et al.*, 2003; Young, 2004). The experimental aspects of the study will be done by using single-case mechanism experiments that will employ semi-structured interviews conducted in an action research setting. The research population will be selected from a pool of high-technology firms that operate in different industry sectors.

The data obtained through the single-case mechanism experiments will be analysed via a descriptive and abductive inference process using causal, mechanistic and rational explanations (Bunge, 2004; Wieringa, 2014; Williamson, 2011). The abductive reasoning process will be supported by a systems dynamic simulation model of the requirements elicitation process (Flood *et al.*, 1991).

The design cycle present in the design and implementation of the single-case mechanism experiment will be based on the INCOSE system engineering framework. The research cycle will be based on a single-case mechanism experiment supported by an architectural framework.

2.4 Research environment

2.4.1 Research stakeholder classes

The research stakeholder classes are the various parties that have a vested interest in the outcome of the design science research effort. Alexander (2005) identified three classes of stakeholders, namely, those involved in operating the system, those involved in containing the system, and those involved in the broader system or environment. This relationship is shown in Figure 23.

The main distinction between these classes of stakeholders is the degree to which they are involved in the system. This involvement decreases as the class of stakeholder moves out from the centre of the onion diagram. The stakeholder who operates the system is the most deeply involved. The next layer represents stakeholders who are not directly involved in operating the system but who are users who depend on the product that the system delivers. The outermost layer represents the stakeholders who are not directly dependent on the product of the system but instead on the information produced by the system (Alexander, 2005).

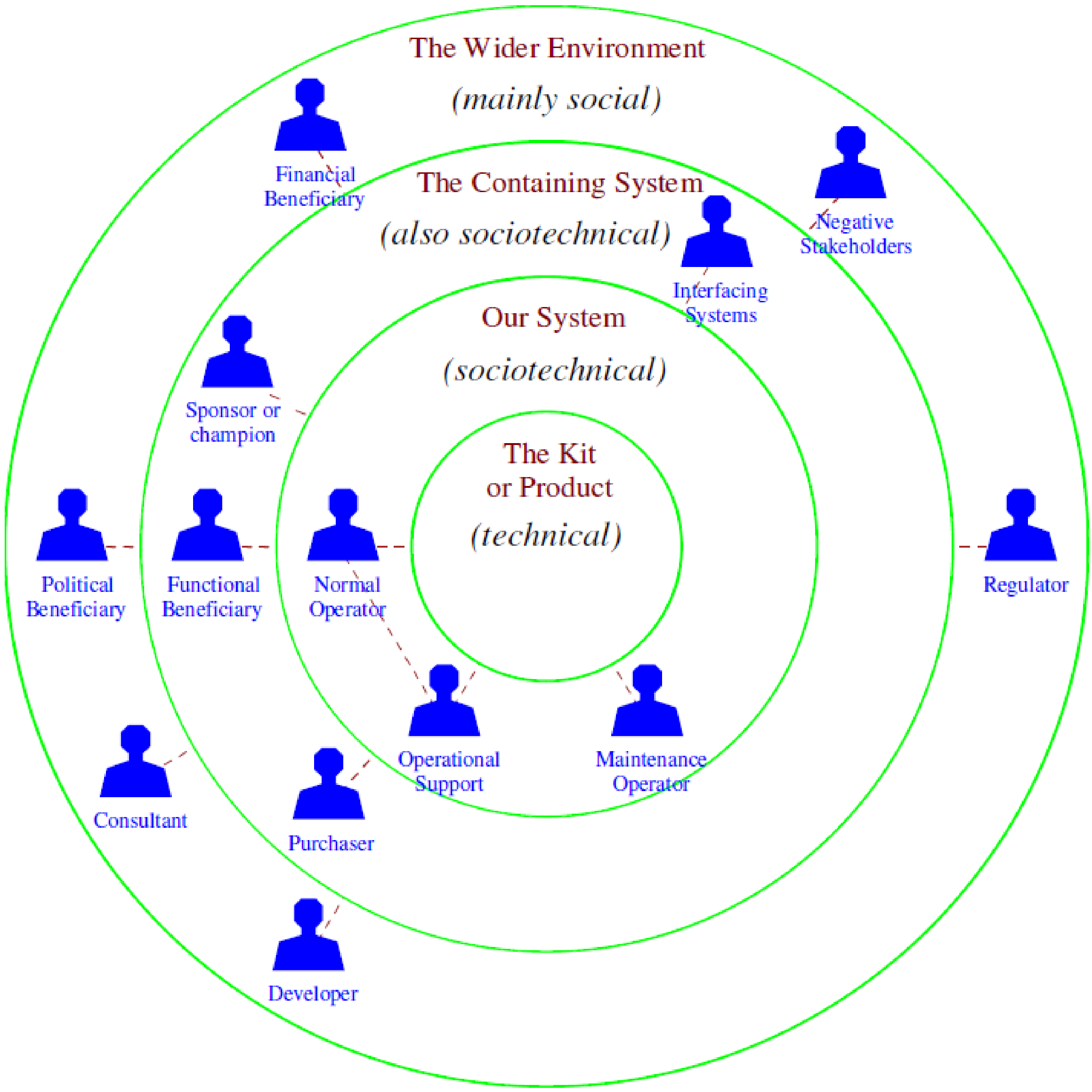


Figure 23: Relationship between the various classes of stakeholders (Alexander, 2005)

2.4.2 High-level research operational concept

The research conducted as part of this study will be done through a combination of theoretical research, modelling, simulation and interviews. The purpose of this study is to arrive at a validated requirements engineering simulation tool that can be used by other researchers and requirements engineering practitioners to evaluate the current state of the requirements engineering environment within a specific organisation.

This current state will be evaluated using various methods including the use of causal loop diagrams, descriptive and abductive inference, and a systems dynamic simulation model. The primary approach will be to evaluate the current state of the organisation, identify and evaluate alternative improvements outside of the organisation and, finally, roll out the potential improvements incrementally.

2.4.3 Object-of-study

2.4.3.1 Introduction

The object-of-study defines the problem space by characterising the environment within which the problem that needs to be solved occurs. This activity includes identifying gaps, describing the problems underlying the gaps, and obtaining agreement on the problem description. The general form of the object-of-study was introduced in paragraph 1.4.1.6.

The process of defining the object-of-study specific to this research was performed using research approaches such as literature surveys and personal reflection based on more than 30 years' experience in the field. The process continued by conducting semi-structured research interviews (related to action research) with SMEs in different fields to determine if any additional parameters could be identified that form part of the problem space (Järvinen, 2007; Lee, 2007).

The object-of-study can be divided into two main areas. First, the requirements engineering process artefact discussed in detail in paragraph 2.4.3.2 and, second, the project problem context that the requirements engineering process artefact interacts with. The project problem context is discussed in detail in paragraph 2.4.3.3. A schematic representation of the object-of-study is shown in Figure 24.

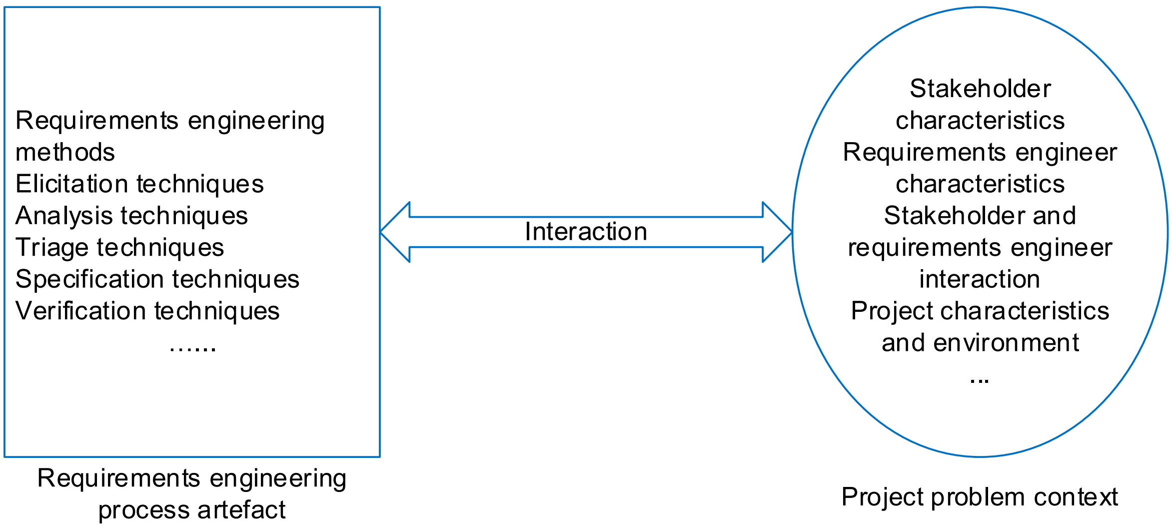


Figure 24: Requirements engineering object-of-study (own contribution)

2.4.3.1 Requirements engineering process artefact

2.4.3.1.1 Requirements engineering

2.4.3.1.1.1 Definition

Sommerville *et al.* (2003) defined the requirements engineering process as a structured set of activities that are followed to derive, validate and maintain a set of requirements that are captured in writing in some form. An alternative definition is provided by ISO/IEC/IEEE 29148 (2011:5) defines requirements engineering as an "… *interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest*" The ISO/IEC/IEEE 29148 specification further states that requirements engineering is concerned with "*discovering, eliciting, developing, analysing, determining verification methods, validating, communicating, documenting, and managing requirements*" (ISO/IEC/IEEE, 2011:5).

A requirement can further be defined as "*... a statement which translates or expresses a need and its associated constraints and conditions*" (ISO/IEC/IEEE, 2011). A common understanding within the engineering industry is that the focus of a requirement should rather be on the 'what' that needs to be solved rather than on the 'how' it should be solved (Faulk, 1997; Ryan and Wheatcraft, 2017).

A requirement is a statement that translates or expresses a need and its associated constraints and conditions (ISO/IEC/IEEE, 2011). It is the golden thread that runs through any engineering effort from the beginning to the end, whether one is managing a technology roadmap, a technology project portfolio or an acquisition project. Requirement defines attributes in terms of capability, characteristics or quality of the system, product or service so as to have value for a customer (Young, 2004).

This definition distinguishes between a need and a requirement. In this context, a need is an expectation stated by the stakeholders at business management level or at business operations level as indicated in Figure 25. A requirement is a formal statement that is structured in a specific way so as to be able to be verified within the design of the artefact being created as well as be verified back to the original need and expectation from which the requirements were developed (INCOSE, 2017).

2.4.3.1.1.2 Transformation of a need into a requirement within an organisation

Figure 25 shows the process of transforming a need into a requirement within an organisation. Requirements are derived from the needs through the requirements engineering process. One need may give rise to several requirements (a one-to-many relationship), whereas a requirement can be traced back to only one need (a one-to-one relationship).

Requirements can be expressed in many different formats, including diagrams as part of a model-based systems engineering approach, tabular format or in a natural language format (INCOSE, 2017).

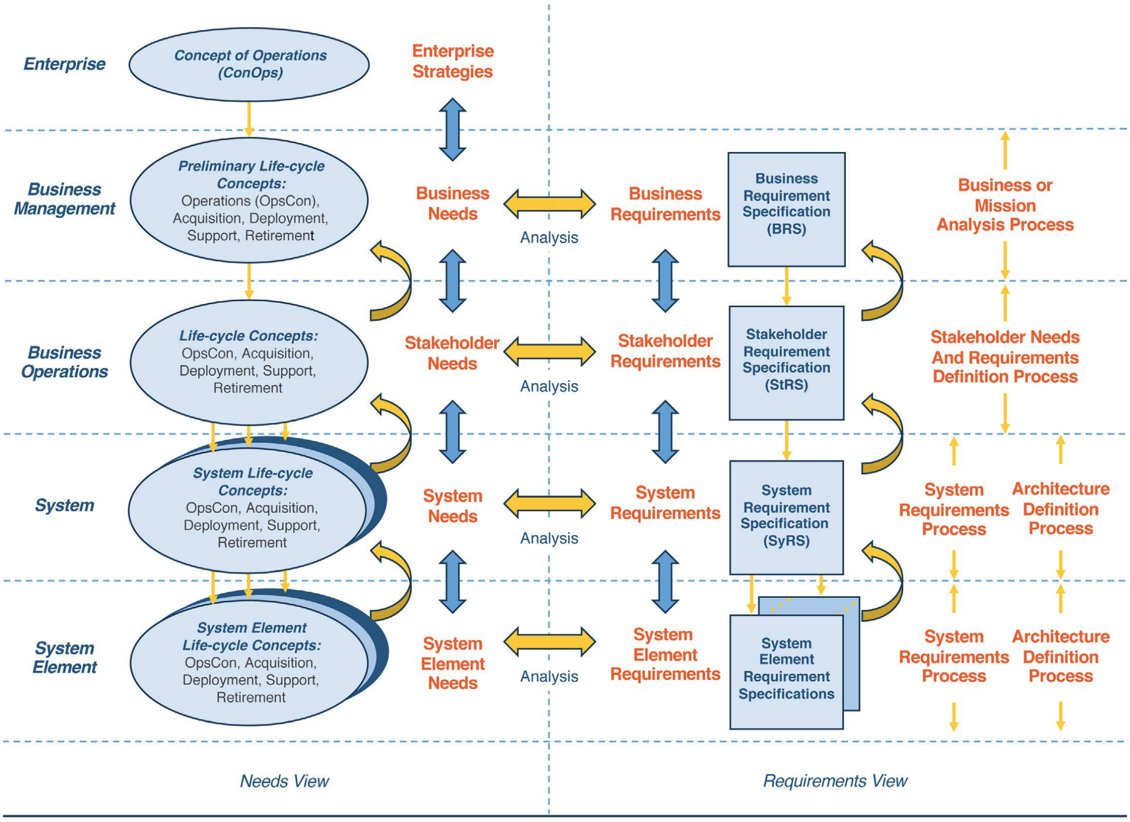


Figure 25: The process of transforming needs into requirements (Ryan, 2013)

Once a set of requirements have been agreed upon and baselined at one level, they will flow to the next level. Here, the requirements will be expanded and detailed based on the decomposition and transformation of the needs at that specific level.

As requirements are developed from the top level down to the lower levels, it can be expected that the requirements statement will become more explicit and specific. At the highest level, the ideal requirement should not specify a solution, thus permitting a wide range of solutions to be considered; while at the lowest level, a requirement statement may be particular to the selected solution (INCOSE, 2017).

2.4.3.1.2 The requirements engineering process

There are various views in the literature regarding the activities or processes that form part of requirements engineering. Jarke and Pohl (1994) identified the following activities: elicitation, expression and validation. Richards (2000) listed requirements acquisition and conversion, concept generation, concept comparison and conflict detection, conflict negotiation, and evaluation. Wiegers (2000) divided requirements engineering into the two sub-disciplines of requirements development and requirements management. Under the sub-discipline of requirements development, he included the processes of elicitation, analysis, specification and verification. Hickey *et al.* (2004) expanded on the process grouped under the requirements development sub-discipline to include the process of triage. The requirements development sub-discipline is thus summarised as follows as a consolidated list of processes:

- **Requirements development**
  - **Discovery and elicitation:** The requirements discovery and elicitation processes are about the systematic extraction, identifying, learning, uncovering, extracting, and surfacing of the needs of customers, users and other potential stakeholders via communication. It requires application domain, organisational and specific problem knowledge (Ahmad, 2008; Sommerville *et al.*, 2003).
  - **Analysis:** The analysis process in the requirements engineering process is involved with analysing the information elicited from the different stakeholders for conflicts, omissions and inconsistencies. This can be done in many ways including by creating and analysing models of the requirements. The primary purpose of this analysis is to increase the requirements engineer's understanding of the problem domain and to enable the search for incompleteness and inconsistency (Sommerville *et al.*, 2003).
  - **Validation:** The purpose of the validation process is to determine the reasonableness, consistency, completeness, suitability, and lack of defects within a set of requirements (Young, 2004).
  - **Negotiation and triage:** The purpose of the negotiation and triage process is to determine which subset of the requirements ascertained by elicitation and analysis is appropriate to be addressed in a specific release of a system. This activity also includes requirements prioritisation, resource allocation and negotiation (Davis AM, 2003; Sommerville *et al.*, 2003).
  - **Documentation or specification:** The purpose of documenting a specification activity is to capture the requirements in a format that will be usable to the different stakeholders, whether it is in the form of a systems requirement specification or in a requirements database (Hickey *et al.*, 2004; Sommerville, 2011; Wiegers, 2000).

- **Requirements management:** Requirements management is a formal process to maintain a set of requirements, including change management (Sommerville *et al.*, 2003). As part of the formal process, this activity also includes the following elements:
  - **Traceability:** This term refers to tracing requirements upward to their source documents, and – for derived requirements – to their parent requirements, and downward to child requirements and design elements.
  - **Requirements allocation:** The term refers to allocating requirements to the physical and functional hierarchies of the system.
  - **Categorising requirements:** This term refers to the capturing of information associated with requirements (e.g., assumptions, rationale).
  - **Compliance verification:** Verifying that the design of the system or final as-built system is compliant with each requirement.
  - **Common repository:** Capturing and tracking all the requirements in a common repository (Pajerek, 1998).

Figure 26 shows a graphic representation of the requirements engineering processes.

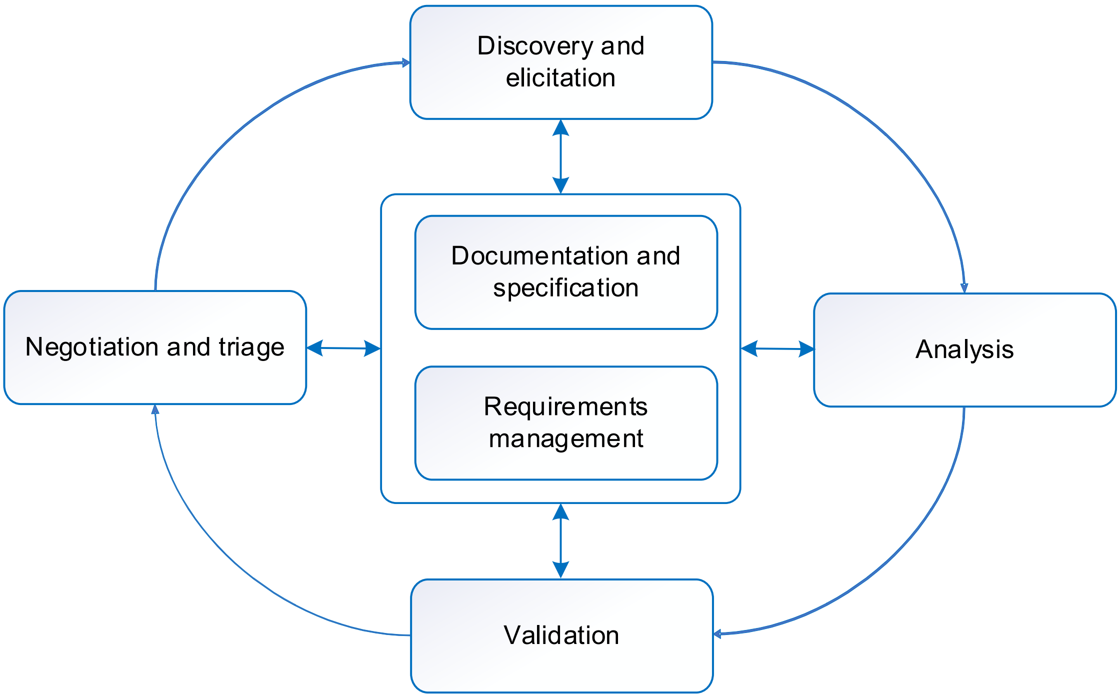


Figure 26: Requirements engineering process model [adapted from Sommerville (2005)]

The requirements engineering process is often viewed as a serial process in which one activity must be completed prior to the next activity commencing. More likely than not, it will be found that the requirements engineering process will be a quasi-parallel process in which some of the activities may overlap to a certain extent as is shown in Figure 27 (Hickey *et al.*, 2004).

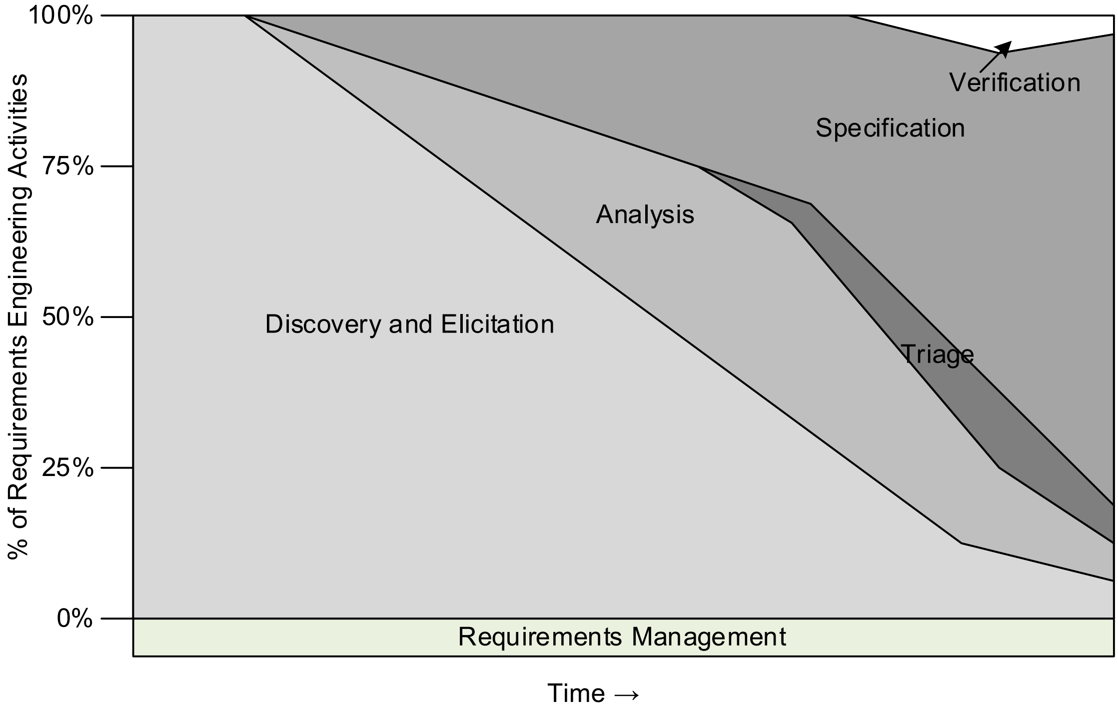


Figure 27: Parallel requirements engineering process model [adapted from Hickey et al. (2004)]

In addition to being a quasi-parallel process, the requirements engineering process is an iterative process in which the overall process model (shown in Figure 26) may be repeated multiple times during the life cycle of a project (Sommerville *et al.*, 2003).

2.4.3.1.3 Sources of requirements

Requirements can be elicited from three primary sources, which are typically used concurrently throughout the requirements engineering process. These sources are:

- **Stakeholders:** The definition of stakeholders that can serve as a source of the requirements is extensive and can range from the acquiring party to even include the requirements engineer who is involved with the elicitation of the requirements (also refer to Figure 23 for a view on the different classes of stakeholders). From this, it is essential to realise that a stakeholder is much more than just the customer or the end user. Pearce II and Robinson Jr

(1991) defined a stakeholder as "*… influenced individuals and groups … that are vitally interested in the actions of the business*". Glinz and Wieringa (2007) defined a stakeholder as *"… a person or organisation who influences a system's requirements or who are impacted by that system*".

The critical aspect of this definition is that a stakeholder is not just the person who is directly involved with the system, but also groups of people such as the community who may be affected by aspects such as pollution that may result from the operation of the system. It is also clear from this definition that the requirements engineer who is involved in the process is also a direct stakeholder as this person can indirectly influence the requirements elicited.

Furthermore, stakeholders have specific characteristics that will have a direct influence on the range and quality of requirements elicited from them. The characteristics typically include the following aspects or dimensions:

  - Inherent knowledge (domain or technical);
  - Experience;
  - Role within the organisation (strategic, tactical or operational); and
  - Interpersonal skills (Anwar *et al.*, 2012).

- **Documentation:** Documents include sources such as feasibility studies, market analyses, business plans, analysis of competing products, specifications, manuals, forms, job descriptions and standards that need to be complied with.
- **Existing, legacy and competing systems:** Existing and competing systems can be considered as a source of requirements and may include systems or features in systems or products that the organisation wish to imitate or copy. In such a case, the complete set of requirements will need to be reversed-engineered from the actual product or service (Ahmad, 2008; Anwar *et al.*, 2012)(Anwar *et al.*, 2012). Legacy systems may be an existing system or product already in service in the organisation that may have functions and features that may need to be recreated.

Within these three primary sources of requirements, the requirements elicited from the stakeholders will be the most prone to error due to several factors including:

- Changing needs resulting from the organisational dynamics and complexity;
- Conflict between stakeholders;
- The interaction between the stakeholder and the requirements engineer;
- The personal perceptions and concerns of the stakeholders;

- The social and political environment within the organisation (Ahmad, 2008; Davis, Fuller, Tremblay and Berndt, 2006; Sommerville *et al.*, 2003; Zowghi and Nurmuliani, 2002).

2.4.3.1.4 Requirements elicitation techniques and technique selection methods

The literature defines various methods and techniques that can be used to elicit requirements. These methods include interviews, questionnaires, task analysis methods, domain analysis, introspection, card sorting, group work, brainstorming, JAD, prototyping, use cases and storyboards among many other techniques (Ahmad, 2008; Al-Zawahreh *et al.*, 2015; Amber *et al.*, 2011; Bochmann, 2009; Carrizo, Dieste and Juristo, 2014; Dieste, Lopez and Ramos, 2008; Zowghi *et al.*, 2005).

The selection of the correct requirement elicitation technique depends on the type and level of interaction between the stakeholder and the requirements engineer. This interaction, as shown in Figure 28, splits the interaction between the stakeholder and the requirements engineer into four quadrants. In quadrant (a), the requirements are known to both the stakeholder and the requirements engineer. In quadrant (b), there are requirements not known by the stakeholder, but which are known by the requirements engineer based on his unique experience in the system domain. Quadrant (c) represents requirements known by the stakeholder due to his experience in the system domain, but not by the requirements engineer. Quadrant (d) represents requirements known by neither the stakeholders nor the requirements engineer (Davis CJ *et al.*, 2006).

Requirements elicitation techniques can further be grouped into four main categories, namely, traditional methods, collaborative methods, contextual methods and cognitive methods (Sharma and Pandey, 2013). Within these methods, different techniques are available that may be used in different circumstances:

- **Traditional methods:** Structured interviews, unstructured interviews, surveys, questionnaires, introspection, reading existing documents, and meetings.
- **Collaborative methods:** Brainstorming, JAD/RAD sessions, observation, prototyping, and focus groups.
- **Contextual methods:** Ethnographic techniques, discourse analysis and socio-technical methods (soft system analysis).
- **Cognitive techniques:** Protocol analysis, task analysis, and knowledge analysis that include card sorting, repertory grids, proximity-scaling techniques, and laddering (Anwar *et al.*, 2012; Sharma *et al.*, 2013).

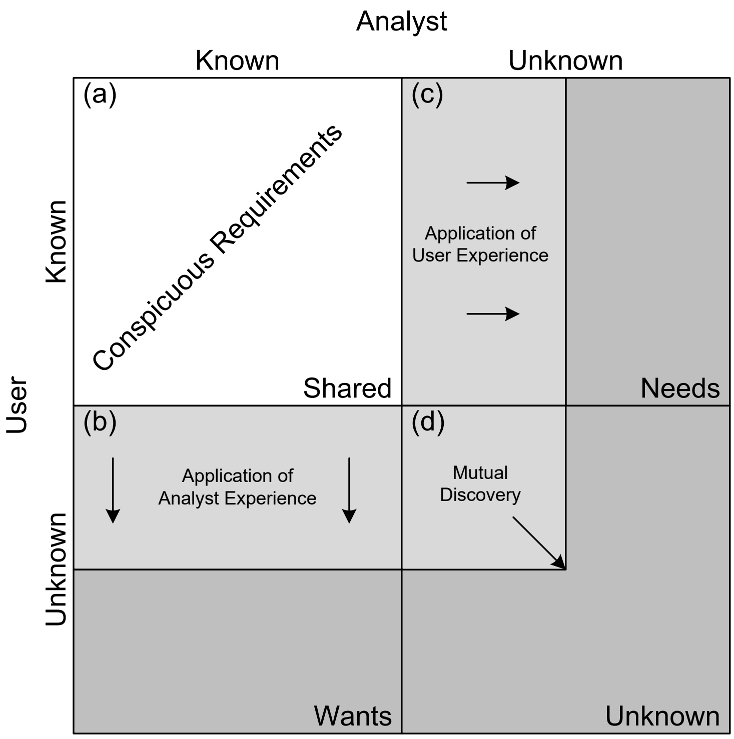


Figure 28: Stakeholder/requirements engineer interaction during requirements elicitation [redrawn from Davis CJ et al. (2006)]

Selecting the correct elicitation techniques is essential to ensure that the desired results are obtained. A study by Hickey and Davis (2003b) identified that requirements engineers typically select a technique based on a combination of four reasons. These reasons include: (a) it is the only technique known to the requirements engineer; (b) it is the favourite technique of the requirements engineer; (c) the requirements engineer is following some explicit methodology that dictates the technique to be used; and (d) the requirements engineer understands intuitively which technique is the most effective under the current circumstance.

In analysing these reasons, it should be clear that the last reason is the one most likely to produce the best results, while the first three reasons may contribute to missed or misinterpreted requirements. Selecting an inappropriate method may produce several of the effects such as communication challenges, more time required for the elicitation process and missed requirements, all contributing to requirements volatility.

2.4.3.1.5 Characteristics of good requirements

During the requirements elicitation process, many requirements will be provided by the stakeholders of which the quality will vary greatly (Fuentes, Fraga, Génova, Parra, Alvarez and Llorens, 2016). Care must be taken to ensure that only proper quality requirements are included in the requirement set. A requirement statement has previously been defined as a formal transformation of one or more needs into an agreed-upon obligation for an entity to perform some function or possess some quality (INCOSE, 2017). From this definition, two groupings can be identified, namely, formal transformation and agreed-to obligation. Within these two groupings, certain characteristics of 'well-formed' requirements can be identified. These characteristics can in turn again be applicable to individual requirements, sets of requirements, requirement statements and attributes of requirement statements.

- **Necessary:** A requirement defines an essential capability, characteristic, constraint, or quality factor. If not included in the set of requirements, a deficiency in capability or characteristic will exist, which cannot be fulfilled by implementing other requirements (INCOSE, 2017). If there is doubt about the necessity of a requirement, then ask: What is the worst thing that could happen if these requirements are not included? If you do not find an answer of consequence, then you probably do not need the requirement (Hooks, 1993).
- **Singular:** A requirement should state a single capability, characteristic, constraint, or quality factor (INCOSE, 2017). A requirement cannot sensibly be expressed as two or more requirements (Halligan, 2012).
- **Conforming:** Individual requirements should conform to an approved standard pattern and style for writing requirements (INCOSE, 2017).
- **Appropriate:** The specific intent and amount of detail of the requirement are appropriate to the level of abstraction of the entity to which it refers (INCOSE, 2017).
- **Correct:** Absence of errors of fact in the specified requirement (Halligan, 2012). The requirement must be an accurate representation of the entity need from which it was transformed (INCOSE, 2017).
- **Consistent:** A requirement is not in conflict with any other requirement, nor is it inconsistent internally (Halligan, 2012). The set of requirements contains individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems they use are homogeneous. The language used in the set of requirements is consistent (i.e., the same word is used throughout the set to mean the same thing) (INCOSE, 2017).
- **Unambiguous:** There is only one semantic interpretation of a requirement (Halligan, 2012). The requirement is stated in such a way that it can be interpreted in only one way (INCOSE, 2017).
- **Complete:** The inclusion of all necessary information such that if the requirement is met, the need will also be satisfied (Halligan, 2012). The requirement sufficiently describes the necessary capability, characteristic,

constraint, or quality factor in meeting the entity need without needing other information to understand the requirement (INCOSE, 2017). When considering a set of requirements, it must sufficiently describe the necessary capabilities, characteristics, constraints, interfaces, standards, regulations, and/or quality factors to meet the entity needs without needing other information (INCOSE, 2017).

- **Feasible:** The requirement can be realised within entity constraints (e.g., cost, schedule, technical, legal, ethical, regulatory) with acceptable risk (INCOSE, 2017). When considering a set of requirements, it should be possible to realise them within entity constraints with acceptable risk (INCOSE, 2017).
- **Verifiable:** The requirement is structured and worded such that its realisation can be proven (verified) to the customer's satisfaction at the level the requirement exists (INCOSE, 2017).
- **Comprehensible:** The set of requirements must be written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part (INCOSE, 2017).
- **Able to be validated:** It must be able to be proven that the requirement set will lead to the achievement of the entity needs within the constraints (such as cost, schedule, technical, legal and regulatory compliance) (INCOSE, 2017).
- **Attainable:** To be considered as attainable, the requirement must be technically feasible and fit within the budget, schedule and other constraints (Hooks, 1993).
- **Clarity:** Each requirement should express a single thought and be concise and simple. It is essential that the requirement must not be misunderstood – it must be unambiguous (Hooks, 1993).
- **Connectivity:** All the terms within a requirement are adequately linked to other requirements, and word and term definitions, causing each individual requirement to properly relate to each individual other requirement in a set (Halligan, 2012).
- **Modifiability:** The necessary changes to a requirement can be made entirely and consistently (Halligan, 2012).
- **Balance:** A set of requirements forms part of an optimal solution to a higher level problem (Halligan, 2012).
- **Functional orientation:** The set of requirements states what the system is to do, how well it is to do it, and the fundamental external interface characteristics, environmental conditions, constraints and any other required qualities (Halligan, 2012).

- **Implementation independence:** The requirement should be free of design and implementation decisions (Faulk, 1997).

2.4.3.1.6 Types of requirements

Requirements can be grouped into various classes. The following classes of requirements are identified in the literature:

- **Functional requirements:** Functional requirements define what the product or service should be able to do or how it should behave. In certain instances, functional requirements can even explicitly state what the product or service should not do (Faulk, 1997; Sommerville, 2011; Wiegers, 2000).
- **Non-functional requirements:** Non-functional requirements place constraints on the product or service to be designed. The constraints include aspects such as the development process and constraints imposed by applicable standards. Non-functional requirements can further be grouped into product or service requirements, organisational requirements, and external requirements (Sommerville, 2011).
- **Business requirements:** The business requirements define the business case that is driving the product or service development including the benefits for both the end user or customer, as well as the business or organisation (Wiegers, 2000).
- **State or mode requirements:** The state or mode requirements define the required states and/or modes of the item, or the necessary transition between one state and another state, one mode and another mode, or a mode in one state to a mode in another state (PPI, 2007).
- **Performance requirements:** The performance requirement defines *how well* a function is to be performed (PPI, 2007).
- **External interface requirements:** The external interface requirements state the required characteristics at a localised point, region, where the item connects to the outside world (PPI, 2007).
- **Environmental requirements:** The environmental requirements constrain the effect of the external environment (natural or induced) on the development items as well as the effect that the developmental item will have on the environment (PPI, 2007).
- **Resource requirements:** The resource requirements defines maximum usage or consumption of an externally supplied resource (PPI, 2007).
- **Physical requirements:** The physical requirements define the physical characteristics (properties of matter) of the item as a whole (e.g. mass, dimension, volume) (PPI, 2007).

- **Design requirements:** The design requirements direct the design (internal to the item), by inclusion (build this way), or exclusion (do not build it this way) (PPI, 2007).

2.4.3.2 Project problem context

2.4.3.2.1 Introduction

The requirements engineering process does not occur in isolation, devoid from any other influence, but will instead occur within a project environment (Forsberg *et al.*, 2005; INCOSE, 2015). A project can be defined as a temporary endeavour in which members from different disciplines come together for a specific purpose. The project team members may not have worked together before (Forsberg *et al.*, 2005). The results are that the project dynamics will differ from project to project. During the execution of a project, several technical and technical management processes are used to move the project from the initial stages to the final delivery.

The project team members include roles such as project management and systems engineering. Different project implementation models can be used, including the following (Scribante *et al.*, 2017b; Twiss, 1990):

1. **Internal development:** All development activities ranging from the requirements elicitation process through to the final delivery of the project is performed or controlled by the organisation's own staff.
2. **External development:** The complete development process is contracted out to a third party for implementation. In this case, the organisation is typically represented by a project champion.
3. **Mixed development:** The organisation performs the first part of the requirements engineering process itself up to the point when it has a fixed set of requirements. These requirements are sent to external contractors and vendors in the form of a request for proposal.

Irrespective of the project implementation model followed, the starting point for the project is the requirements that define the problem space regarding the 'what' that is to be solved. If the development is performed internally in the organisation, a phased approach can be followed in which the requirements are established during the initial phases of the project. These requirements can serve as a basis for budgeting and planning for the implementation.

If the development is performed by an external organisation or in a mixed development format, the basis of the contract is typically either a fixed price or a cost-plus type contract. In this case, it is essential that the scope of the development effort already be quantified to such an extent that the external contractor can do a proper costing without adding a significant risk portion to the price due to unknown issues. These uncertainties regarding cost, performance and schedule can also result in a failed or challenged project if not addressed appropriately.

2.4.3.2.1.1 Project environment as a complex environment

The conclusion from the preceding discussion is that the project environment can be considered as complex. Complexity is an integral part of project management and systems engineering and one of the core reasons for the existence of the systems engineering discipline (Sheard, 2013).

Complex systems exhibit a certain degree of self-organisation. This self-organisation gives rise to a novel, emergent behaviour that is not evident in the individual parts (Bar-Yam, 1997; INCOSE, 2015). An additional identifying characteristic of complex systems is that they are purposeful as they have a definable objective and function (Bar-Yam, 1997). To understand the behaviour of a complex system, one must understand the behaviour of the individual parts and their interaction with one another. It is thus not possible to describe the whole without describing the individual elements, and it is not possible to describe the individual elements without describing their interaction (Bar-Yam, 1997).

Complicated systems or processes stand in contrast to complex systems. A complicated system or process exist if the relationship between the different parts is based on improved or fixed relationships. These fixed relationships allow for reasonable predictions of the time, cost and technical resources required to complete the process (INCOSE, 2015). Table 9 identifies certain of the properties of complex systems.

Table 9: Central properties of complex systems (own contribution)

No.	Property
1	Number of elements
2	Interactions between the different elements and the relative strength
3	Formation and operation including the timescales
4	Diversity and variability
5	Environment and its demands
6	Activities and their objectives

Bar-Yam (1997) identified two approaches to organise the properties of complex systems when they are being studied. The first approach is to consider the relationships between the elements, the parts, and the whole.

1. The parts of a complex system can be themselves complex.
2. The system can consist of simple parts but exhibit complex behaviour (emergent complexity).
3. The system can consist of complex parts but exhibit simple behaviour (emergent simplicity).

A graphical representation of the characteristics of complex systems according to Sheard and Mostashari (2009) is shown in Figure 29.

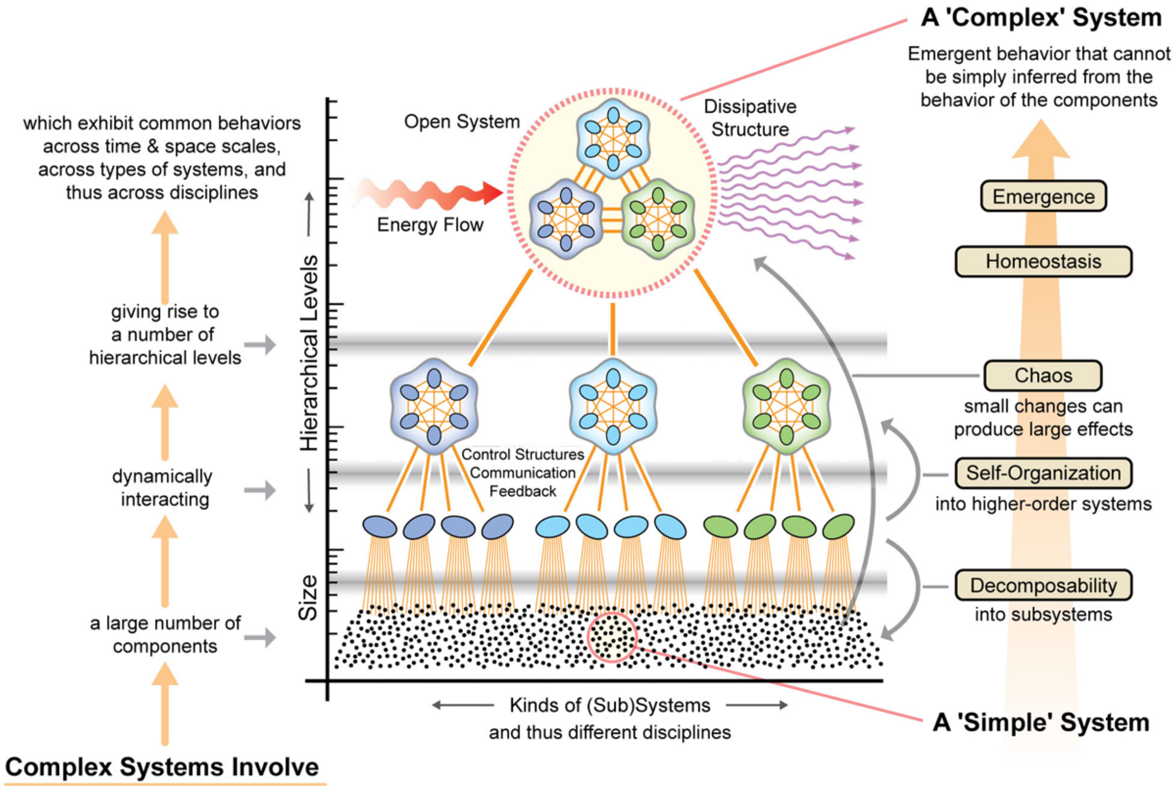


Figure 29: Characteristics of complex systems (Sheard *et al.*, 2009)

The second approach is to start by understanding the relationship of the systems with their individual descriptions.

A high-level measurement for the complexity of a system is to look at the amount of information that is required to describe the system (Bar-Yam, 1997).

Complexity comes in many forms and ranges from project type, the number of stakeholders, etc. (Sheard, 2013). Sheard (2013) identified five factors that can create complexity in a system as shown in Table 10.

Table 10: Factors contributing to complexity in a system (Sheard, 2013)

No.	Description
1	The (technological) system that is being designed and built
2	The (socio-technical) project doing the building
3	The technological environment into which the system will be required to operate once being built
4	The socio-political system related to the technological environment (system stakeholders)
5	The subjective human experience when thinking about, designing, or using the system

Three of the five factors identified include the concept of human involvement in the design, specification and operation of the system. If a human is involved in a system, it tends to move immediately from a being complicated system to be a complex system. The concept of complexity in the project execution environment arises from different causes, including the maturity of the organisation, the number and type of stakeholders, and the required quality of the resulting solution (Sheard, 2013). Research published suggests that there may as many as 32 complexity types spread over 12 disciplines and domains (Young *et al.*, 2010). One of the lesser recognised areas of complexity is the result of the organisation that needs the solution, the organisation creating the solution, and the created solutions all being socio-technical systems.

Midgley (2016) had a different view of what complexity entails. He identified four domains of complexity that consist of the following aspects:

- Natural world complexity describes the complexity of “what is”.
- Social world complexity describes the complexity of “what should be”.
- Subjective world complexity describes the complexity based on what an individual is thinking, intending or feeling.
- Metal-level complexity combines the complexity of the first three complexity types.

When attempting to model a complex system, the best that one can hope for is an approximation of the complex system. This can be attributed to the so-called “darkness principle” that postulates that “*no system can be completely known*” [Skyttner (2001) as quoted by Richardson (2004)]. This can be seen as a direct consequence of the inherent non-linearity of complex systems and that each of the elements in a complex system is ignorant of the behaviour of the overall system in such a way that it only acts on the information available locally (Richardson, 2004).

2.4.3.2.1.2 Project environment as a socio-technical environment

The project and system engineering environment can be considered to be a socio-technical environment, and that behaves as a socio-technical system. In such a social-technical system the concept of socio-technical thinking must also be considered. Socio-technical thinking can be defined as a systematic knowledge of the relationships between technical objects, the natural environment, and social practices (Ropohl, 1997). The original concept of socio-technical thinking and systems is credited to the research work done by the Tavistock Institute in the British coal mining industry following the post-war reconstruction (Trist, 1981). During this period, an increase in mechanisation was introduced to improve productivity. However, the result was not an increase in productivity, but rather a sharp decrease. The turnover of workers in the coal mines was high, and the morale of the remaining few workers low. While researching the problem, the Tavistock Institute found one mine that was not affected by these problems. The key difference was how the workers had organised themselves to perform the work using the higher level of mechanisation while improving the workers' cohesion and participating in decisions regarding the working arrangements (Trist, 1981).

Ropohl (1997) identified five types of social-technical knowledge, summarised in Table 11.

Table 11: Five types of social-technical knowledge (Ropohl, 1997)

No.	Description
1	Technical know-how
2	Functional rules
3	Structural rules
4	Technological laws
5	Socio-technical understanding

The conclusion that can be drawn from this is that complexity is part of the day-to-day environment that system engineers must deal with. In many cases, complexity is the reason for the need to apply systems engineering principles in the first place. This complexity arises in many situations – not just from the technology used in the process, but also from the humans who are involved in different roles within the process. Ropohl (1997) remarks that, *"… every invention is an intervention: an intervention into nature and society*". The result of this is therefore that the technical development strategies that are employed must now be expanded to include socio-technical awareness (Ropohl, 1997).

It was previously defined that a socio-technical system is created in which two jointly independent systems, the social and the technical, interact in a correlative manner to produce a single outcome (Bostrom & Heinen, 1977). In such a socio-technical system, the technical system is involved with the various processes, different tasks and technologies required. The social system relates to the different attributes of the people, the relationship between the various stakeholders, and the reward systems that are present in such an organisation, as well as the reporting and authority structure present (Bostrom *et al.*, 1977b). Such a socio-technical system has a high level of dynamic complexity. This dynamic complexity is a result of the interactions of the stakeholders over time, including time delays between deciding and implementing, or the delay in observing a problem situation and correcting it. The effect of this complexity is that the learning loop is slowed down, thus reducing the amount of improvement that can be achieved within a given period (Sterman, 2000).

2.4.3.2.2 Stakeholders and role players

If several stakeholders have been involved in the process of requirements elicitation, they should have a common goal to achieve a successful outcome and conclusion of the project. Nonetheless, it is inevitable that conflict will arise among these stakeholders. On the one hand, these stakeholders are representatives of the end user, decision maker or group and, as such, they have a mandate that may dictate specific concerns, priorities and responsibilities. However, as individuals, they have their own perspectives and perceptions of what is required and what is essential (Ahmad, 2008). If there is conflict of interest among stakeholders and even within a stakeholder, it has been proposed that negotiation strategies may be required to enhance stakeholder collaboration in order to arrive at the end goal of a successful project (Ahmad, 2008).

2.4.3.2.2.1 Stakeholder characteristics

Stakeholders have specific inherent characteristics that have a direct influence on the elicitation of the correct requirements. These characteristics are the following:

- **Domain knowledge:** The stakeholders require specific domain knowledge to participate and add value in the requirements elicitation process. While it may seem obvious that the stakeholder has the domain knowledge required, it may not always be so for various reasons, including organisational politics or hierarchical standing within the organisation (Scribante *et al.*, 2015).
- **Technical knowledge:** The stakeholders also require a basis of technical knowledge. This technical knowledge includes the fundamentals of the various technologies that are in play in the domain and knowledge of items such as development methodologies and tools that can be used (Anwar *et*

al., 2012). The domain knowledge together with the technical knowledge support the understanding of the need by the stakeholder.

- **Stakeholder or customer introduced misinformation:** Depending on the organisational structure, it may happen that the person, or the team, who is driving the requirements elicitation process may not have access to, or include the actual end user ("the guy on the ground") who will be using or operating the system. This type of situation is more likely to occur in organisations where a formal hierarchical structure exists, such as a large corporate/government organisation or military organisation. In such a case, the external specialist or consultant may only interact with, or have access to middle or senior management who may have a perception of what the needs are, but do not have actual first-hand experience (Scribante *et al.*, 2015).

  An additional risk area that the consultant or analyst should avoid is a situation in which certain representatives of the customer may be trying to influence the outcome of the requirements elicitation process. This situation typically occurs when the representative is trying to favour a specific solution during the requirements elicitation process (Scribante *et al.*, 2015).

  In order to counter this type of situation, the external specialist or consultant should ensure that they are familiar with the business of the organisation, as was described in the previous section. This familiarity should enable the external specialist or consultant to identify all the relevant stakeholders or groups of stakeholders and ensure that they are included in the requirements elicitation process. Techniques that allow for the cross-verification of stated requirements should also be used so as to try and eliminate ambiguities and identify manipulated requirements (Scribante *et al.*, 2015).

  The ideal situation is that the actual end user forms part of the stakeholder group that the requirements engineer interacts with. Depending on the organisational structure within the stakeholders' organisation, it may, however, happen that the requirements engineer does not have access to this group. In such a case, the actual end user may be represented by a specialist or manager, who may have their own perception of what the need is that must be solved. This perception can, however, be widely different from the actual need (Scribante *et al.*, 2015).

- **The role of a stakeholder's dual nature in requirements elicitation:** Ahmad (2008) identified a duality in the nature of stakeholders. On the one hand, as individuals, they each have their own perspectives and perceptions of what the need is that must be solved. On the other hand, as stakeholders or representatives of the stakeholders, they may have different concerns, priorities and responsibilities.

The human context within which the delivered solution will operate also plays a fundamental role in requirements elicited so as to ensure that the solution is adopted and the resistance to change is kept to a minimum (Fuentes-Fernández, Gómez-Sanz and Pavón, 2010).

It has already been identified previously that the need that must be satisfied is represented by the various stakeholders. Ahmad (2008) stated that when dealing with these stakeholders, it is inevitable that conflict will occur since each stakeholder has their own perspectives and perceptions as individuals of what the need is. However, as stakeholders and, thus, as representatives or surrogates of the end users, they may have different concerns, priorities and responsibilities. This duality in the nature of being a stakeholder is shown in Figure 30.

Fuentes-Fernández *et al.* (2010) identified that the human context within which a system will operate is fundamental to its requirements. While this may not seem to be related to the requirements of the system, it may, however, play a significant role in achieving its successful adoption of the system. Fuentes-Fernández *et al.* (2010) further identified that a gap may exist in the skill set of those who are performing the requirements elicitation as they may instead have a background in a technical discipline and may not be trained to elicit this kind of information.

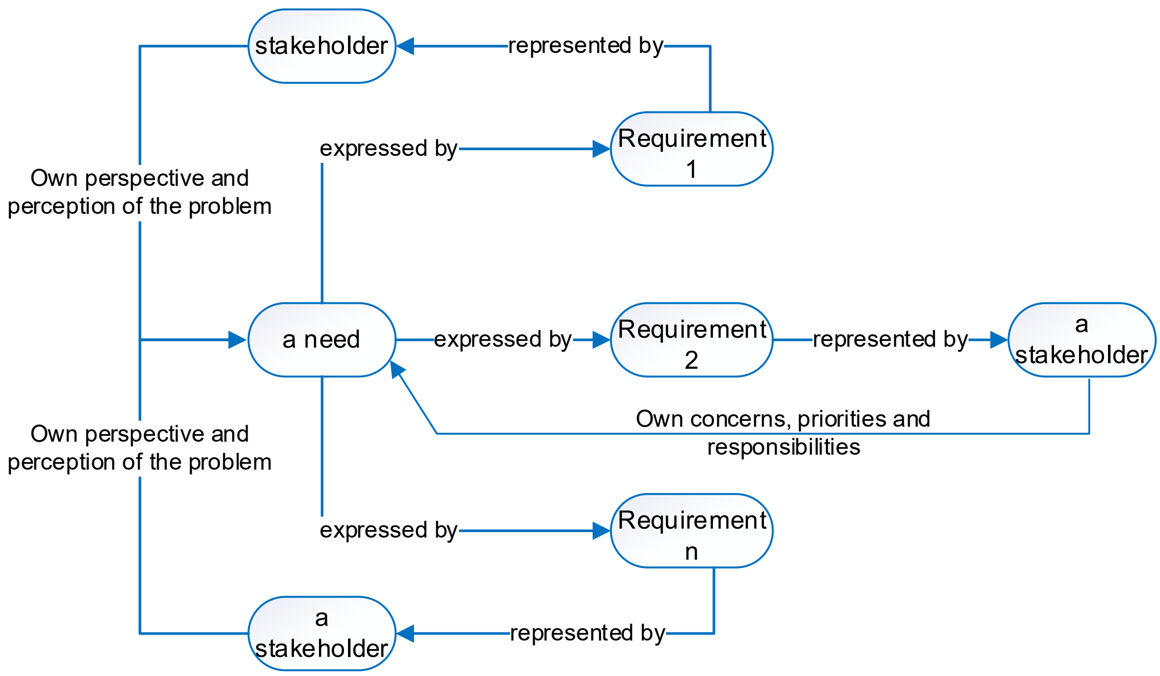


Figure 30: Duality in the nature of the stakeholder [constructed from Ahmad (2008)]

- **Resistance to change:** Resistance to change can occur among the stakeholders in an organisation for various reasons, including interference with need fulfilment, selective perception, habit, inconvenience or loss of freedom, security in the past, fear of the unknown, threats to power or influence, knowledge and skill obsolescence, organisational structure, and limited resources (Yilmaz and Kılıçoğlu, 2013). This type of behaviour can again lead to the stakeholder attempting to manipulate the requirements engineering process in order to avoid a new system, product or process that is different from the existing or legacy system.

- **Legacy system knowledge:** If a system incorporates legacy requirements, the stakeholder will be required to have the necessary knowledge to support the requirements elicitation process in this regard.

2.4.3.2.2.2 Requirements engineer characteristics

The requirements engineer has similar inherent characteristics as the stakeholder, which has a direct influence on the elicitation of the actual requirements. These characteristics are the following:

- **Domain knowledge:** In order to increase the quality of the elicited requirements, the requirements engineer requires a certain amount of knowledge of the nature of the business and the needs that must be solved. While the requirements engineer may be able to get more in-depth knowledge about the business by interviewing the relevant people, this may not be enough in many cases. The level of familiarity that the requirements engineer requires of the business does not only include familiarity with the operation of the business but also extends to domain knowledge that describes the nature and culture of the organisation (including specific terminology and abbreviations that may be used within the organisation).

  The requirements engineer can take various actions to improve their domain knowledge. A study done by Hadar, Soffer, and Kenzi (2014) identified several recommendations for the requirements engineer, including that requirements engineers who lack domain knowledge must learn the domain terminology before starting with requirements elicitation sessions. They should, in addition, engage support within the organisation for the purposes of both obtaining a complete understanding of the customer's needs and communicating with the customer during the elicitation process.

  An alternative method that the requirements engineer may use to enhance their understanding of the nature of the client's business is described by Kaiya and Saeki (2006). They suggested that an external specialist or consultant describes the domain knowledge in terms of a domain ontology.

  Hadar *et al.* (2014) found that requirements engineers who do have the necessary domain knowledge should avoid fixation and preconceptions that may lead to an incomplete and inaccurate understanding of the customer's needs.

- **Technical knowledge:** The requirements engineer, like the stakeholders, also requires a basis of technical knowledge. This technical knowledge includes the fundamentals of the various technologies that are in play in the domain and knowledge of items such as development methodologies and tools that can be used (Anwar *et al.*, 2012). The domain knowledge, together with the technical knowledge, supports the understanding of the need by the requirements engineer.
- **Requirements engineer introduced misinformation:** Another source of error that may influence the accuracy of the requirements elicited from stakeholders is the so-called "*requirements engineer-induced misinformation effect*", as described by Appan and Browne (2012), that defines mis-information as distorted, false, or other erroneous and misleading information. In their paper, Appan and Browne (2012) identified that this misinformation effect may lead the user to recall misinformation that may have been introduced by the requirements engineer rather than their true beliefs and knowledge of the fact. The overall effect of this misinformation is to reduce the correctness of the requirements elicited.

  In order to reduce the chance of requirements engineer-induced misinformation, Appan and Browne (2012) recognised that the choice of elicitation technique is essential, given the relative strengths and weaknesses of interviews and surveys to yield accurate information. If the requirements engineer does decide to use an interview as an elicitation technique, they must take care to remain neutral during the requirements elicitation process to reduce the so-called "demand effect". This demand effect relates to people being interviewed tending to respond with an answer that they think the person conducting the interview wants to hear, rather than answering from fact.
- **Requirements engineer experience:** The experience of the requirements engineer must also be such that the person is able to analyse the situation and select the correct requirements elicitation method that will yield the best results (Scribante *et al.*, 2015).
- **Requirements engineer personality:** The personality of the requirements engineer must enable them to interact with the stakeholders to elicit requirements effectively (Scribante *et al.*, 2015). This will include having conflict management skills, having communication skills, and creating a trusted environment within which to work.

2.4.3.2.2.3 Stakeholder and requirements engineer interaction

The interaction and relationship between the stakeholder and the requirements engineer are vital for the successful elicitation of requirements within an organisation. The following aspects form part of this interaction and relationship:

- **The social nature of requirements elicitation:** The social nature of requirements elicitation was recognised by Chakraborty, Sarker and Sarker (2010). They identified the social nature inherent in the requirements elicitation process based upon the collaboration between the various stakeholders and the requirements engineer. During this collaboration process, knowledge regarding the needs and the requirements is shared and discussed to create a shared understanding. It is essential that a trust relationship is maintained to ensure successful collaboration.
- **Communication between stakeholder and requirements engineer:** Accurate communication between stakeholders and the analyst is of crucial importance to ensure that the need and the individual requirements of the stakeholders are understood. This communication channel can be interrupted due to various reasons, including:
    - Lack of a standard dictionary or ontology to define the concepts
    - Different first or home languages
    - Different levels of domain knowledge of the subject at hand (Aranda, Vizcaíno and Piattini, 2010)

  Communication plays a vital role in the requirements elicitation process – whether communication is in verbal or written format. In addition to the standard communication aspect inherent in the human and social nature of requirements elicitation, additional problems may be encountered if such an exercise is conducted over national and internal borders by multi-national teams. In these type of project, the requirements engineer may not only have to deal with a lack of face-to-face communication but also with issues such as different time zones and cultural diversity. These challenges may lead to misunderstanding and even conflict in the process (Aranda *et al.*, 2010).

  Communication may further be impeded in multi-national projects if the native language of the various stakeholders and the requirements engineer are not the same. This could lead to a misunderstanding of the questions or the discussion points during the requirements elicitation process. The result is that incorrect, incomplete or ambiguous requirements are elicited (Scribante *et al.*, 2015).
- **Conflict:** Conflict is a common occurrence when group interaction is present. Aspects that may introduce conflict in the requirements elicitation process

most likely occur when there is limited domain knowledge present, fluctuating or conflicting requirements, and a breakdown in communication and coordination (Easterbrook, 1991b).

Conflict must be resolved as part of the requirements elicitation process. In a worst-case scenario, suppression of conflict may lead to the requirements elicitation process breaking down or stakeholders withdrawing (Easterbrook, 1991b). Conflict can, however, play a decisive role in the requirements elicitation process as it can be used as a tool to counter resistance to change and to counter stagnation in the elicitation process (Easterbrook, 1991b).

- **Trust:** Trust can be defined as the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control the party (Hengstler, 2016).
- **Cultural difference:** In a multi-cultural organisation or when suppliers and customers come from different cultural backgrounds, misunderstanding regarding the need and requirements can occur. Aranda *et al.* (2010) identified that in such instances, misunderstandings can be caused by different factors such as using ambiguous words, expressions and even body language.

2.4.3.2.3 Project characteristics

The following project-related characteristics can be identified:

- **Project type:** The type of the project being undertaken has a direct influence on the complexity of a project. For example, whether a bespoke system or an off-the-shelf solution is required.
- **Project size:** The size of the project in terms of value, geographical distribution, and the number and type of requirements to be elicited.
- **Project complexity:** The complexity of the project being undertaken has a direct influence on the number and type of requirements to be elicited, which in turn can lead to duplicated or missed requirements. For example, a system requiring critical safety features vs a commercial application.
- **The number of stakeholders:** The number of stakeholders who are involved has a direct influence on the number and type of requirements to be elicited, which in turn can lead to duplicated or missed requirements (Sommerville *et al.*, 2003).
- **Existing or legacy requirements:** In many cases, the purpose of a new system is to either upgrade or replace an existing system to adapt to new business challenges or changes in the technological landscape. In these

cases, it is crucial to ‘rediscover’ the original or legacy requirements of the existing system.

Rayson, Garside and Sawyer (1999) identified a potential risk of critical requirements that are implicit in legacy systems going unsupported in new or upgraded systems. They identified a further risk that may occur if specific functionality inherent in the existing system is included in the new system without the requirement driving this functionality being known. This risk exists because business change often takes place against a background of poor organisational memory (Rayson *et al.*, 1999).

If the external specialist or consultant fails to consider legacy requirements, this can typically lead to defects in terms of the completeness and consistency of the elicited requirements.

- **Project constraints:** Kerzner (2013) identified the constraints of time (project duration), cost (project budget), and technical performance within which a project must be executed. These elements act as constraints on the project and play an essential role in the requirements engineering process.
- **Changes in technology:** Changes in technology have been identified as one of the significant challenges in the requirements engineering/elicitation process. This is due to the domain and technical knowledge not being relevant or applicable anymore and solutions that worked previously no longer being relevant (Berenbach *et al.*, 2009).
- **Requirements volatility:** Requirements volatility can be defined as the change in the requirements set due to missed requirements, changing requirements or duplicate requirements. This volatility can affect the quality of the requirements elicited and may directly affect the success of the project (Coulin, 2007; Verner *et al.*, 2008; Zowghi *et al.*, 2005).

2.4.3.3 The object-of-study for the requirements engineering research tool

Paragraph 1.4.1.6 and Scribante *et al.* (2018a) discussed the concept of a modified object-of-study that can be used when a research instrument or research tool is being designed. The requirements engineering problem context consists of the requirements engineering artefact that interacts with the project environment to produce requirements. Figure 31 shows the object-of-study for the research tool. This research tool object-of-study shows that the original artefact and context are now combined to become the new context. The research tool becomes the new artefact that interacts with this new context to improve something. That something is the improvements in the overall requirements engineering process.

2.4.3.4 Requirements volatility as a measure of project requirements health

In an ideal world, one would expect to be able to elicit a stable set of requirements that will not change over the lifespan of the project. In the real world, requirements elicitation is a learning rather than a gathering process (Reifer, 2000). Requirements change due to various factors. These include errors in the original requirements, evolving customer needs, changes in the business environment or company organisational policies, incorrect stakeholder selection, the use of inappropriate elicitation techniques, stakeholder characteristics, and stakeholders changing their understanding of the need.

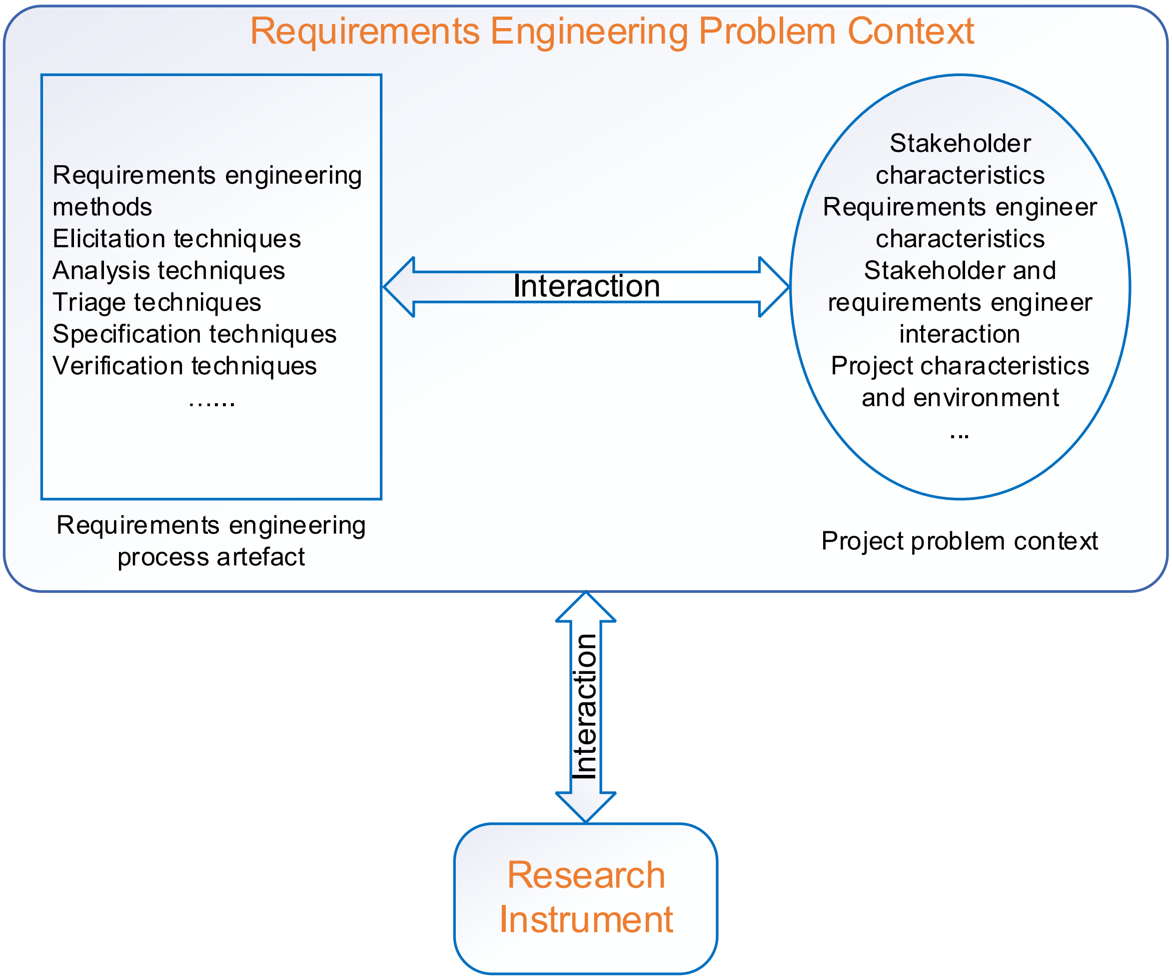


Figure 31: The redefined object-of-study for a requirements engineering research tool (own contribution)

Requirements volatility can be defined as the growth or change in requirements during a project development life cycle. Requirements volatility is also known by various other terms in the literature including requirements evolution and requirements creep. Requirements volatility can be measured in terms of the number of requirement additions, deletions or modifications over a specific period of time (Ferreira, Collofello, Shunk and Mackulak, 2009).

Javed, Maqsood and Durrani (2004) emphasised that the problem in development projects is not as such that requirements do change, but rather that there are not adequate processes for dealing with this change.

The requirements elicited from the human element tend to be more unstable or volatile initially but should become more stable as the requirements engineering process progresses. However, if the requirements elicitation phase is not managed correctly, this volatility could result in an incorrect set of requirements being used for the subsequent phases of project delivery, which in the end, could lead to a failed project. One way of understanding this requirements volatility and the potential adverse effect that it can have on the final set of requirements is by examining the various interrelationships and factors that are involved in the requirements elicitation process. One possible way is to model these factors in the form of a system dynamics model that will show the interactions of the various elements and the effect that each of these factors has on the stability of the final set of requirements.

2.5 Chapter summary and conclusion

The purpose of this chapter in the thesis was to identify the research problem and the context within which the research will occur. It was identified that the research problem occurs within a complex socio-technical environment. Within such an environment it is not always possible to find a solution that will work completely. It is more feasible to treat the problem as an improvement problem and work to find aspects that can be made better to improve the overall situation.

Applicable theoretical and conceptual frameworks were identified to support the research effort in establishing why the research is relevant and why the research process is rigorous.

The research environment was characterised by establishing the research stakeholder groups and the high-level research operational concept. The object-of-study was defined in detail in terms of the requirements engineering process artefact and the problem context. The redefined object-of-study for the research tool was also introduced and described.

Chapter 3 will describe and discuss the specification phase of the research tool. The focus of Chapter 3 will be to move from the stakeholder’s problem domain to the research tool solution domain.

CHAPTER 3. RESEARCH PROBLEM DEFINITION PHASE

"If I had an hour to solve a problem, I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions." – Albert Einstein

3.1 Introduction

The research problem definition phase is the second phase in the research process to be addressed in this thesis. The relative position of this phase in the overall research process is shown in Figure 32.

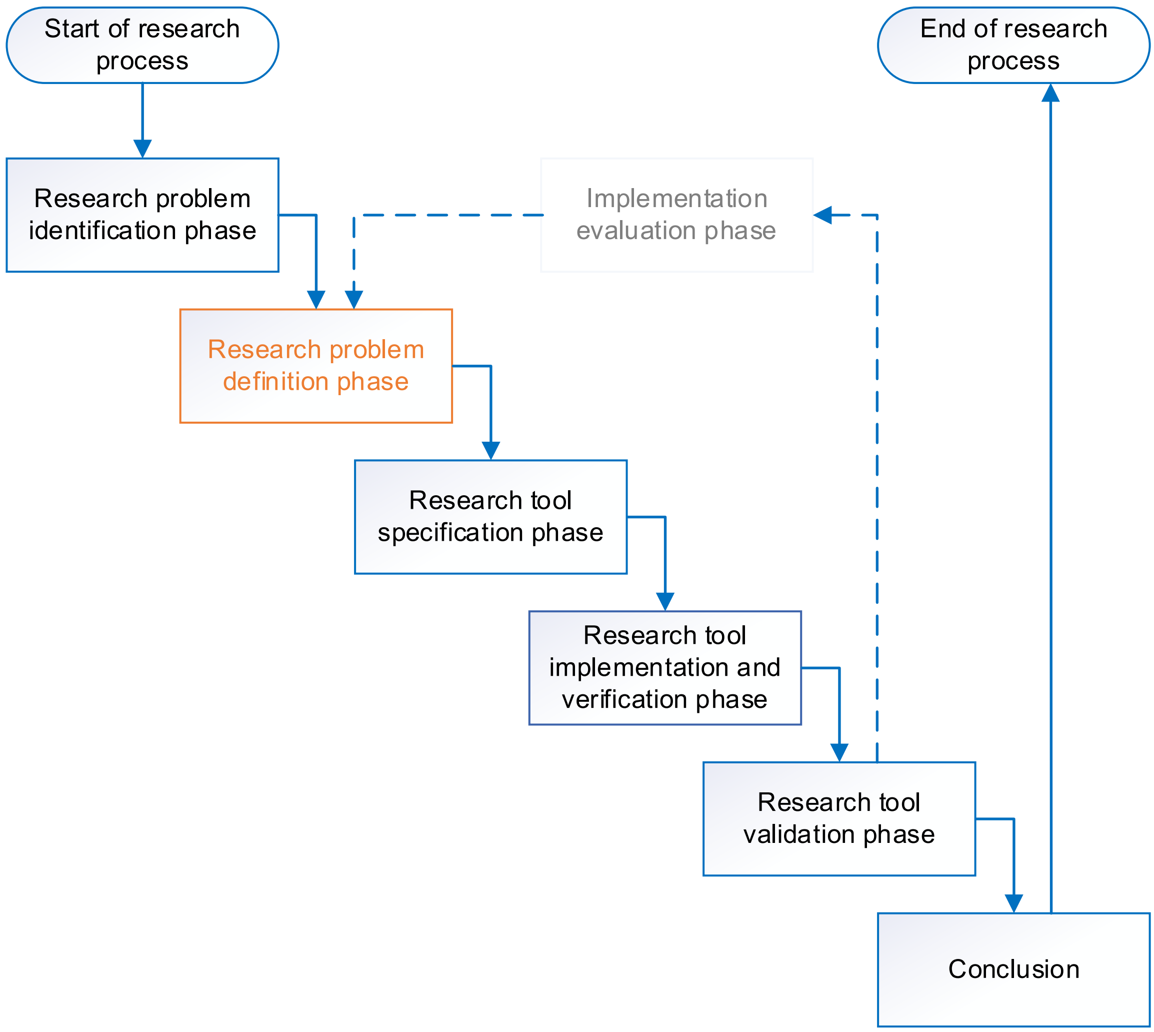


Figure 32: Relative location of the research problem definition phase in the overall research process

The primary objective of the research problem definition phase, as was discussed in paragraph 1.4.2, is to arrive at the set of stakeholder requirements that will form the basis for designing and implementing the research tool. This will be done by analysing the research problem in terms of (a) the stakeholders and their goals;

(b) a description of the observed phenomena; (c) a conceptual framework; (d) the identified knowledge and research questions; (e) an identification of the population of organisations and SMEs from which a sample will be selected; and (f) the research operational concept. These activities take place in the problem domain when looking at the problem to be solved from the viewpoint of the stakeholder and are shown in the form of an FFBD in Figure 33.

The research problem definition can be equated in purpose and process to the stakeholder needs and requirements definition process as is defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

3.2 Knowledge questions

Knowledge is created by answering knowledge questions. The outcome of knowledge questions does not result in a fundamental change in the world but instead searches for knowledge about the world as it is at present (Wieringa, 2014).

From the discussion in the preceding paragraphs, one can identify the following knowledge and research questions:

1. What are the observed phenomena?
2. What are the potential causes of these phenomena?
3. What are the mechanisms that cause these effects and what system components produced this event?
4. What are the contributions of these phenomena to the observed problems?
5. Is it possible to identify any specific sensitivity to any of the parameters?
6. Is there any outside motivation for the stakeholders or organisations that could explain the behaviour?

3.3 Research stakeholders

Research stakeholders are one of the most significant elements to be considered when defining a research problem. Each research stakeholder will have specific goals and desires. In certain cases, there may even be a conflict between the goals and desires of the different stakeholders (Ahmad, 2008; Glinz *et al.*, 2007; Laplante, 2009; Wieringa, 2014).

A stakeholder can broadly be defined as any individual, groups of individuals, or organisation that have a vital interest in the actions of the business or the outcome of the project (Pearce II *et al.*, 1991; Sharp, Finkelstein and Galal, 1999; Wieringa, 2014). The role of the stakeholders is to define the goals of the system and in doing so, to define the requirements of the system (Wieringa, 2014).

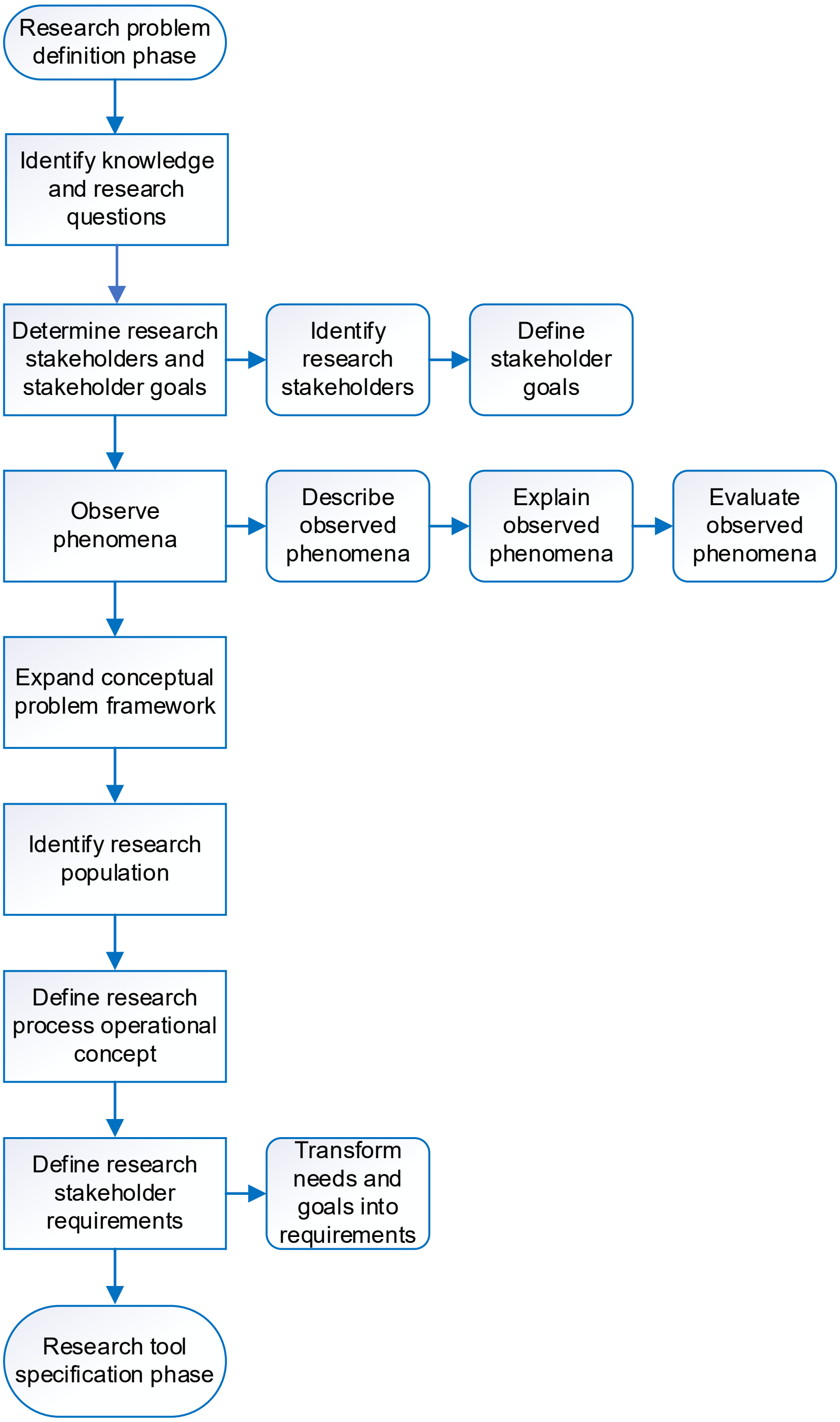


Figure 33: Research problem definition phase FFBD

In the context of this thesis, the term 'business' refers to the application of the research tool to the problem domain.

Alexander (2005) identified three classes of stakeholders, namely, those involved in operating the system, those involved in the containing system, and those involved in the broader system. These classes have been discussed in appropriate detail in paragraph 2.4.1. Based on these three classes, the following stakeholders as shown in Table 12 can be identified in the context of this study.

Table 12: Requirements engineering research tool stakeholders

No.	Classes of stakeholders	Stakeholder examples
1	The system	Researcher, SMEs.
2	Containing system	Other researchers and practitioners, sponsors of the research project.
3	Wider system	Project managers, system engineers and quality assurance managers involved in delivering projects; university management interested in the research output.

The researcher and the SMEs are directly involved in the operation of the research tool and are as such the primary stakeholders. Other researchers, practitioners and any sponsors of the research project are grouped to form the containing system. These stakeholders are interested in the new knowledge produced that may be applicable in their situation. The stakeholders who are grouped on the broader system include project managers, systems engineers and even the management of the university. The interest of these stakeholders in the research product is more related to the benefits that the research produces, including optimised processes, research papers published, and citations generated.

3.4 Determine research stakeholder goals and needs

The research stakeholder goals are expressed by their undiscovered needs and desires. A need in this context can be defined as a condition requiring relief, and a goal is a destination where one wants to end up at.

The undiscovered needs and desires identified here will later be transformed into unstructured requirements via a requirements elicitation process. The following goals are defined per class of stakeholder:

3.4.1 Stakeholders who are involved in the system

The primary stakeholders who are directly involved in the system have been identified as the researcher performing this study and the SMEs who are interviewed as part of the single-case mechanism research experiment.

The need of the researcher in the context of this study is to establish an in-depth understanding of the requirements engineering process. The goal of the researcher is to design a tool that can be used to perform research in the requirements engineering environment so as fulfil the need.

The goals of the SMEs are to ensure that their experiences are conveyed accurately so that the tool can have value within their field of expertise and, in the process, make their job easier, more productive and satisfying (Williams D. and Kennedy, 2000).

3.4.2 Stakeholders who are involved in the containing system

The stakeholders who are involved in the containing system have been identified as other researchers, other requirements engineering practitioners, and sponsors of the research project.

The goals of other researchers and requirements engineering practitioners are to have either a requirements engineering research tool available that they can use to perform further research in the field of requirements engineering or a tool that can be used in practice to improve the requirements engineering process within a specific project.

The goals of the sponsor of the project (if any) can be identified as being comparable to those of other researchers (an academic institution is sponsoring the research project) or the requirements engineering practitioner (an organisation is sponsoring the research project). A further goal of the project sponsor could be a future scenario in which a project or a system is delivered on time, within budget and with the fewest errors (Kerzner, 2013; Williams D. *et al.*, 2000).

3.4.3 Stakeholders who are involved in the broader system

The stakeholders who are involved in the broader system have been identified as systems engineers and project managers who are involved in delivering other projects. The goals of these stakeholders are like those of the requirements engineering practitioners as described in paragraph 3.4.2.

3.5 Observed phenomena

A project is a temporary endeavour that brings together a multi-disciplinary team to produce deliverable items, whether it is in the form of hardware, software, services, or other items (Forsberg *et al.*, 2005). These deliverable items can either be in the form of new items that are the result of a radical innovation process or enhanced deliverable items that are the result of incremental innovations (Burgelman *et al.*, 1988). The execution of a project takes place within the constraints of time, costs, and performance (Kerzner, 2013). Kerzner (2013) identified a fourth constraint in the form of good customer relations for a project executed for an external customer.

Certain of the observed phenomena have already been discussed as part of the current knowledge in paragraph 2.3.2. The most visible phenomena mainly relate to projects not achieving their set goals (normally related to schedule, cost, and technical performance) due to a wide range of reasons, including aspects that can be linked in some form or other to the requirements engineering process. The effect of these failures is to push the project up against one of the four constraints identified previously, leading to compromises having to be made in terms of the schedule, budget, technical performance or the satisfaction of the customer.

3.5.1 Description of the observed phenomena

The main observed phenomena are a failure of the project being executed at the time. Failure can have a different meaning for different stakeholders. For the purposes of this discussion, we use the Standish group's definition that distinguishes between a successful project, a challenged project and a failed project (Hastie *et al.*, 2015; Johnson and Mulder, 2016). In terms of this definition, a successful project is one that has been completed on time and within the set budget, and that achieved the specified functionality and features all within good customer relations. A challenged project is a project that has been completed and is operational but has exceeded the allocated time and allocated budget. It may further offer fewer functions and features than originally contracted. A failed project can be classified as one that is cancelled during the delivery period (Attarzadeh *et al.*, 2008).

Williams T. *et al.* (2012) identified several early warning signs that may indicate that a project is heading towards being challenged or failed. These warning signs are shown in Table 13.

Table 13: Early warning signs of potential problems in a project (Williams T. *et al.*, 2012)

Project setup	Early stages	Project execution
• Sponsor(s) with unclear role • Lack of an implemented governance framework • Poor project definition • Lack of clarity in rationale, goals, and benefits • A poorly developed business plan • Poor definition of scale and resources that are needed • Unclear what assumptions are valid about the project • Lack of relevance of the proposed solution compared with the needs • The need for the development of new technology • Main risks not identified • Sponsor(s) having unclear expectations • Vague or unclear reasons for undertaking the project (unclear thinking) • Needs considered not real • Inconsistent arguments about agendas • Uneasy comments and body language • The way questions are asked and how answers are given • Specific conditions exist that will make cultural aspects important	• Lack of a good business case • Deterioration of relations between the participants • Lack of a common definition of roles and responsibility • The project team over-relying on the consultant/ contractor's people to "fix it" • Numbers/information missing in documents • Assessments not performed • Documentation not completed • Inappropriate quality of information and documentation produced • Missing competence in the project team • Guidelines for early phase assessments and "behaviour" not followed • Disputed major decisions and complications arising from these • Main risks not identified • Leadership issues • The way answers are given to critical questions when the answers are vague • Strained atmosphere • Lack of a culture of openness and good communication between the actors • Confusing or wavering changes in position over time • Uneasy comments and body language • Stating uncertainty, unwillingness to conclude • Parties unwilling to share relevant information • Parties voicing reservations and politically hedging their positions	• People in "acting positions" with no authority to recommend action • Lack of documentation • An excess of "no cost/no time" effects leading to optimism bias • Contractor unfamiliar with domain responsibility • High level of sub-contractors' claims and extension of time claims • Plans and reports too late and/or not clear • Contract obligations not fulfilled • Milestones/activity definitions unclear or missing • Missing competence in the project team • Remaining risks not identified • Leadership issues • Lack of commitment to make decisions • Frequently changing decisions • Continually unfulfilled promises • Vague answers to critical questions • When people work too much or too little • Uneasy comments and body language • Not showing trust in the project organisation

3.5.2 Explanation of the observed phenomena

The occurrence of the observed phenomena can be explained in terms of the interaction between various characteristics that form part of the complex project environment and the object-of-study. These characteristics were discussed in detail in paragraph 2.4.3 and are summarised in Table 14.

Table 14: Requirements engineering process artefact and project problem context characteristics [Adapted from Scribante et al. (2016c)]

Stakeholder characteristics • Domain knowledge • Technical knowledge • Stakeholder or customer introduced misinformation • The role of the stakeholder's dual nature in requirements elicitation • Resistance to change • Legacy system knowledge	**Stakeholder and requirements engineer interaction** • Communication between stakeholder and requirements engineer • Cultural differences • Social nature of requirements elicitation • Conflict • Trust
Requirements engineer characteristics • Domain knowledge • Technical knowledge • Requirements engineer introduced misinformation • Requirements engineer experience • Requirements engineer personality	**Project characteristics** • Project type (developmental, bespoke/customised) • Project size • Project complexity • Number of stakeholders • Number of requirements • Existing or legacy system requirements • Project constraints consisting of: o Project duration o Project budget o Technical performance • Changes in technology • Requirements volatility

In order to identify the architectural structures that are present in the object-of-study, a causal loop diagram is constructed to identify the mechanisms and causes of the observed phenomena as well as the underlying interrelationship between the different elements(Kim, 2017; Sterman, 2017). This causal loop diagram is shown in Figure 34. This causal loop diagram is constructed relative to the contribution of the various parameters to the requirements volatility. The purpose of a causal loop diagram is to identify feedback structures within the system being examined (Sterman, 2000).

The structure of the causal loop diagram is an example of what a typical diagram will look like and what elements it may contain. It is foreseeable that this diagram may vary depending on the exact project situations in which more or fewer elements may be identified and as a result be included in the diagram (Goodman, 2017)good.

A causal loop diagram details the interaction between different elements of the system with the arrows and the (+) or (–) signs indicating the effect of one parameter on the following parameter. An example from Figure 34 is the interaction between the conflict element and the communication between stakeholders' and requirements engineer element. In this case, a minus sign is shown at the end of the arrow. This indicates that as conflict increases in the system, the communication between the stakeholders and the requirements engineer decreases.

An example of a series of events can be demonstrated by the chain requirements engineer domain knowledge, requirements engineer understanding of the need, requirements engineer misinformation, and requirements volatility as is shown in Figure 34. One can construct the causal dependency in the following way. As the requirements engineer domain knowledge increases, it will cause the requirements engineer understanding of the need to increase. As the requirements engineer understanding of the need increases, it will cause the requirements engineer misinformation to decrease. As requirements engineer misinformation to decrease, it will cause the requirements volatility to decrease.

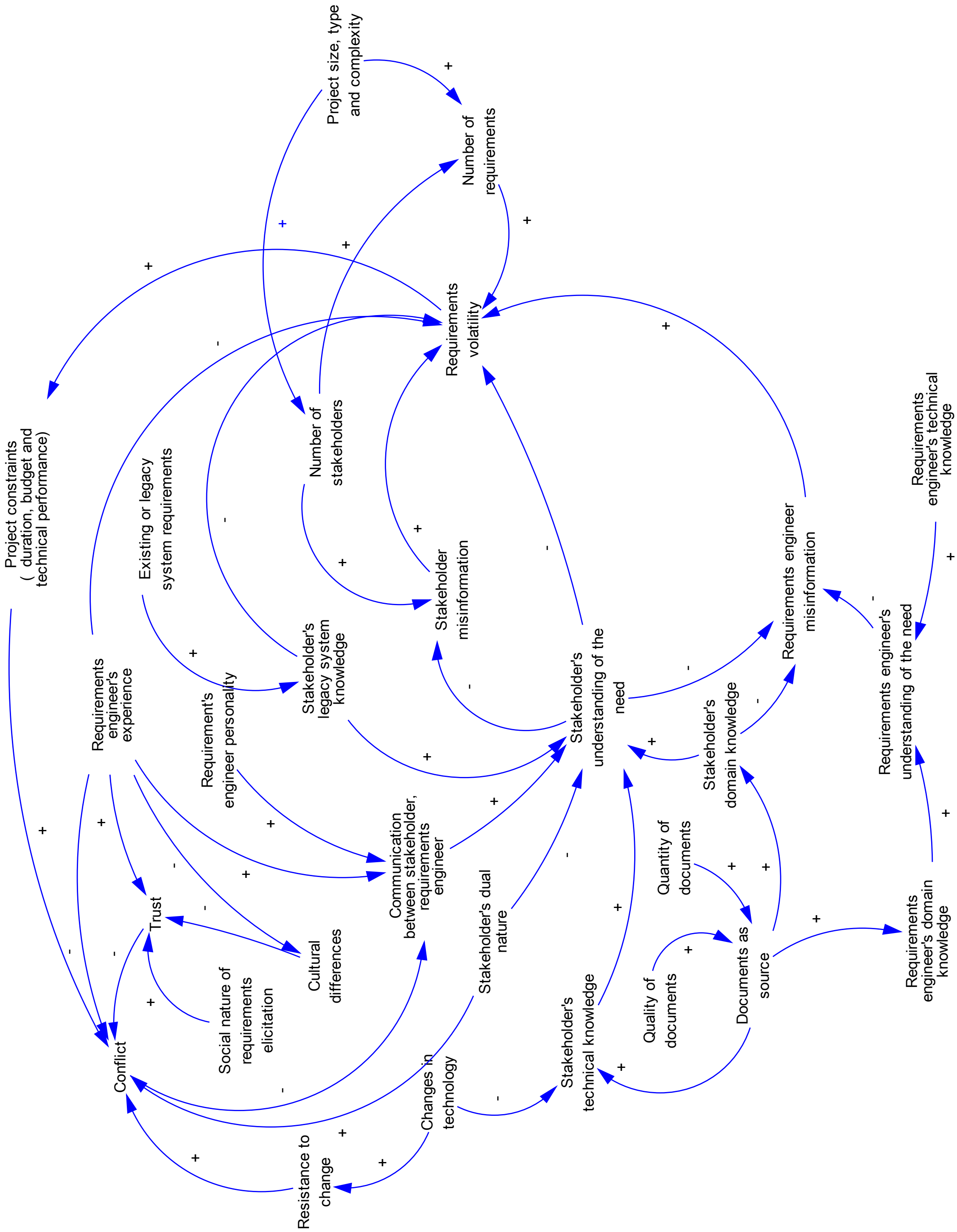


Figure 34: Causal loop diagram showing the interaction between the various elements (own contribution)

One of the metrics that is used in this thesis to evaluate the stability of the set of requirements in a project is to look at the requirements volatility within the project. By using the causal loop diagram as a basis, the following mechanisms and causes for different parameters that contribute to requirements volatility can be identified. The first-level parameters are shown in Figure 35.

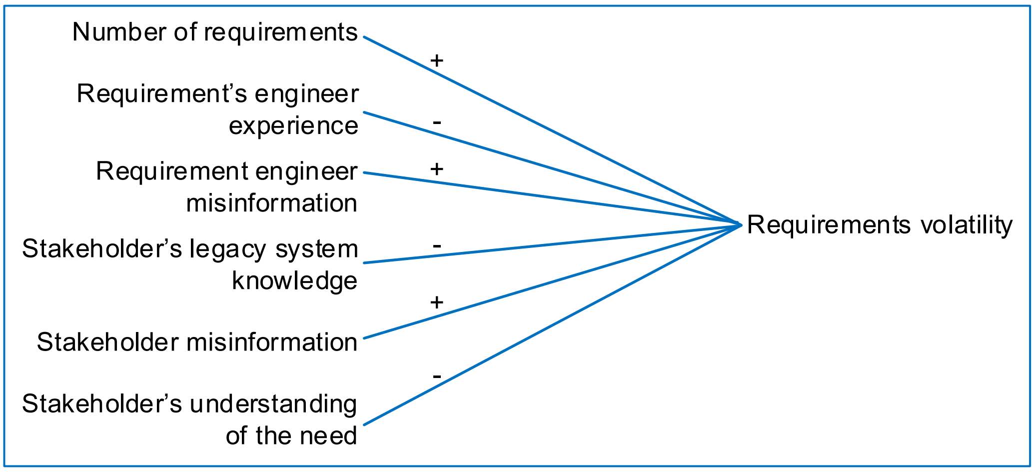


Figure 35: Factors that give rise to requirements volatility (own contribution)

The following mechanisms can be identified and explained from

Figure 35.

1. An increase in the project size, type and complexity will increase the volatility of the requirements set.
2. The greater the experience a requirement engineer has, the less the volatility of the requirements set will be.
3. The more the misinformation introduced by the requirements engineer or the stakeholder, the more the volatility of the requirements set will be.
4. An increase in legacy system knowledge or the understanding of the need by the stakeholder will decrease the volatility of the requirements set.

By examining the causal loop diagram shown in Figure 34, it quickly becomes clear that there are not only single mechanisms at play in the problem situation but also multiple and multi-layer mechanisms. By peeling the proverbial onion, one layer more one can identify the secondary mechanisms in play. These secondary factors are shown in Figure 36. By examining the secondary mechanisms, it becomes clear that there are specific parameters that influence multiple other parameters. These parameters are indicated in brackets as, for example, (stakeholder's understanding of the need). This type of analysis allows one to start identifying essential parameters that may require additional investigation.

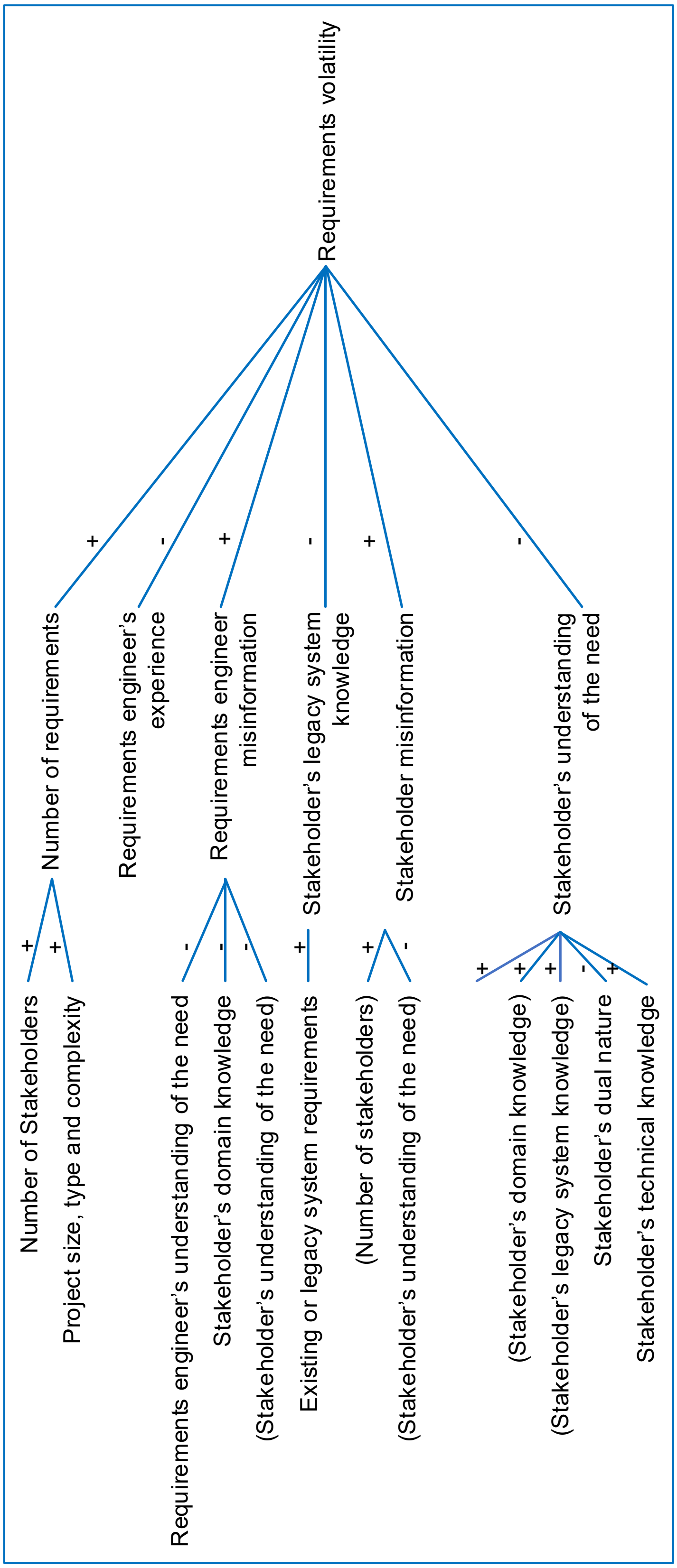


Figure 36: Secondary mechanisms causing requirements volatility (own contribution)

Additional investigation can be done by tracing the mechanism in the opposite direction. The result is shown in Figure 37.

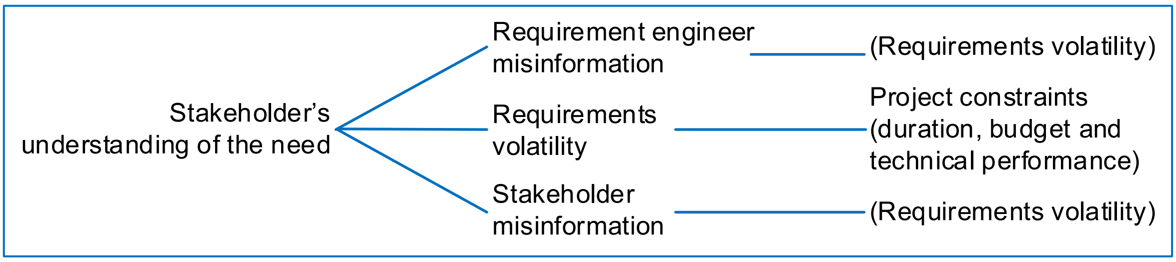


Figure 37: Effect of the stakeholder's understanding of the need on the secondary parameters (own contribution)

The causal effect of the other factors on requirements volatility can be examined similarly, leading to an increased understanding of the overall requirements engineering process.

3.5.3 Evaluation of the effect of the observed phenomena

The effect of these mechanisms on the stability of the requirements set is best examined by investigating the feedback loops that are present and determining whether these loops are balancing loops or reinforcing loops. The different feedback loops are shown in Figure 38 to Figure 40.

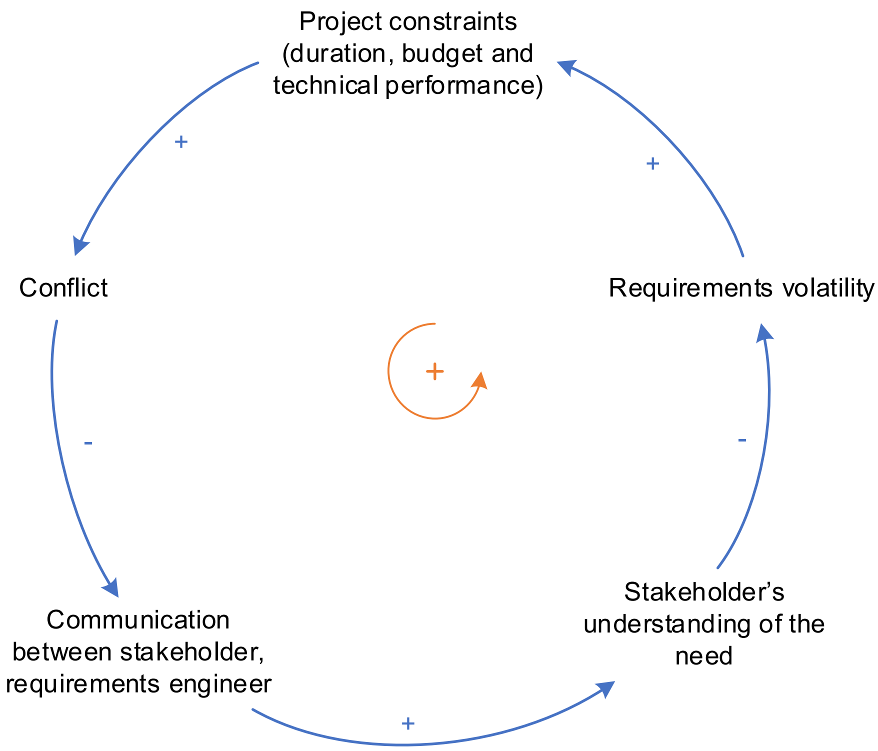


Figure 38: Feedback Loop 1 (own contribution)

The feedback loop shown in Figure 38 can be interpreted in the following way. As the stakeholders' understanding of the need decreases, this increases requirements volatility. As requirements volatility increases this in turn again increases the effect of the project constraints. As the effect of project constraints increase, they, in turn, increase the conflict present between the various stakeholders, which decreases the communication between the stakeholders and the requirements engineer. As this communication decreases it further decreasing the stakeholders' understanding of the need. This loop is a reinforcing loop, which tends to amplify a problem situation that is present.

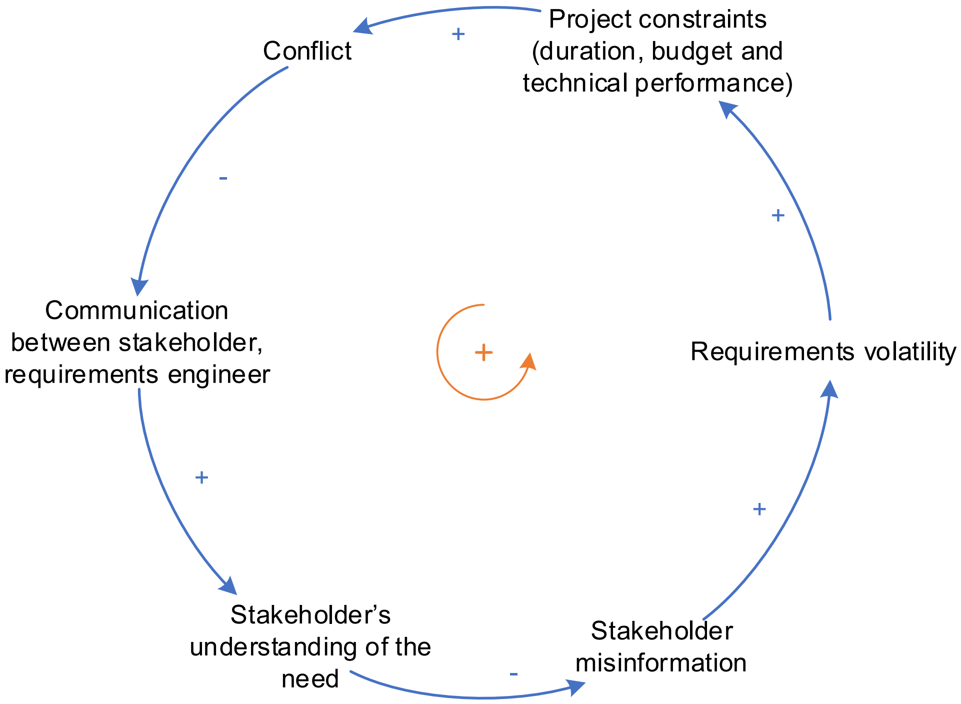


Figure 39: Feedback Loop 2 (own contribution)

The second feedback loop that is shown in Figure 39 can be interpreted in the same way as the first feedback loop. As for the first loop, the second loop is also a reinforcing loop. The third feedback loop that is shown in Figure 40 can again be interpreted as for the first two feedback loops that were shown. As for the previous feedback loops, this is again a reinforcing loop.

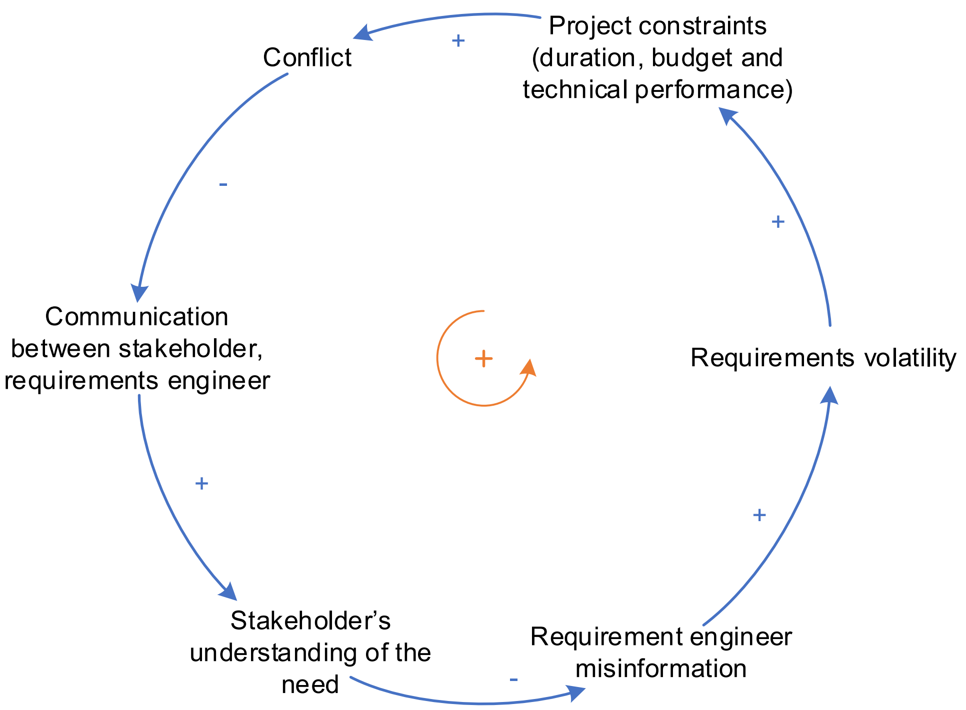


Figure 40: Feedback Loop 3 (own contribution)

Two observations can be made by examining the feedback loops shown in Figure 38 to Figure 40 and the causal loop diagram shown in Figure 34. The first observation relates to the relative importance of the stakeholders' understanding of the need by observing the number of factors it is affecting. These factors are shown in Figure 41.

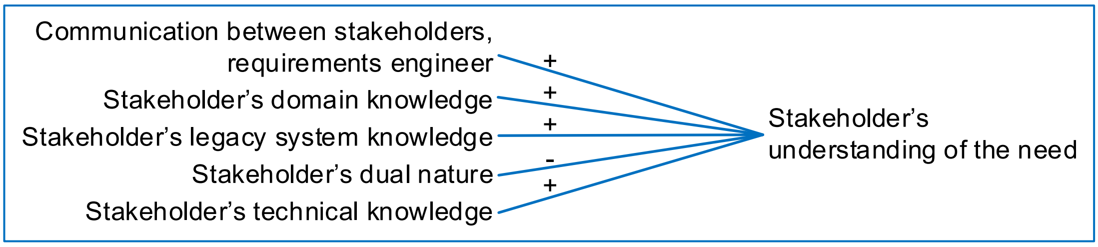


Figure 41: Factors impacting on the stakeholder's understanding of the need (own contribution)

The second observation is that the nature of the feedback loops confirms one of the earlier comments made in paragraph 1.3.2. When working in a complex socio-technical system, a complete solution to a problem situation may prove to be challenging, and that the best one can hope for is an improvement in the situation. The feedback loops are all reinforcing loops, which implies that there is no single parameter that will tend to stabilise it automatically. The requirements engineer must actively work to reduce the effect of aspects such as conflict and increase communication among the stakeholders to reduce the requirements volatility.

3.6 Expanded conceptual framework

Paragraph 2.3.3 discussed an initial conceptual framework and identified the function of a conceptual framework to define what is studied, why is it studied (relevance), and how it will be studied (methodology and rigour). This paragraph will expand on the principles established in Chapter 2.

Jabareen (2009) defined a concept as consisting of components. The characteristics of the concept is further defined by the characteristics of the individual components. A conceptual framework can be defined further as consisting of a network of interconnected concepts that provides a complete understanding of the observed phenomena.

The following components of the conceptual framework for this thesis can be identified in this research:

1. The requirements engineering process artefact that consists of methods such as the requirements engineering process and requirements elicitation techniques.
2. The project problem context that consists of the internal and external stakeholders and role players.
3. An organisational hierarchy may exist between the internal stakeholders and role players, which may affect their cooperation within a team environment.
4. An informal hierarchy may exist between external stakeholders and role players that may also, firstly, affect their ability to work together and present a common need and, secondly, affect their ability to cooperate with the internal stakeholders and role players.
5. Each of these stakeholders may have their own priorities, concerns and responsibilities. They also have their own perspective and perception of the need that must be addressed (Ahmad, 2008).

These components interconnect to function as a complex socio-technical system with the aim of achieving a common goal or perform a common function (Rutherford, 2018). The interconnections of the various components are not static and can react to their environment, respond to changes and find ways of surviving (Rutherford, 2018).

The interconnections can manifest themselves not just as a direct or physical connection but can also be defined by the flow of information. These interconnections can form feedback loops, which can affect the dynamic behaviour of the system. These interconnections and feedback loops have already been identified and demonstrated in paragraph 3.5.

3.7 Research sample population

The research sample population will be selected from the overall population of industries that run development projects and that use some form of a requirements engineering process. The sample population should not only include examples of where the requirements engineering process proved to be problematic, but also where the process was deemed to be successful. The relationship between the researcher, the object-of-study sample and the population from which the sample will be selected as shown in Figure 15. The relative position of the population from which the sample will be selected in relation to the researcher is shown in Figure 42.

The function of the research conducted within these industries (e.g. the banking sector or services sector) is to use the data generated to validate the research tool. It is not intended that any of the suggestions for improvement be returned to the industries for implementation as this falls outside the scope of the research conducted for this thesis.

3.8 Research process operation concept

The research will be carried out using a combination of fieldwork and laboratory work and will include some or all the following steps. The steps may also require some iteration depending on the case being investigated.

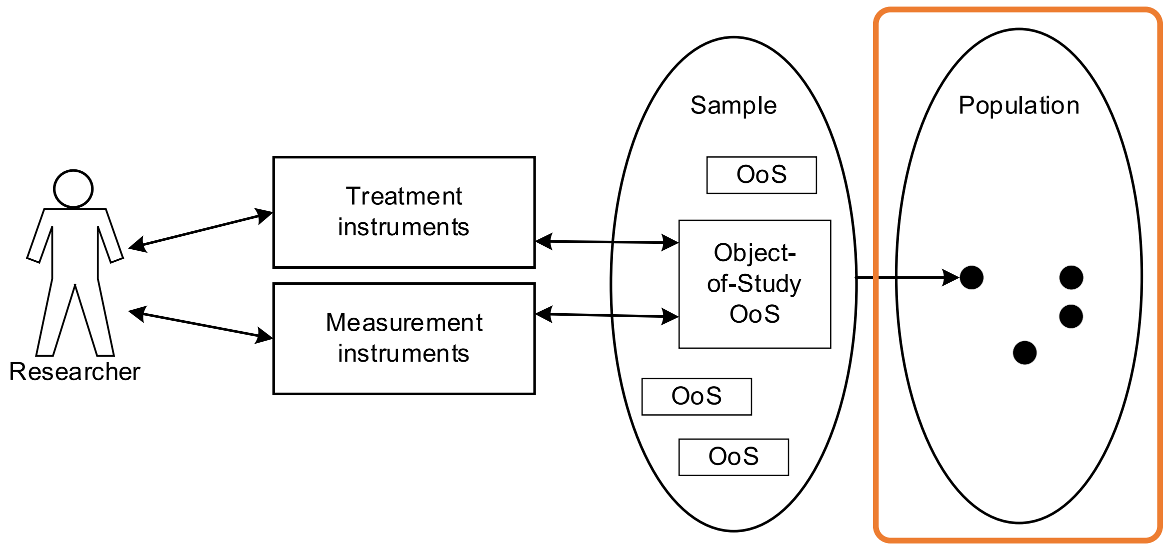


Figure 42: Relative position of the population from which the sample will be selected in relation to the researcher [redrawn and adapted from Wieringa (2014)]

The research process will be conducted as several single-case experiments. The research tool that will be used during these experiments will need to include both a measurement capability and an evaluation capability. The central premise for the research will be to investigate the parameters identified previously that define the object-of-study. These parameters will be discussed and reviewed with the identified research stakeholders or SMEs in an action research setting. Once the applicability of the different factors has been confirmed or expanded, the research stakeholders or SMEs will be requested to provide their opinion of the behaviour of these parameters during the definition and design phase of the project using the measurement capability of the research tool. After the behaviour has been measured, it will be evaluated using the evaluation capability of the research tool.

The research process defined by the researcher for this thesis will include the following steps:

1. Identifying possible research subject candidates in the form of stakeholders within the object-of-study or SMEs who can be interviewed.
2. Explaining the research process to the research subjects. If they consent to the process, they will be requested to sign the ethical clearance form. If required, any relevant management approval as part of the ethical clearance process will also be obtained.
3. Obtaining a brief background of the research subject to establish their experience and the role that they play within the organisation.
4. Establishing the phenomena that are observed within the organisation and comparing these with the identified factors or parameters that form part of the object-of-study and are included in the research questionnaire. These parameters will also be reviewed with the research subject to identify any unnecessary or missing parameters. The list of parameters will also be discussed with the research subject to ensure that there is clarity regarding the meaning of the various parameters.
5. Discussing the causal loop diagram with the research subject to ensure that it represents the situation in the specific organisation. This is important as the causal loop diagram will be used as the basis for the single-case mechanism experiments and inferences that will be conducted later as part of the laboratory experiment.
6. Updating the list of parameters that define the object-of-study based on the outcome of the discussion with the research subject and updating the parameters and their relationship as they are included in the causal loop diagram.
7. Understanding the interpretation and behaviour of the various parameters as they are represented by the research subject. The behaviour is provided by the research subject in terms of their behaviour versus time, which is therefore essential for the researcher to understand.

8. Evaluating the results using the simulation tool once the interview has been completed and identifying potential areas where a cascade of parameters may occur that can either increase or lessen the effect of the various parameters on the volatility of the requirements used within the project.

3.9 Identified stakeholder requirements

The stakeholder requirements can now be derived from the information contained in the previous sections. These requirements are summarised in Table 15.

Table 15: Stakeholder requirements

No.	Stakeholder requirement
SH1	The requirements engineering research tool shall provide a mechanism that can be used to investigate behaviour in the requirements engineering domain.
SH2	The requirements engineering research tool shall provide a mechanism to evaluate potential solutions under simulated conditions without requiring implementation of these solutions in the real world.
SH3	The requirements engineering research tool shall support an ethical research process.

3.10 Chapter summary and conclusion

The research problem to be addressed in this thesis was defined in terms of the relevant research stakeholders, their goals and needs, and the observed phenomena that occur in the project problem context. These observed phenomena were identified from the object-of-study defined in Chapter 2.

The mechanisms present in the project problem context were identified using a causal loop diagram that illustrated the interaction between the different elements. Three feedback loops were identified from the causal loop diagram. It was further determined that all three feedback loops had reinforcing behaviour, which can lead to uncontrolled growth if not managed correctly. This observation also supports the notation that within a complex socio-technical system, such as the one being investigated, a total and permanent solution to the problem will not be achievable.

An expanded conceptual framework was discussed that identified and considered different components of the conceptual framework. This conceptual framework was used to identify the knowledge and research questions. These knowledge and research questions were used to state three stakeholder requirements for the research tool. These requirements will be analysed further and transformed into system requirements in Chapter 4. These system requirements will be used to formulate the specification for the research tool.

CHAPTER 4. REQUIREMENTS ENGINEERING RESEARCH TOOL SPECIFICATION PHASE

"Anyone can write a specification, but if nobody implements it, what is it but a particularly dry form of science fiction." – Ian Hickson

4.1 Introduction

The stakeholder requirements for the research tool were identified in Chapter 3. The primary objective of the research tool specification phase presented in this chapter (and previously discussed in paragraph 1.4.3) is to transform the stakeholder problem-orientated view to a solution-orientated view from which the requirements and specification will be identified for the research tool design, implementation and verification phase that is presented in Chapter 5.

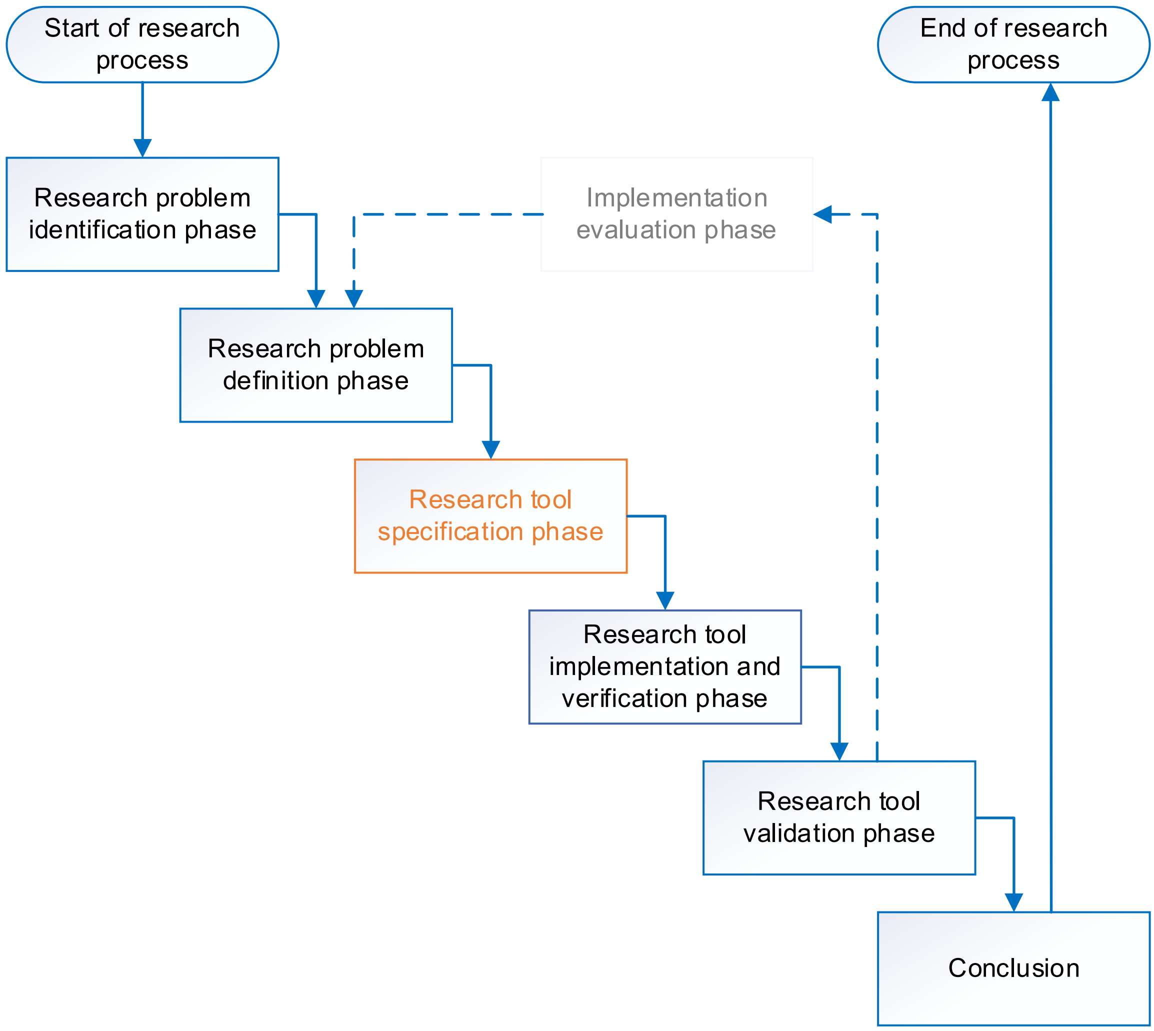


Figure 43: Relative location of the requirements engineering research tool specification phase in the overall research process

The relative position of the research tool specification phase in the overall research process is shown in Figure 43 above.

The main functions that form part of this phase are: (a) identification and definition of the research tool requirements; (b) analysis of the research tool requirements; (c) definition of the research tool architecture; and (d) evaluation of the research tool to the contribution to the stakeholder goals. This process is shown in Figure 44.

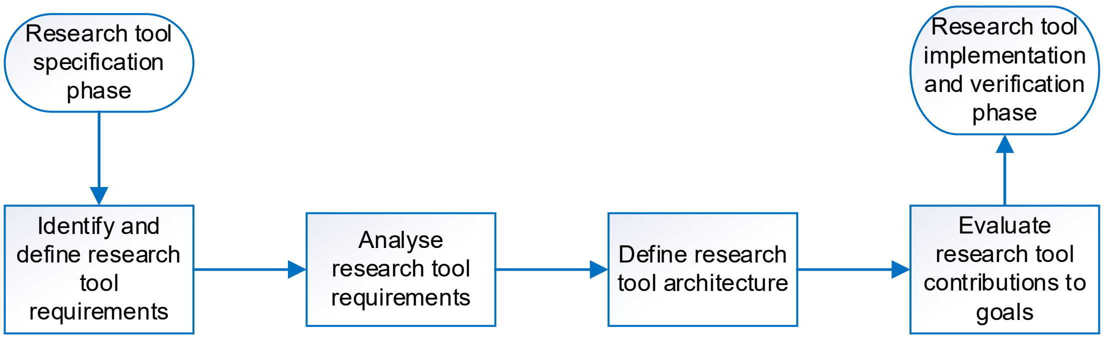


Figure 44: Requirements engineering research tool specification phase FFBD

The research tool specification phase can be equated to the system requirements definition process and architecture definition process as defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

4.2 Identification and definition of the research tool requirements

The requirements engineering process that converts the undiscovered stakeholder needs into the unstructured stakeholder requirements is generally referred to as the requirements elicitation or discovery process (Carlshamre, 2001; Sommerville and Sawyer, 1997). This elicitation or discovery process forms the first part of the requirements engineering process as shown in Figure 26 and discussed in paragraph 2.4.3.1.2.

The process of transforming the unstructured stakeholder needs into the research tool requirements is done during the requirements analysis process (Ryan, 2013). This process is shown in Figure 45.

The first step in transforming the stakeholder needs into the research tool requirements is for the requirements engineer is to understand the bigger picture of the overall problem situation that must be solved. This is done by creating a mental model of the overall problem situation. This mental model will be used as a guide or structure within which the elicited requirements can be mapped. The inputs for this mental model are the stakeholder requirements identified in Table 15 and the resulting research process operational concept defined in paragraph 3.8.

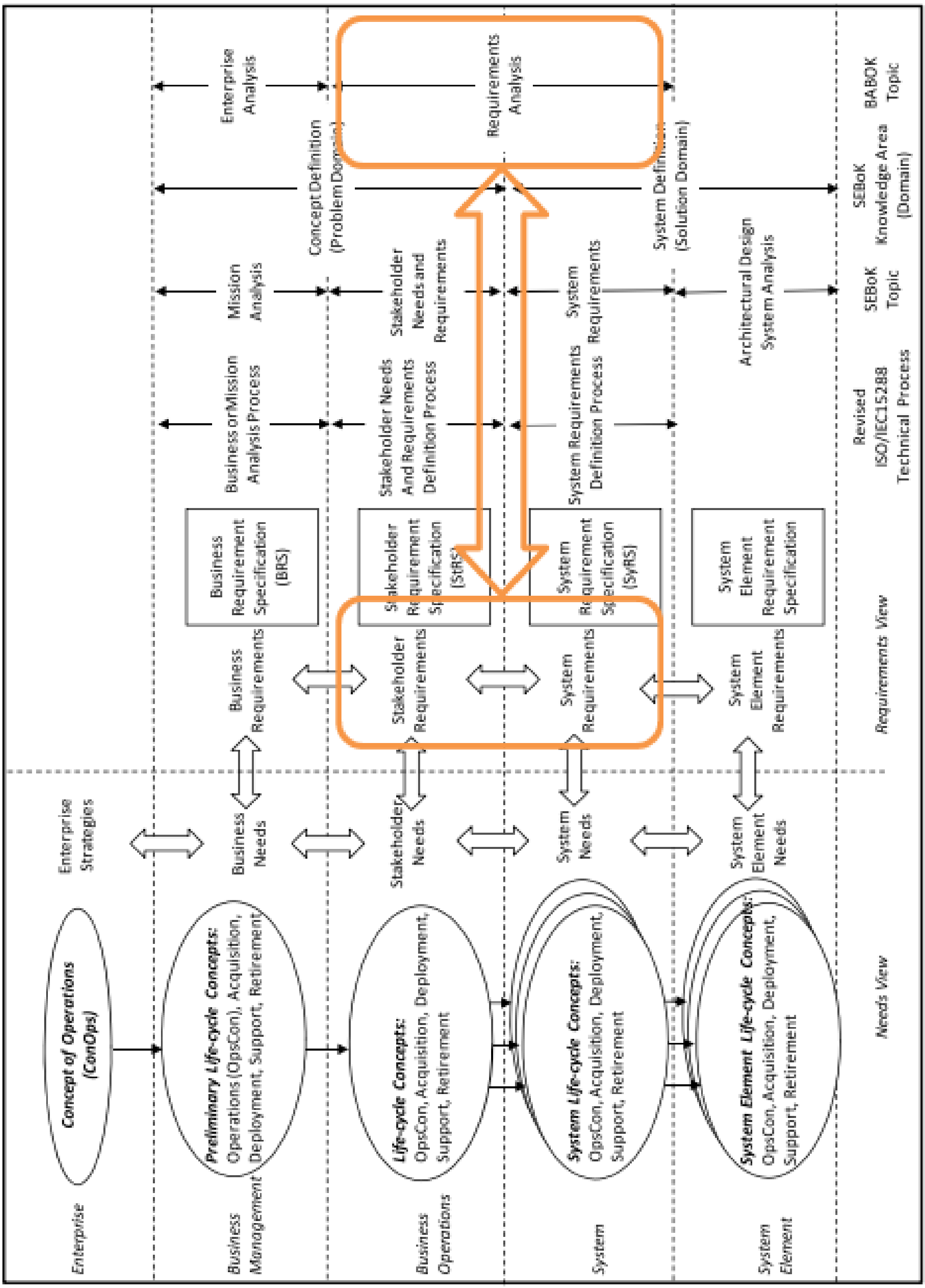


Figure 45: Process of transforming the stakeholder requirements into the research tool requirements [(adapted from Ryan, 2013)]

4.2.1 Research tool mental model

The function of the research tool is to investigate and learn about a specific problem situation with the aim of improving the situation within the requirements engineering domain. One possible mental model that can be used is based on soft systems methodology as was defined by Checkland (1999).

Soft systems methodology is a systems thinking methodology that can be used in organisational process modelling. Soft systems methodology focuses on the analysis of complex situations by taking different views of the definition of the problem situation (Checkland, 1999).

The critical elements of the soft systems methodology are (a) the description of the real-world situation of concern, and (b) a comparison of a model of the systems with the perceived real-world situation. The result of this methodology is a better understanding of the situation, and that an agreed-upon action is needed to improve the situation. This inquiring/learning process is shown in Figure 46.

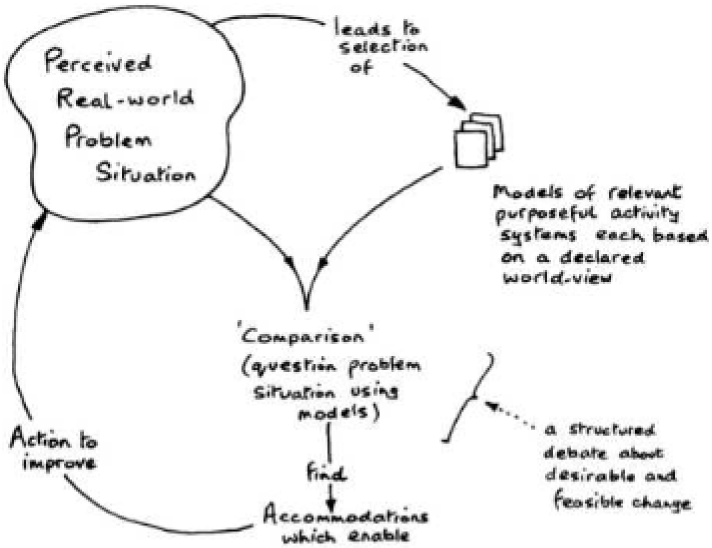


Figure 46: Inquiring/learning cycle of soft systems methodology (Checkland, 1999)

The basic soft systems methodology principles are the following (Checkland, 1999):

1. The real-world situation is considered to consist of a complexity of relationships.
2. These relationships can be explored via models of purposeful activity that are based on specific world views.
3. The inquiry into the real-world situation of concern is made using the models as the source of the questions.

4. An “action to improve” will be based on finding a version of the situation within which conflicting interest can find common ground.
5. The inquiry process can prove to be never-ending (i.e. it can be classified as an improvement problem as was discussed in Chapter 3).

4.2.2 Requirements engineering research tool top-level requirements

By using the mental model (discussed paragraph 4.2.1) as a framework, the following research tool top-level requirements can be defined. The real-world situation of concern represents the current situation regarding the requirements engineering environment in the organisation that is being researched as part of this thesis.

4.2.2.1 The real-world situation of concern

R1. The research tool shall provide a method to identify the parameters that can be used to describe the requirements engineering environment (real world of concern) being researched.

R2. The research tool shall provide a method to document the relationship between the different parameters that can be used to describe the requirements engineering environment (real world of concern) being researched.

R3. The research tool shall provide a method for the researcher to measure the current state or current behaviour of the parameters that can be used to describe the requirements engineering environment (real world of concern) being researched.

4.2.2.2 Comparison of models with the perceived real-world situation

R4. The research tool shall provide a method to model an ideal-world scenario.

R5. The research tool shall provide a method to process the current state of the parameters of the requirements engineering environment (real world of concern) that is being researched into a model of the real-world situation.

R6. The research tool shall provide a method to compare different ideal models with the “real-world situation of concern”.

R7. The research tool shall support an ethical research process.

4.2.2.3 Action needed to improve the situation

R8. The research tool shall provide a method to evaluate different models of improvement in order to select the best alternative prior to implementation.

4.3 Analysis of the requirements engineering research tool requirements

The requirements analysis process was discussed in paragraph 2.4.3.1.2. The purpose of the analysis process is to determine if there are any conflicting, omitted or inconsistent requirements. The boundaries of the system will also be defined (Sommerville *et al.*, 2003).

4.3.1 Conflicting, omitted or inconsistent requirements

The process of determining conflicting, omitted or inconsistent requirements is done by tracing the allocation of the stakeholder requirements down to the research tool requirements. Based on these parameters, the requirements as stated in paragraph 4.2 are concise, clear and unambiguous. The resulting tracing of the stakeholder or user requirements to the research tool is shown in Figure 47.

4.3.2 Research tool boundaries

The following boundaries and implications for the design of the research tool are applicable:

1. The research tool will be used in a complex socio-technical environment. It will not be released to the general public but will be used by researchers or other requirements engineering practitioners. This implies that the user of the tool will be a highly trained and capable person. Thus, the "user friendliness" and level of documentation of the research tool can be lower.
2. The purpose of the tool is not necessarily to arrive at a complete solution to the problem being investigated, but rather to enhance the researcher or the requirements engineer's understanding of the problem being investigated.

4.4 Requirements engineering research tool architecture

Several decisions must be made to identify the correct architectural solution to the design problem. These decisions include identifying the following approaches that will be used to analyse the data collected: (a) the experimental research method, including the inference design; (b) the data collection method; and (c) the design of the data analysis method.

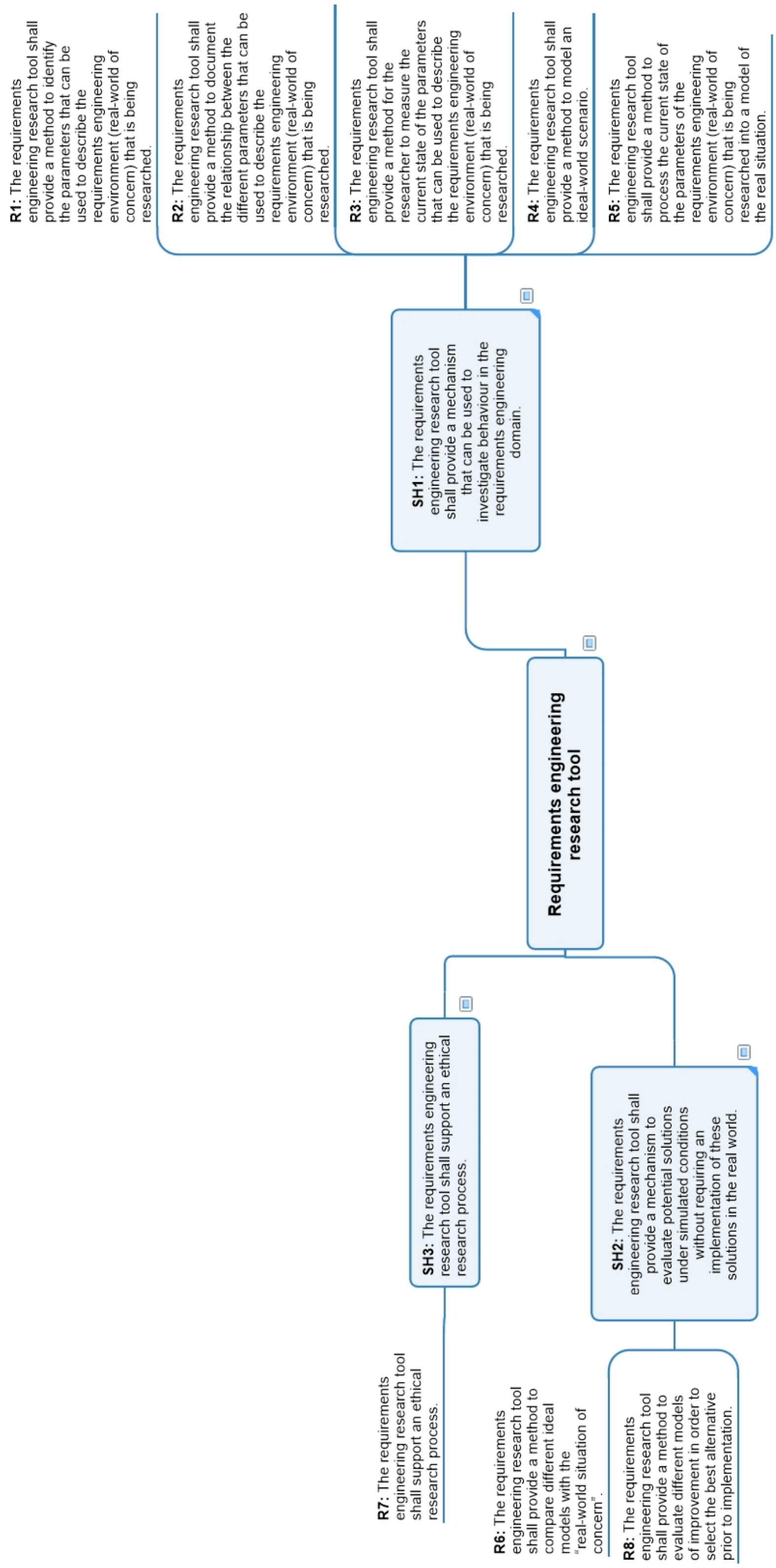


Figure 47: Requirements engineering research tool requirements tracing

4.4.1 Experimental research method

The design science research methodology that forms the basis of the research presented in this thesis focuses on investigating an artefact within its problem context. It has been established in Chapter 1 that the requirements engineering problem context is a complex socio-technical system. Here, the term 'socio' relates to involving society in the problem context. For this reason, the selection of the experimental research method should consider this factor (Panchal and Szajnfarber, 2017; Szajnfarber and Gralla, 2017).

Since the object-of-study (consisting of the artefact interacting with the problem context) can vary extensively over the different samples being studied, the experimental design should support the investigation of each of the sample objects-of-study individually. Franklin, Allison and Gorman (1996) identified that the type of experiment that supports this type of research is a single-case study. When one includes the need to explain the mechanics that may influence the socio element present in the object-of-study that gives rise to the observed phenomena, this leads to the selection of a single-case mechanism experiment (Elster, 2015; Franklin *et al.*, 1996; Wieringa, 2014). This type of experiment supports an inductive reasoning process that draws on inferences from both the experiences of the researcher and on the empirical data collected (Franklin *et al.*, 1996).

A single-case mechanism experiment is based on a test of a mechanism within a single object. The researcher manipulates the independent variable of the case to explain the responses in terms of mechanisms internal to the case (Franklin *et al.*, 1996; Wieringa, 2014). The results of a single-case mechanism experiment are usually described in terms of the architecture and components of the object-of-study. This can be used to explain the observed phenomena in terms of the mechanisms that exist in the object-of-study (Wieringa, 2014). The cases studied using a single-case mechanism experiment can include both social systems and technical systems, or representative models of these systems. Single-case mechanism experiments are also ideally suited to problem investigation scenarios as they can provide insight into the observed behaviour in objects-of-study in the real world (Wieringa, 2014).

4.4.2 Single-case mechanism experiment sampling

The samples for the single-case mechanism experiment will be selected from the larger population of organisations (refer to paragraph 3.7 describing the research sample population) that execute complex projects using a system engineering process and that are experiencing one or more of the phenomena previously identified in paragraph 3.5. The object-of-study sample will be represented by the research subject in the form of one or more of the direct stakeholders in the project or an SME who has a thorough knowledge of the organisation and the problems being experienced.

4.4.3 Data collection

The single-case mechanism experiment will use data that has been collected from the various stakeholders or SMEs who will be identified when the sample population has been selected. The data required for the measurement activity will be collected using a questionnaire, which will include a qualitative recording method of the response of the research subject. This questionnaire will be completed by the stakeholder or SME during a semi-structured interview in an action research[1] setting. The semi-structured interview and action research setting are required as the researcher is an active participant in the interview. The researcher may find it necessary to discuss the results or share some of their own experiences in order to obtain and interpret the data correctly.

4.4.4 Data preparation and interpretation

The data obtained as part of the data collection process will be in a qualitative format. This data will need to be converted into a quantitative format that can be used in a numerical simulation model. Figure 48 shows the process conceptually.

Figure 48: Data preparation and interpretation process [adapted from Wieringa (2014)]

The response data provided in a graphical form will be converted to a numerical format using a digitisation process.

4.4.4.1 Descriptive inference process

The digitised data will be prepared via a descriptive inference process to ensure that the results fall within the bounds of the experiment. The validity of the data set will be confirmed after the digitisation and data preparation process to ensure that it is still representative of the data collected during the interviews and that the various phenomena are still visible. If any data has to be removed from the data set, it must

[1] Action research is a research approach that situates the research in a local context and focuses on a local issue. The research is conducted by the researcher and for the purpose of the researcher. It leads to some form of action or a change implemented by the researcher in the research context (Checkland and Holwell, 2007; Ary, Jacobs, Sorenson and Razavieh, 2010; Jrad, Ahmed and Sundaram, 2014).

be justifiable within the bounds of the single-case mechanism experiment (Wieringa, 2014).

This descriptive inference process shall include the following activities:

1. Symbolic data shall be interpreted
2. Data shall be summarised
3. Interview data shall be transcribed
4. Invalid data shall be removed or ignored

The descriptive inference process summarises the data into descriptions of the observed phenomena. This process requires that the data must be prepared in a way that it can be processed. Data preparation can be described as transforming the data from its original form into a form that makes it possible to be processed.

Data preparation includes activities such as transcribing interviews, transforming measurement scales to facilitate quantitative analysis, removing outliers or data that is out of bounds, removing records with missing data, and cleaning up primary data. Data preparation can be valid if there is not a change between the observed phenomena and the version represented by the data.

4.4.4.2 Abductive inference process

Abductive inference can be described as an inference to the best explanation. The purpose of abduction is to infer something in relation to unobserved causes or explanatory reasons for the observed events (Schurz, 2008). Abductive inference can be described as ampliative[2] and uncertain. Abduction is typically used in case-based research to explain case observations (Wieringa, 2014). Three types of explanations can be found, namely, causal, architectural, and rational explanations.

It is important to note that all forms of abductive inference can be fallible. This can be explained by an example in which an earlier event made a difference to a variable. In this case, a comparison is made between what would have occurred had the earlier event not taken place and what would have occurred if the earlier event did take place. When individual cases are considered, this may prove to be an unverifiable statement (Wieringa, 2014).

The following questions shall be considered during the design of the abductive inference process:

[2] Ampliative (from Latin *ampliare*, "to enlarge"), a term used mainly in logic, meaning extending or "adding to that which is already known".

1. What possible causal or mechanistic explanations can be put forward to explain the observed phenomena?
2. What data is required to formulate those explanations?
3. Could the selection or sampling mechanism influence the explanations?
4. Are there factors that will cause the research tool to influence the outcome of the experiments? [adapted and expanded from Wieringa (2014)]

A phenomenon (explanandum) is explained by citing the earlier phenomenon (explanans) that caused it (Elster, 2015). Two different types of explanations are considered in this thesis, namely, causal and mechanistic explanations.

4.4.4.2.1 Causal explanations

A causal explanation is used to describe the change in a variable *Y* as a result of an earlier change that occurred in a variable *X*. This can be defined as a difference-making view of causality that explains a change in *Y*. Causal inference is a reasoning process used to provide a causal explanation of observed phenomena (Wieringa, 2014). A causal explanation moves in a backwards direction. The cause of an event or phenomena is the event that initiated the event (Williamson, 2011).

4.4.4.2.2 Architectural and mechanistic explanations

An architectural explanation defines that a phenomenon occurred in the object-of-study due to components of the object-of-study interacting to produce the observed phenomenon (Wieringa, 2014). The interactions can collectively be described as the mechanisms that produced the phenomena (Wieringa, 2014). Mechanisms can be classified as being either deterministic (always produce the phenomena) or nondeterministic (do not always produce the phenomena). A mechanism may further include a feedback element as part of the architectural explanation (Elster, 2015).

A mechanistic explanation moves in a downwards direction. A mechanism can be described as the constitution of reality that produced a phenomena (Williamson, 2011). Architectural explanations are only possible when the object-of-study has an architecture with components that can produce these interactions (Wieringa, 2014). Architectural explanations are usually the result of case-based experiments. In a single-case mechanism experiment, the research experiments using a single case to try and explain the observed behaviour in term of the object-of-study architecture (Wieringa, 2014).

4.4.5 Evaluation and interpretation of results

Williams (2001) identified that measurement within the requirements engineering environment allows for requirements engineers to (a) understand the problem situation; (b) establish a baseline, and (c) make assessments and predictions. The term 'measurement' here is not only considered to be the process of assigning numbers to phenomena according to some form of rule, but also the evaluation of the results.

In deciding how the measurement process will be applied, it is important to once again consider that the requirements engineering domain is a complex socio-technical problem (as was established in Chapter 1). Flood & Jackson (1991) proposed that when approaching such a complex socio-technical problem from a systems thinking point of view, different system metaphors should be used as viewpoints that can be used to examine the problem from different angles; therefore, presenting a richer and more complete picture.

The metaphors are: (a) the machine metaphor or closed system view; (b) the organic metaphor or open system view; (c) the neuro-cybernetic metaphor or viable system view; (d) the cultural metaphor; and (e) the political metaphor (Flood *et al.*, 1991). Based on the metaphors or viewpoints identified, they developed a grouping that identified specific systems thinking approaches that can be applied to different problem contexts. This mapping is shown in Table 16. The vertical axis of the table represents the complexity of the system under consideration. The horizontal axis of the table refers to the relationship between the different participants present in the socio-technical system.

Table 16: Problem context groupings (Flood *et al.*, 1991)

	Unitary	Pluralist	Coercive
Simple	• OR[3] • SA[4] • SE[5] • SD[6]	• SSD[7] • SAST[8]	Critical system heuristics

[3] OR: Operational research
[4] SA: System analysis
[5] SE: Systems engineering
[6] SD: Systems dynamics
[7] SSD: Social system design
[8] SAST: Strategic assumption surfacing and testing

	Unitary	Pluralist	Coercive
Complex	• VSD[9] • GST[10] • Socio-tech[11] • Contingency Theory	• Interactive planning • SSM[12]	No suggestion

A simple system stands in contrast to a complex system in terms of the number of elements; the magnitude of the interaction between the elements as well as the level of organisation of the interaction; the stability of the system over time; and whether the system is open or closed (Flood *et al.*, 1991).

The differences between a unitary, pluralist or coercive relationship among the different participants can be distinguished in terms of the degree of common interest that they share; the compatibility of their values and beliefs; the degree to which they agree on the ends and means; the level of participation in decision-making; and the degree to which they act in accordance with the agreed objectives (Flood *et al.*, 1991).

When designing an evaluation tool that can be used in a single-case mechanism experiment, one is dealing with a simple, unitary system. The selected evaluation tool must also be capable of supporting the establishment of a baseline and assessment and prediction, as was identified by Williams (2001). The most suitable design alternative that can be used is a system dynamic simulation model. System dynamics modelling provides a mechanism for modelling complex systems that defies common intuitive solutions. Furthermore, the system dynamics model provides a mechanism for assisting the users of the simulation in the way that they perceive the situation or problem, which enhances learning in a complex world. A system dynamics model should communicate the new mental model to the audience or modify previous mental models that may exist (Forrester, 1986; Sterman, 2000).

System dynamics is a method for describing, modelling, simulating and analysing dynamic complex feedback systems in terms of the processes, information, organisational boundaries and strategies such as found in business and other social systems [Wolstenholme E (1990) quoted by Pruyt (2013)]. Feedback refers to the situation of *X* affecting *Y* and, in turn, *Y* affecting *X* – perhaps through a chain of causes and effects. One cannot study the link between *X* and *Y* and, independently, the link between *Y* and *X*, and predict how the system will behave. Only the study

[9] VSD: Viable system diagnosis
[10] GST: General system theory
[11] Socio-Tech: Socio-technical systems thinking
[12] SSM: Soft systems methodology

of the whole system as a feedback system will lead to correct results [System Dynamics Society quoted by Khan and McLucas (2008)].

A system dynamics model can only add value to a process if the purpose and objective of the model have been clearly identified upfront. This will be done by creating a dynamic hypothesis that specifies the design of the system dynamics model. The output of the system dynamics model should organise, clarify and unify the knowledge surrounding the problem (Forrester, 1986).

4.5 Requirements engineering research tool design specifications

Based on the research tool requirements presented in 4.2.2 and the preceding discussion, the main design specification of the research tool can be defined (shown in Table 17).

Table 17: Requirements engineering research tool main design specifications

Specification	Description
Type of experiment to be supported	Single-case mechanism experiment.
Data collection method	Questionnaires completed in a semi-structured interview within an action research setting.
Data evaluation and interpretation method	Descriptive inference, abductive inference, causal and mechanistic explanations. Evaluation with respect to requirements volatility using a systems dynamic simulation model.

4.5.1 Research scenario

The artefact can consist of different types of items, which include among others, constructs, models, methods, instantiations algorithms, methods, notations, techniques, and even conceptual frameworks (Hevner *et al.*, 2004; Wieringa, 2014). During the design activity, the artefact's desired functionality and architecture will be defined. Different alternatives for the research tool should be considered and evaluated before selecting an architecture or design that meets the requirements.

The design of the specific implementation of the research tool should be consistent with the architectural entities as defined in models and views of the system architecture (INCOSE, 2015). The research tool must support the specific research and inference design including the operational research concept that was defined previously in paragraph 3.8. In doing so, the object-of-study, sampling of the research population, and the specific measurements that will be performed must be defined. To design the inference process and method of the data collected, it must be decided whether descriptive inference, statistical inference, abductive inference or analogical inference will be used.

The objective of this research is to design a research tool that can be used by both researchers and requirements engineering practitioners to do problem investigation, improvement investigation and improvement validation within the problem context they are working.

The research tool will not only consist of a simulation model but also the questionnaire used to collect the data. The questionnaire will be used to take a snapshot of the current situation. The results will be put through a data preparation exercise to normalise the data. Thereafter, the data will be used to calculate a requirements volatility score using the simulation model.

Improvement investigation – Selective changes will be made on a simulation basis to determine the overall effect thereof on the requirements volatility score.

Improvement validation – The suggested improvements will be rolled out into practice and again be validated using the same process as described first.

The primary purpose of the research tool will be to investigate problems in the requirements engineering domain. Its secondary purpose will be to evaluate potential solutions on a theoretical basis prior to the solutions being implemented in an actual organisation.

We are potentially interested in the social and technical architecture of the organisation being investigated. For the moment, this is a curiosity-driven research goal to determine who the different groups, or social actors, are within the organisation and how the social network operates.

To learn more about the social network, the researcher will interview members of the organisation to determine how the organisation functions. The researcher will enquire from the members of the organisation about their experience over time regarding the behaviour of the various parameters identified previously.

This type of research setup will function in a single-case mechanism type experiment. The environment within which it functions is a single complex socio-technical system, which will require mechanism research since it investigates the interaction between different mechanisms in an organisation.

When the research tool is to be used in implementation evaluation or validation research, any potential modification to the way that the requirements engineering process works in the organisation can first be evaluated. This is done on a theoretical level in a laboratory or artificial setting using the research tool prior to it being implemented on a limited scale in the actual organisation. This suggested approach will limit the number of iterations being evaluated in a live socio-technical environment.

Because we restrict ourselves to problem investigation, the object-of-study consists of an artefact prototype interacting with a simulation of the context. The conceptual framework will be based on the original conceptual framework but may need to be expanded if additional issues are discovered in a new problem investigation activity.

The knowledge questions may include any of the following (Wieringa, 2014):

- Effect questions: What effects are produced by the interaction between the artefact prototype and the simulated context? Why?
- Requirements satisfaction questions: Do the effects of the simulation satisfy requirements? Why (not)?
- Trade-off questions: What happens if the artefact architecture is changed? Why?
- Sensitivity questions: What happens if the context is changed? Why?

4.5.2 Single-case mechanism flow diagram

The function of the research tool is to support the research process using a single-case mechanism experiment within the requirements engineering domain. The FFBD of the experiment is shown in Figure 49.

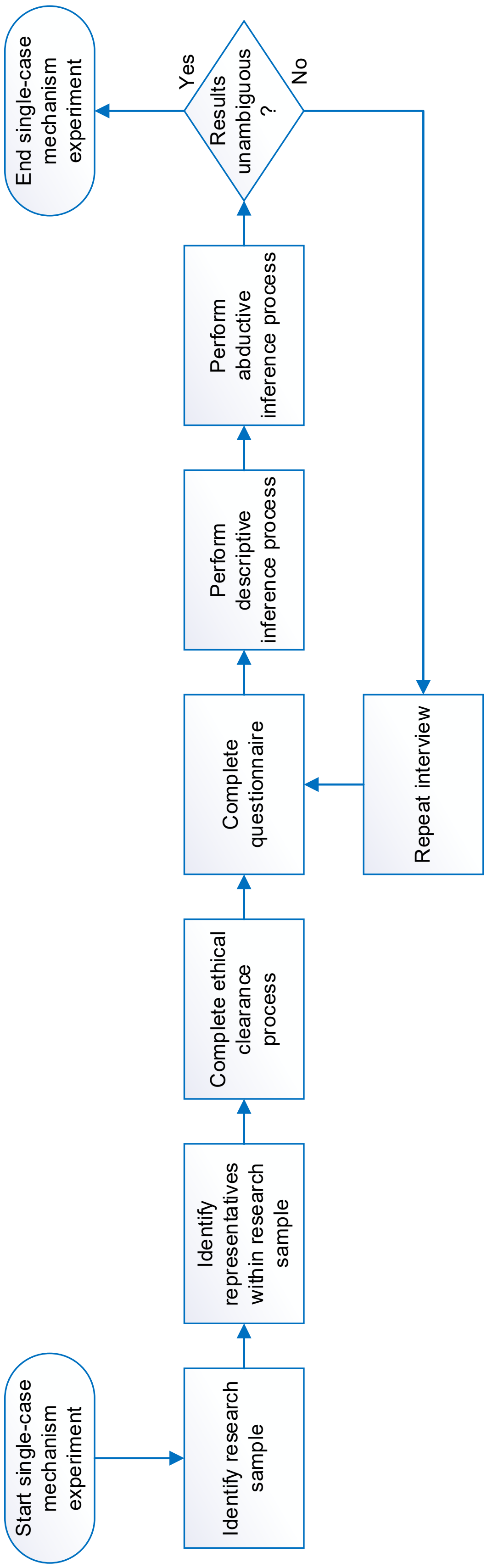


Figure 49: Single-case mechanism experiment FFBD

The descriptions of the different functions are as follows:

1. **Identify the research sample**. The overall research population is discussed in paragraph 3.7. The research sample will be selected from a range of different types of industries that run development projects using some form of requirements engineering process. These industries will include the following:
    a. Healthcare infrastructure industry
    b. Financial industry (banking systems)
    c. Human resources
    d. Space technologies

    The specific samples that will be selected from these sample populations should not just include cases in which the use of the requirements engineering process was unsuccessful or problematic but should also preferably include cases in which the use of the requirements engineering process was successful in making meaningful comparisons. The relationship between the research sample and the research population is shown in Figure 50.

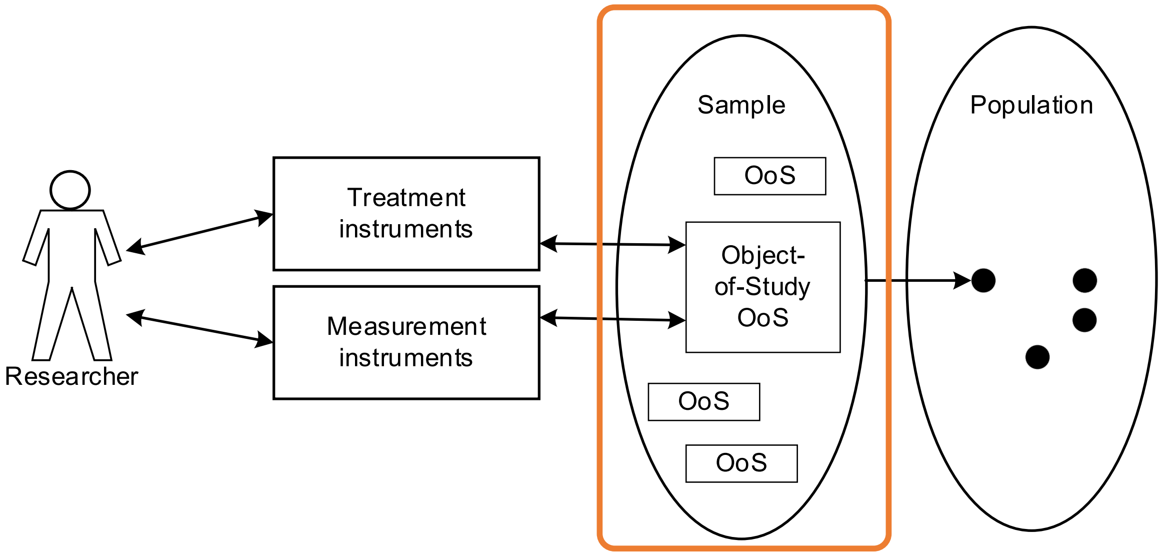


Figure 50: Relationship between the research sample and the research population [redrawn and adapted from Wieringa (2014)]

2. **Identify representatives within the research sample.** Representatives will be identified from the identified research sample. Depending on the overall research stakeholders, these representatives will either be stakeholders directly involved in the process, or SMEs who are involved within the

organisation and who can act as competent surrogates or proxies for the organisation.

3. **Complete ethical clearance process.** One of the main stakeholder requirements identified in Chapter 3 was that the research process must be conducted on an ethical basis. To ensure this, the ethical clearance process as prescribed by the academic institution will be followed. An example of the informed consent form is included in Appendix B.
4. **Complete questionnaire.** The questionnaire will be discussed and completed with the SME or stakeholder who is being interviewed. This interview will take the form of a semi-structured interview and will be conducted in an action research setting. This allows the researcher to ensure that all the aspects of the interview are clear and unambiguous.
5. **Perform descriptive inference process.** The data will be cleaned using a descriptive inference process (Wieringa, 2014). This process will include digitising the data and checking data validity.
6. **Perform abductive inference process.** The data will be analysed using an abductive inference process.
7. **Clarify research results.** Based on the outcome of the abductive inference process, it may be required to clarify some aspects of the research process to date. This may require that some parts of the interview be repeated.
8. **Finalise research experiment.** When all the data has been analysed, the results will be compiled and disseminated.

4.6 Contribution to research tool goals

The research goal identified in paragraph 2.3.1 was split between a knowledge goal and a design goal. Table 18 shows the contribution of the requirements to the goals.

Table 18: Contribution of the research tool requirements to the research goals

Research goals	Research tool requirements
Knowledge goal: To identify and describe the observed phenomena, their interactions and the effect of these interactions as observed in the complex requirements engineering process that may result in the incorrect set of requirements being used in the project implementation.	**R1:** The research tool shall provide a method to identify the parameters that can be used to describe the requirements engineering environment (real world of concern) being researched. **R2:** The research tool shall provide a method to document the relationship between the different parameters that can be used to describe the requirements engineering environment (real world of concern) being researched. **R3:** The research tool shall provide a method for the researcher to measure the current state of the parameters that can be used to describe the requirements engineering environment (real world of concern) being researched.
Design goal: To improve the understanding of the requirements engineering process by designing and implementing a requirements engineering research tool that can be used to investigate and explain the observed phenomena observed in the requirements engineering process.	**R4:** The research tool shall provide a method to model an ideal-world scenario. **R5:** The research tool shall provide a method to process the current state of the parameters of the requirements engineering environment (real world of concern) being researched into a model of the real situation. **R6:** The research tool shall provide a method to compare different ideal models with the "real-world situation of concern". **R7:** The research tool shall support an ethical research process. **R8:** The research tool shall provide a method to evaluate different models of improvement in order to select the best alternative prior to implementation.

4.7 Chapter summary and conclusion

The purpose of this chapter was to establish the requirements and specification for the research tool. This specification was derived from the research tool requirements established in Chapter 3. The requirements of the engineering research tool were defined using the soft systems methodology as the mental model of the processes that must be performed. The identified requirements were analysed in terms of consistency, omissions and ambiguity to arrive at the final set of requirements. The research tool architecture was defined, including the experimental method to be implemented by the research tool. An FFBD of the envisaged operation of the research tool was also derived.

In Chapter 5, the derived requirements and specification for the research tool will be converted into a design of the research tool, which will be realised and verified.

CHAPTER 5. REQUIREMENTS ENGINEERING RESEARCH TOOL DESIGN, IMPLEMENTATION AND VERIFICATION PHASE

"Design is not just what it looks like and feel like. Design is how it works." – Steve Jobs

5.1 Introduction

The purpose of the requirements engineering research tool design, implementation and verification phase is to define the architecture and design of the research tool according to the requirements established previously in Chapter 4. The design will subsequently be realised, and its operation will be verified according to the identified requirements. This research tool will be used in a single-case mechanism experiment (as discussed in paragraph 4.5.1).

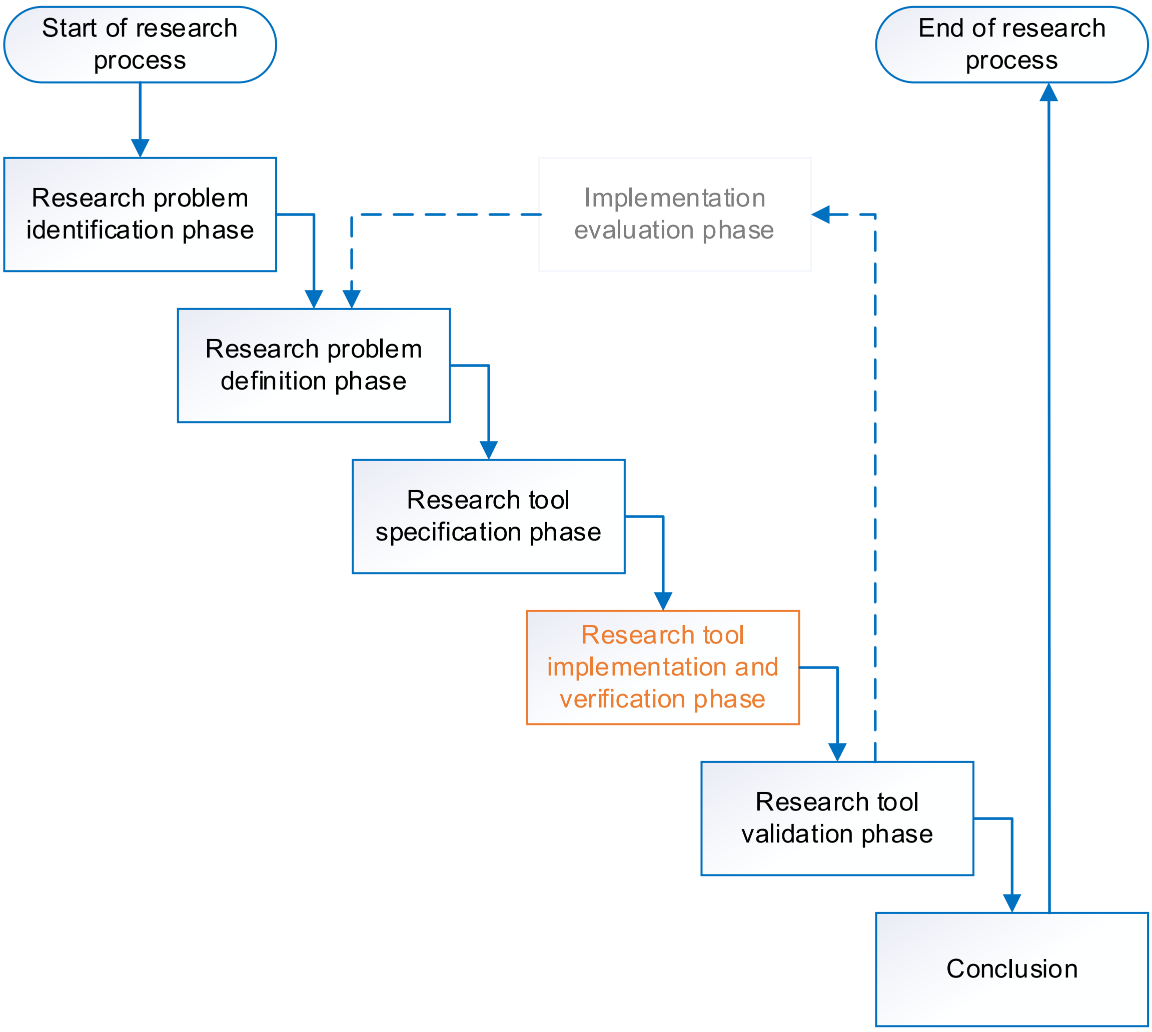


Figure 51: Relative position of the research tool design, implementation and verification phase in the overall research process

The overall research design was discussed in paragraph 1.4 together with the solution implementation and verification phase in paragraph 1.4.4. Figure 51 above shows the relative position of the research tool design, implementation and verification phase in the overall research process.

The primary functions of the research tool implementation and verification phase are to: (a) design the research tool; (b) implement the research tool, and (c) verify the research tool. The functions are shown schematically in the FFBD depicted in Figure 52.

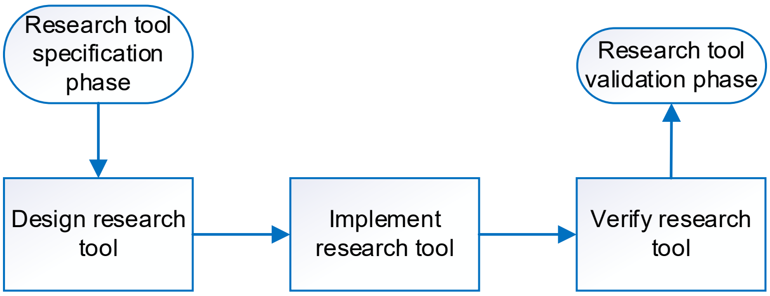


Figure 52: Research tool implementation and verification phase FFBD

This phase includes activities from the design definition process, system analysis process, implementation process, integration process and the verification process as defined in the INCOSE Systems Engineering Handbook (INCOSE, 2015).

The research tool may consist of one or more building blocks or elements that are integrated to create the realised system (product or service) that satisfies the identified requirements, architecture, and design.

5.2 Single-case mechanism experiment design specifications

The single-case mechanism experiment was discussed in detail in paragraph 4.5.2 along with a flow diagram as shown in Figure 49. This research method defines the environment and the boundaries within which the research tool is used.

A single-case mechanism experiment is specified as it supports the test of a specific mechanism within a single object-of-study with a defined architecture (Wieringa, 2014). The purpose of the experiment is to explain the cause-and-effect behaviour of the object-of-study in terms of its architecture (Wieringa, 2014).

The research setup for a single-case mechanism experiment is shown in Figure 53. The researcher interacts with the object-of-study in two ways. In the first instance, the researcher measures the behaviour of different parameters that describe parts of the object-of-study. This is done via a questionnaire that is completed during a semi-structured interview in an action research setting.

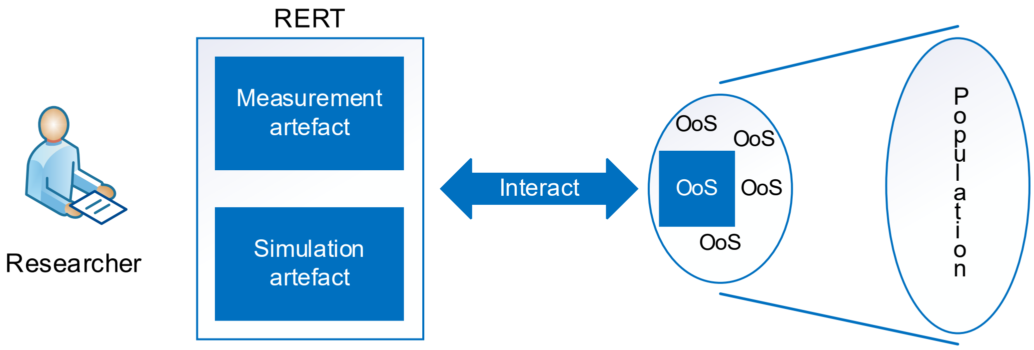


Figure 53: Single-case mechanism research setup (own contribution)

In the second instance, the researcher uses the measurement results from the interviews and evaluate them by simulating the requirements engineering process to identify possible cause-and-effect behaviours that could affect requirements volatility. Requirements volatility was defined in paragraph 2.4.3.4 as the change in the requirements set due to changed, missed and deleted requirements.

The primary purpose of the single-case mechanism experiment is to assist the researcher in determining the contribution of the various parameters or combination of parameters identified previously on the requirements volatility. Table 19 shows the design specifications for the single-case mechanism that can be identified:

Table 19: Single-case mechanism experiment design specification

No.	Specification
SC-1	The single-case mechanism experiment shall support the specified data analysis and inference design.
SC-2	The single-case mechanism experiment shall be repeatable.
SC-3	The single-case mechanism experiment shall adhere to the ethical guidelines as are applicable to this research.
SC-4	The single-case mechanism experiment shall support the necessary validation models, sampling requirements, treatments, and measurement.

5.3 Requirements engineering research tool design specification

The research tool requirements, as presented in paragraph 4.2.2, can be grouped into three broad categories. The first grouping, Category 1, relates to the tool requiring an interaction with the stakeholder or SME. The requirements allocated to this category are R1, R2 and R3. The second grouping, Category 2, relates to the processing and analysis of the data that has been collected. The requirements allocated to this category are R4, R5, R6 and R8. The last grouping, Category 3, refers to the characteristics of the research method. The requirement allocated to this category is R7.

Figure 54 shows the architecture of the research tool. It consists of two artefacts that can be used interchangeably. These two artefacts are the measurement artefact related to requirements R1, R2 and R3, and the simulation artefact related to requirements R4, R5, R6 and R8. The overall research process was conducted within the prescription of requirement R7.

The function of the measurement artefact is to provide a means to identify, elicit and capture the behaviour of the various parameters of the problem context within the object-of-study. The purpose of the simulation artefact is, firstly, to provide a means to model a real-world scenario and, secondly, for the researcher to experiment with the behaviour of the various parameters in the context of the architecture of the object-of-study. These two artefacts were used within an ethical research environment.

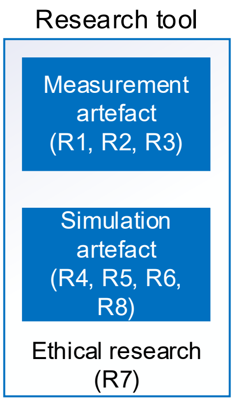


Figure 54: Requirements engineering research tool architectural structure (own contribution)

5.4 Measurement artefact design, implementation and verification

5.4.1 Measurement artefact design specifications

The primary data collection tool for this study is a research discussion document. This document is completed by the SME in an action research setting. The research discussion document includes the following sections:

5.4.1.1 Stakeholder or SME background information

The purpose of the background section is to capture relevant information of the stakeholder or the SME being interviewed to establish their fields of expertise and their level of experience within the specific domain being researched. The collection of data was done according to the ethical research guidelines applicable to this research.

The following information was collected:

1. Background experience of the stakeholder or the SME in their field of expertise and the number of years of experience in each of the individual fields.
2. The typical project characteristics of a small, medium and large project regarding the number of requirements, total project duration, project definition, and design duration as is relevant to their area of expertise.
3. The split of requirements between the different sources of requirements.

5.4.1.2 Establish completeness of the factors that may influence requirements volatility

The causal loop diagram shown in Figure 34 was included in the research discussion document. This causal loop diagram was used to show the already identified parameters and the already identified interaction between the various parameters. The researcher used this diagram to initiate the discussion with the stakeholder or the SME regarding the applicable parameters, the unnecessary parameters, and the interaction between the various parameters.

5.4.1.3 Establish the behaviour over time of the factors that may influence requirements volatility

The research discussion document provided a brief premise of the research problem being investigated and the research process being followed. Instructions for completing the questionnaire were also provided.

The stakeholder or the SME was requested to evaluate the behaviour of the parameters identified in Table 20 in terms of their behaviour over a time period for the following categories:

- The expected elicitation behaviour
- The expected error creation behaviour
- The expected error detection behaviour

Missing parameters that were identified during the initial discussions were added to the list as required.

Table 20: Parameters that may affect the outcome of the requirements engineering process

Project environment related	Stakeholder characteristics
• Project type, size and complexity • Number of stakeholders • Existing or legacy system requirements • Project duration • Number of requirements	• Domain knowledge • Technical knowledge • Stakeholder misinformation • Stakeholder's nature • Understanding the need • Resistance to change • Changes in technology
Requirements engineer characteristics	**Stakeholder and requirements engineer interaction**
• Domain knowledge • Technical knowledge • Requirements engineer misinformation • Requirements engineer's experience • Requirements engineering's personality	• Social nature • Communication • Conflict • Trust

5.4.2 Measurement artefact implementation

5.4.2.1 Function of questionnaire

The function of the questionnaire was to capture the results of the interview between the researcher and the stakeholder being interviewed. The research questionnaire consisted of two parts. The first part focused on capturing relevant background information regarding the research subject in relation to the interview. This included the name of the person being interviewed, the date of the interview, the fields of experience of the subject, and the number of years of experience in the field. Table 21 shows the information that was captured.

Table 21: SME information required

SME information	
Name	**Interview date**
Field of experience	**Years in field**
High-technology commercial systems	
High-technology military systems	

Financial systems	
Commercial systems	
Medical system	
Transport systems	

The second part of the information to be captured pertained to typical requirements that form part of a small, medium and large project. The estimated total duration of the project, as well as the estimated duration of the project definition and design phase, was also captured. Table 22 shows the information that was captured.

Table 22: Estimation of the number of requirements and typical project duration

Estimation of the number of requirements and typical project duration

Field:

Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects			
Medium projects			
Large project			

The third part of the questionnaire elicited information regarding the distribution of requirements originating from the various sources including documentation, legacy systems and stakeholders. Table 23 shows the information that was elicited.

Table 23: Split of requirements between the different sources

The split of requirements between the different sources		
Field:		
Source	**Distribution (%)**	**Comments**
Documentation		
Legacy system requirements		
Stakeholders		

5.4.2.2 Establish completeness of the factors that may influence requirements volatility

The completeness of the identified factors and the envisaged interaction between these parameters were established based on a review of the causal loop diagram that was previously shown in Figure 34.

The following brief premise of the research problem being investigated, and the research process being followed was included in the research discussion document:

> *"The first stage of the requirements engineering process is to convert the needs of the stakeholders using a discovery and elicitation process into unstructured requirements. In its purest form, this should be a linear process where all the undiscovered needs are converted into unstructured requirements over a period at a fixed elicitation rate. What is, however, found in the real world is that the discovery and elicitation process is not perfect. Errors can be introduced due to many different factors, including those identified in Table 14. These errors will result in requirements being incorrectly discovered and elicited, which will, in turn, require correction at a later stage in the requirements engineering process that will have an impact on the cost, schedule and performance constraints of the project."*

The questionnaire provided a means for the research subject to provide their response to the stated questions. This was in the form of a graph. The following responses were provided: (a) the expected elicitation behaviour vs time; (b) expected error creation vs time; and (c) expected error detection vs time.

Provision was made for the respondent to indicate the applicability in terms of one of the following:

Table 24: Applicability of the supplied behavioural data

Applicability		
Stakeholders	Documentation	Legacy systems
Missed	Deleted	Changed

Provision was made for a response to the questions as listed in Table 25: If necessary, a short description was provided of each parameter to enlighten the subject. Provision was made for some blank response forms in case of additional elements were identified during the interview process that needed to be recorded.

Table 25: Behavioural question responses

Groups	Elements
Project related	**Project type:** Bespoke (tailored or customised) vs an off-the-shelf system.
	Project size: Value, geographical distribution and the number of requirements needed to describe the need of the stakeholders.
	Project complexity: A project that may use an off-the-shelf type solution vs a system that may require critical safety features.

Groups	Elements
	Number of stakeholders: The number of stakeholders who represent the need and are involved in the project.
	Number of requirements: The number of requirements used in the implementation of the project.
	Existing or legacy system requirements: The need to upgrade or replace an existing system to adapt to new business challenges or changes in the technological landscape.
	Project duration: The influence of project duration and time pressure to complete the requirements engineering process.
	Changes in technology: The effect of changes in technology. Change in technology has been identified as one of the significant challenges in the requirements engineering/elicitation process. This is due to the domain and technical knowledge not being relevant or applicable anymore. Solutions that worked previously may no longer be relevant.
Stakeholder characteristics	**Domain knowledge:** The influence of the amount of the domain knowledge that the stakeholder possesses. Domain knowledge is knowledge of the nature of the business and the needs that must be solved. Not only does this include familiarity with the operation of the business but it also extends to the knowledge that describes the nature and culture of the organisation, including specific terminology and abbreviations that may be used within the organisation.
	Technical knowledge: The influence of the amount of technical knowledge that the stakeholder possesses. Technical knowledge includes the fundamentals of the various technologies that are in play in the domain, as well as knowledge of items such as development methodologies and tools that can be used.

Groups	Elements
	Stakeholder misinformation: The effect of stakeholder misinformation. The requirements engineer may not have access to or include the actual end user (the guy on the ground) who will be using or operating the system. In such a case, the external specialist or consultant may only interact with, or have access to middle or senior management who may have a perception of what the needs are but do not have actual first-hand experience. Stakeholder misinformation may also occur when certain representatives of the customer are trying to influence the outcome of the requirements elicitation process in a specific direction. This characteristic may also be related to the implementation independence and attainability of the requirements.
	Stakeholder's nature: The effect of the stakeholder's nature on the requirements engineering process. As individuals, each stakeholder has their own perspective and perception of what the need is. However, as stakeholders and thus as representatives of the end users, they may have different concerns, priorities and responsibilities.
	Resistance to change: The effect of the stakeholder's resistance to change on the requirements engineering process.
Requirements engineer characteristics	**Domain knowledge:** The effect of the requirements engineer's domain knowledge on the requirements engineering process. If requirements engineers do have the required domain knowledge, they should avoid fixation and preconceptions that may lead to an incomplete understanding of the stakeholder's needs.
	Technical knowledge: The effect of the technical knowledge of the requirements engineer on the requirements elicitation process.

Groups	Elements
	Requirements engineer introduced misinformation: The effect of the requirements engineer misinformation on the requirements engineering process. This misinformation effect may lead to the stakeholder recalling misinformation that may have been introduced by the requirements engineer rather than recalling their true beliefs and knowledge of the fact. The overall effect of this misinformation is reducing the correctness of the requirements elicited. This characteristic may also be related to implementation independence and attainability of the requirements.
	Requirements engineer's experience: The effect of the requirements engineer's experience on the requirements engineering process.
	Requirements engineer's personality: The effect of the requirements engineer's personality on the requirements engineering process. The personality of the requirements engineer must also be such that the person is able to interact with the stakeholders to elicit requirements effectively.
Stakeholder – requirements engineer interaction	**Social nature:** The effect of the social nature of requirements elicitation. The requirements elicitation process involves collaboration between the various stakeholders and those responsible for the requirements elicitation process. During this collaboration process, knowledge regarding the system requirements is shared and discussed to construct a shared understanding of the requirements. This collaboration and knowledge sharing within the requirements elicitation process can be characterised as problematic since the various groups contribute distinct kinds of knowledge and experience to this activity, and due to this, trust among the different parties cannot be guaranteed.

Groups	Elements
	Communication: The effect of good or inadequate communication between the stakeholder and the requirements engineer. Communication plays a vital role in the requirements elicitation process. In addition to the standard communication aspects inherent in the human and social nature of requirements elicitation, specific problems may be encountered should projects be executed over international borders. In these types of projects, the project teams responsible for the requirements elicitation process may not only have to deal with a lack of face-to-face communication but also with other issues such as different time zones and cultural diversity. This may lead to misunderstanding and even conflict in the process.
	Conflict: The effect of conflict between the requirements engineer and the stakeholder. Conflict is a common occurrence in a situation in which group interaction is present. Aspects that may introduce conflict in the requirements elicitation process are most likely to be found when there is limited domain knowledge present, fluctuating or conflicting requirements, and a breakdown in communication. However, coordination conflict also plays a decisive role in the requirements elicitation process in the sense that it can be used as a tool to counter resistance to change and to counter stagnation in the elicitation process.
	Trust: The effect of trust or lack of trust between the stakeholder and the requirements engineer.

The responses of the SME were captured on the response worksheets shown in Figure 55.

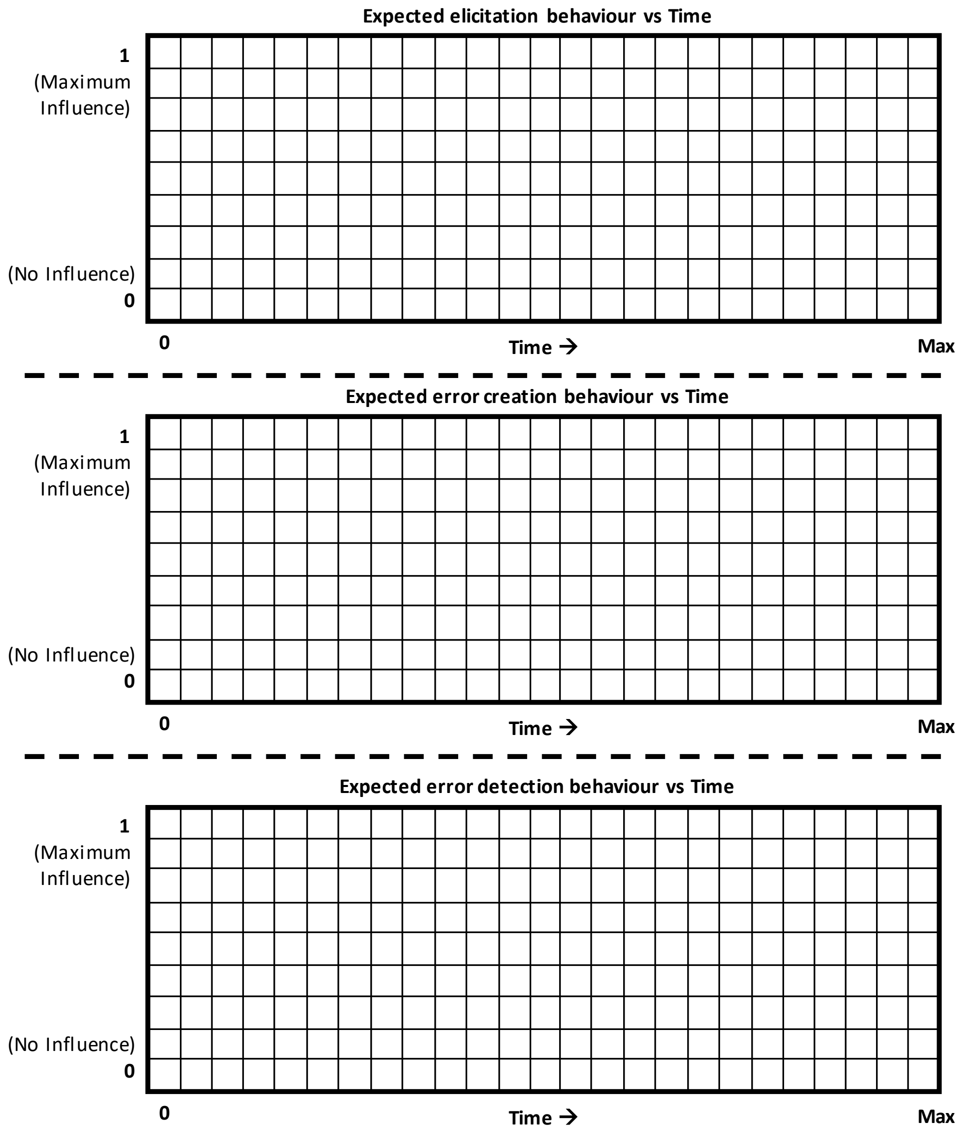


Figure 55: SME response worksheet (own contribution)

5.4.3 Measurement artefact verification

The measurement artefact was verified in two ways. The first verification was performed by reviewing the completeness of the behavioural parameters as shown in Table 25. The review was conducted by interviewing two SMEs with specific experience in the requirements engineering domain. The results of the verification exercise were the following:

5.4.3.1 Comments from the first SME

The purpose of interviewing the first SME was more exploratory in nature. A table was compiled prior to the interview that mapped the requirements challenge characteristic to the different problem areas identified from the literature and personal experience. The purpose of this interview was to confer with an SME regarding this table and discuss the items to get an understanding from an industry expert on these matters.

Table 26: Comments and suggestions for improvement from the first requirements engineering SME

Comments and suggestions for improvement	Review of comment
The following comments were made during this exploratory interview:	
1. What is the difference between domain knowledge and understanding of the need? Is it possible that understanding of the need is a combination of domain knowledge and technical knowledge?	Domain knowledge is seen as the knowledge a stakeholder has regarding the working and operational environment. Understanding of the need relates to the operational requirement that the product or system must fulfil. Technical knowledge is seen as a broader technical understanding of what is possible.
2. Look at including aspects such as corporate strategic position under the understanding of the need topic. In other words, do not only look at the understanding of the need from a technical viewpoint but also from a business viewpoint.	To be included with the stakeholder's domain knowledge topic description.
3. What is the relationship between technical knowledge and changes in technology?	Technical knowledge is about the specific field within which the delivered system must operate in. Changes in technology will affect technical knowledge. It is about what is now possible.
4. Missing topic: A project sponsor, project custodian or project champion is crucial. Need the buy-in and support from the project champion. Look at the role of the project sponsor vs the project owner.	The project sponsor is an important element in the overall execution of the project. The role of the project sponsor in the elicitation process is less dominant.

Comments and suggestions for improvement	Review of comment
5. Stakeholder's nature. More detail required. Also, look at cross-cultural influences and language differences.	Topic description to be expanded with cross-cultural influences and language differences as discussion points.
6. Requirements engineer characteristics: The following parameters are missing: a. Ability to create order from chaos. Is this a capability that is inherent in the person or can it be acquired? b. How quickly does the requirements engineer grasp the problem or the issue? This is also related to the rate of gaining domain knowledge.	To be included as part of the requirements engineer experience discussion topic.
7. Stakeholder – requirements engineer interaction. Trust should be added as a parameter. What are the aspects that influence trust?	To be included in research discussion document.
8. Project environment? a. Project sponsor on the stakeholder side. b. Project related: What type of project are we looking at? • A greenfield, stand-alone project? • A stand-alone project, integrated into an existing system? c. Market pull or technology push-type system.	To be included in the project type topic description.
9. Project complexity is equivalent to different integration points.	Noted.
10. What will the effect be of changing the stakeholder or requirements engineer mid-way during a project?	Interesting observation. Could possibly be used as a research topic in the future.

5.4.3.2 Comments from the second SME

Table 27: Comments and suggestions for improvement from the second requirements engineering SME

Comments and suggestions for improvement	Review of comment
2. Stakeholder – requirements engineer interaction: a. Conflict → Conflicting interest between contracting parties. b. Individual → Slower requirements, not articulated. c. Influence of consultant.	Noted.
3. Project related: Contract type → Examples can be cost-plus, fixed price, phase project.	To be included in the project type topic description.
4. The amount of new technology required: a. Maybe form part of the type of project. b. Integration projects context. c. Development project context.	To be included in the project type topic description.
5. Stakeholder characteristics: No comment.	Noted.
6. Requirements engineer characteristics: Resistance to change – very valid characteristic.	Noted.
7. Project methodology: Consider scrum and agile methodologies to be considered.	Noted.

5.5 Simulation artefact design, implementation and verification

The design, implementation and verification of the simulation artefact are addressed in the following sections.

5.5.1 Simulation artefact design specifications

The design of the simulation artefact is done using a systems dynamic simulation model as determined in paragraph 4.4.5.

A systems dynamics model consists of a number of elements that include cause-and-effect diagrams, stocks (quantities that accumulate/de-accumulate over time) and flows (the rate of change of stocks), feedback, delays and non-linearity [Forrester, J (1999) quoted by Smit, Brent and Musango (2014)].

The purpose of the simulation artefact is to understand the dynamics and interactions of the various factors that may influence the requirements elicitation phase of a project.

Figure 56 shows the dynamic hypothesis for the requirements discovery and elicitation process. The undiscovered need of the stakeholder is transformed into unstructured requirements via the requirements elicitation process. This elicitation process is however not perfect, as controlled by the elicitation behaviour parameter, resulting in incorrect unstructured requirements being captured erroneously. These errors will manifest in the form of volatile requirements. These volatile requirements will only be detected at a later stage in the requirements engineering process in the form of requirements that were missed and must be added, requirements that were duplicated and must be deleted, or requirements that were elicited incorrectly and must be corrected. After detecting the volatile requirements, the incorrectly elicited requirements will be removed from the unstructured requirements and added back to the undiscovered need to be processed via the discovery and elicitation process again.

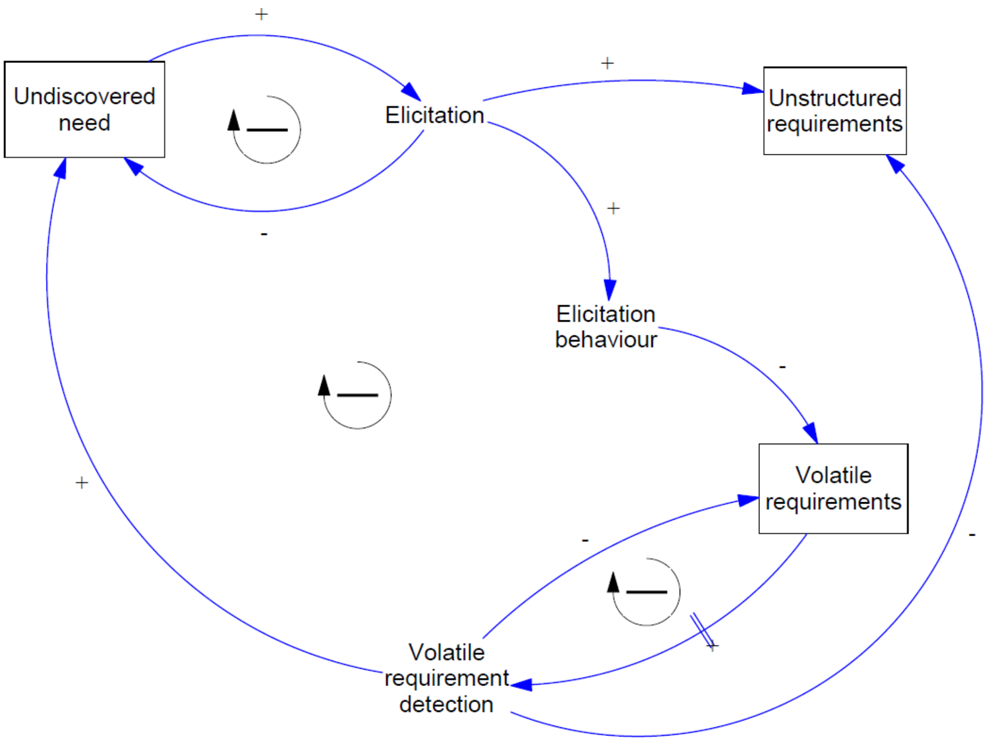


Figure 56: Dynamic hypothesis model for a single-stage requirements engineering (own contribution)

The graphical results of the measurement artefact were prepared and converted to numerical data using a digitisation process as part of the descriptive inference activities.

5.5.2 Simulation artefact implementation

The simulation artefact was implemented based on elements of the dynamic hypothesis as shown in Figure 56. The model verification and validation were done in a gradual way to build up confidence in the results and boundaries of the simulation model. This process started at the model conceptualisation stage and continued even after implementing the results (Barlas, 1994). Figure 57 shows the system dynamics simulation model.

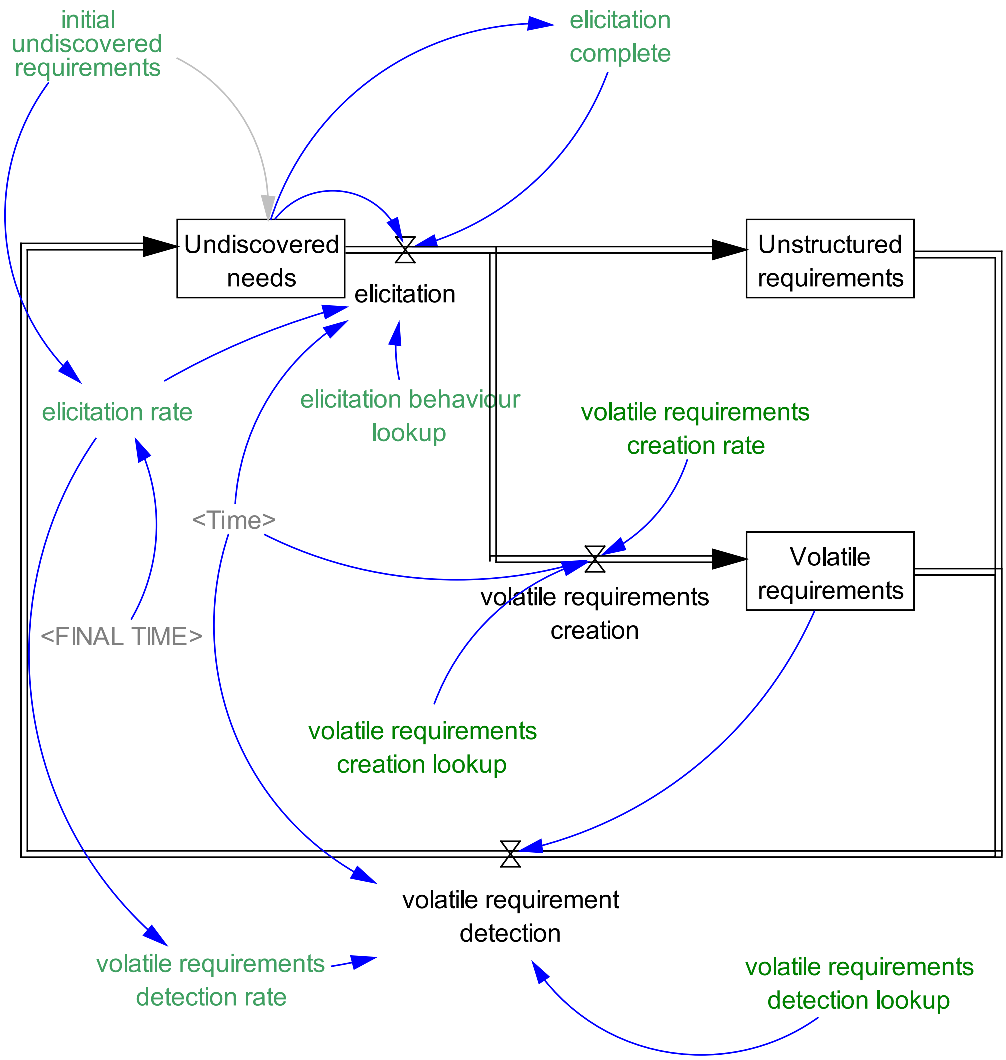


Figure 57: Measurement artefact system dynamics simulation model (own contribution)

The elements of the simulation model are the following:

1. Undiscovered needs: This stock element represents the yet to be discovered needs of the stakeholder.
2. Unstructured requirements: This stock element represents the unstructured requirements of the stakeholder that were obtained via the elicitation process (flow element) from the undiscovered needs.
3. The behaviour of the elicitation flow element is a function of the elicitation behaviour obtained as part of the interviews with the SMEs. The elicitation rate can be determined from the number of requirements that the SME typically expects will define the product or service being designed divided by the typical time available for the definition phase of the project.
4. As was described in the dynamic hypothesis, the elicitation process is not perfect, and errors are made during the process. The creation of these errors is controlled by the volatile requirement creation flow element. The behaviour of the volatile requirement flow element is controlled via the error creation behaviour rate and the error creation behaviour lookup table. The incorrectly elicited requirements are accumulated in the volatile requirement stock and is subtracted at the same time from the unstructured requirements stock. The parameters for the volatile error creation lookup function is obtained from the digitised results of the questionnaire.
5. The error detection process operates on the volatile requirement stock via the volatile requirement flow element. The behaviour of the volatile requirements flow element is controlled via the error detection rate parameter and the error detection lookup function. Once the errors are detected, they are added back into the undiscovered needs stock element. The parameters for the error detection lookup function are obtained from the digitised results of the questionnaire.

Table 28 shows the mathematical equations for the different elements.

Table 28: Mathematical equations for the systems dynamic simulation model

Simulation model element	Element type	Equation
Undiscovered needs	Stock element	$\int (volatile\ requirements\ detection - elicitation)$ *Initial value = initial undiscovered requirements*
Unstructured requirements	Stock element	$\int (elicitation - volatile\ requirement\ detection)$ *Initial value = 0*

Simulation model element	Element type	Equation
Volatile requirements	Stock element	$\int(volatile\ requirement\ creation - volatile\ requirement\ detection)$ *Initial value = 0*
Elicitation	Flow element	*IF THEN ELSE [elicitation complete, 0, MIN (elicitation behaviour lookup (Time) × elicitation rate, Undiscovered needs)]*
Volatile requirements creation	Flow element	*elicitation × error creation behaviour lookup (Time) × (1 − error creation rate)*
Volatile requirements detection	Flow element	*MIN [Volatile requirements, error detection rate × error detection lookup (Time)]*
Elicitation behaviour lookup	Lookup table	
Volatile requirements creation lookup	Lookup table	
Volatile requirements detection lookup	Lookup table	
Initial undiscovered requirements	Constant	
Elicitation rate	Variable	*Initial undiscovered requirements/ Final time*
Elicitation complete	Control variable	
Time	System variable	
Final Time	System variable	

5.5.3 Simulation artefact verification

The solution verification approaches were previously discussed in paragraph 1.4.4, and a list of design evaluation methods was presented in Table 5.

The primary verification method that was used for the system dynamics simulation model was a black box testing approach. This testing was done by evaluating known scenarios using known inputs and evaluating the outputs.

5.5.3.1 Verification Scenario 1

For the first verification scenario, a simple simulation model using a constant elicitation rate and a constant error creation rate was used for concept demonstration and evaluation purposes.

Table 29: Simulation parameters for Verification Scenario 1

Parameter	Value
initial undiscovered needs	1000 requirements
elicitation rate	100 requirements/day
volatile requirements creation rate	0
volatile requirements detection rate	10 requirements/day
elicitation complete	Control variable
Simulation time step	0.0625
Simulation end time	50 days

Figure 58 to Figure 60 show the results for Verification Scenario 1. Since the error creation rate was 0 (perfect), the undiscovered needs were all converted into unstructured requirements at a rate of 100 requirements per day and completed after ten days. The model was designed so as not to be able to elicit more requirements than were available, resulting in the gradual rounding of the graph as the undiscovered needs approached 0 and the unstructured requirements approached 1000.

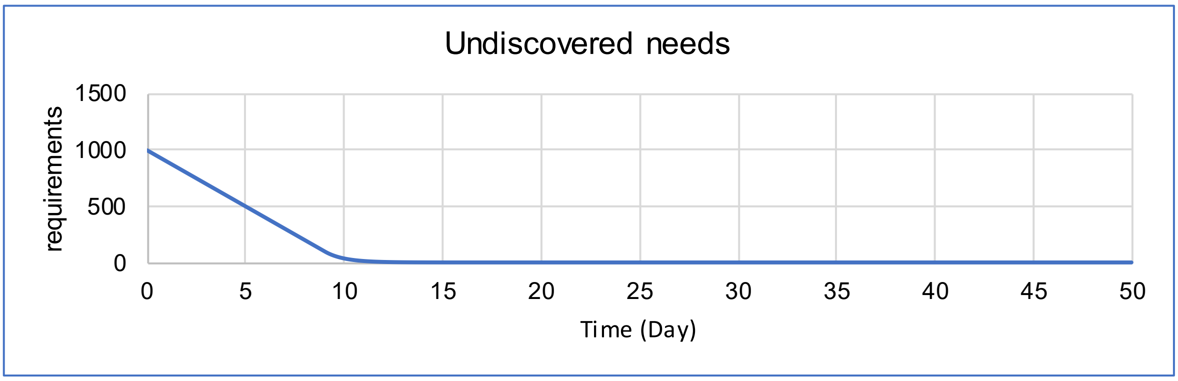


Figure 58: Undiscovered needs from Verification Scenario 1

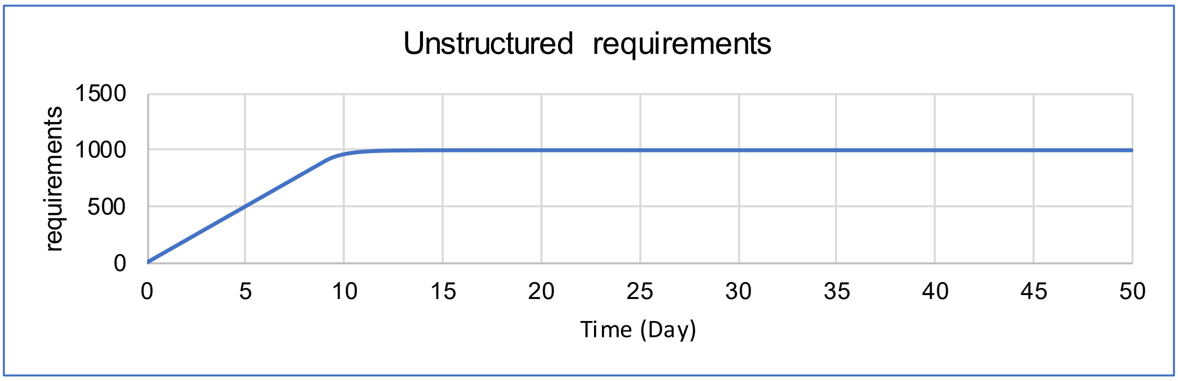


Figure 59: Unstructured requirements from Verification Scenario 1

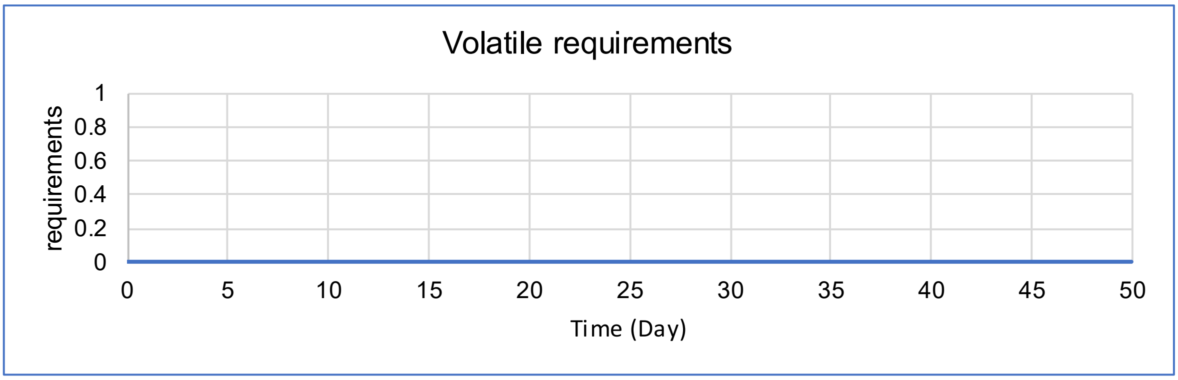


Figure 60: Volatile requirements from Verification Scenario 1

5.5.3.2 Verification Scenario 2

The purpose of the second scenario was to investigate the behaviour of the model if the elicitation process was not perfect, resulting in requirements that would need to be changed at some point in the future. The same basic simulation model as shown in Figure 57 was used with the parameters as shown in Table 30.

Table 30: Simulation parameters for Verification Scenario 2

Parameter	Value
initial undiscovered needs	1000 requirements
elicitation rate	100 requirements/day
volatile requirements creation rate	0.8
volatile requirements detection rate	10 requirements/day
elicitation complete	Control variable
Simulation time step	0.0625
Simulation end time	50 days

The results for Verification Scenario 2 are shown in Figure 61 to Figure 63. The behaviour for Verification Scenario 2 shows the effect of eliciting incorrect requirements with the error creation parameter is set to 0.8, resulting in 80% of the elicited parameters being classified as volatile requirements. These requirements were removed from the unstructured requirements and added back to the undiscovered needs, from where they were once again detected at a rate of 100 requirements/day, resulting in the behaviour shown in Figure 63. The elicitation process was only completed after approximately 25 days (when the volatile requirements reduced to 0), which was more than double the results from the ideal case. The volatile requirements peaked at a maximum value of 102 at approximately 11 days after which it declined to 0 after approximately 25 days.

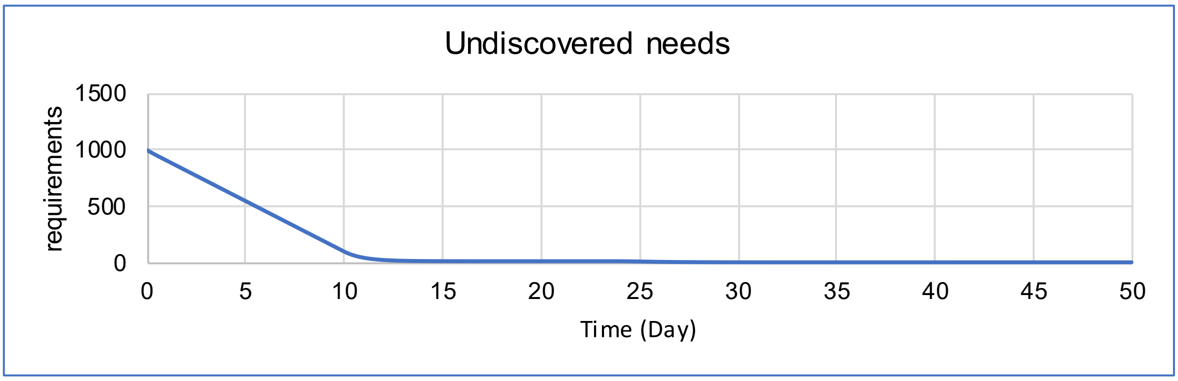


Figure 61: Undiscovered needs from Verification Scenario 2

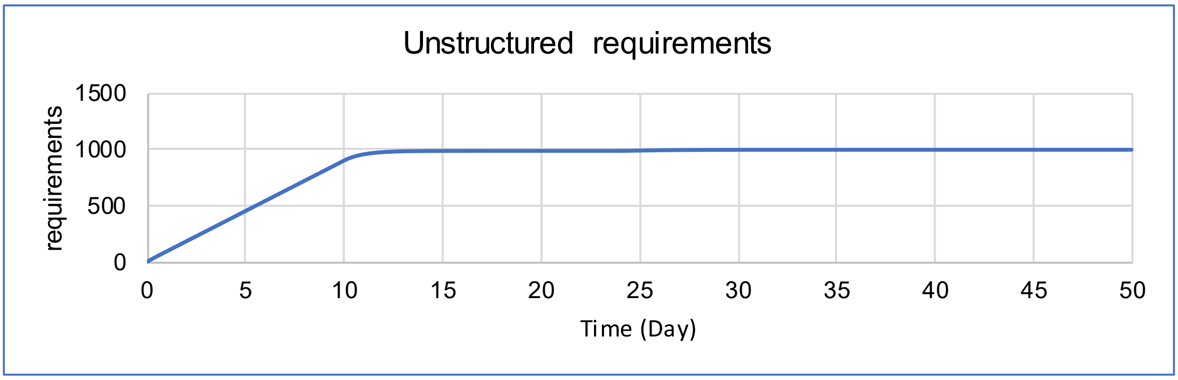


Figure 62: Unstructured requirements from Verification Scenario 2

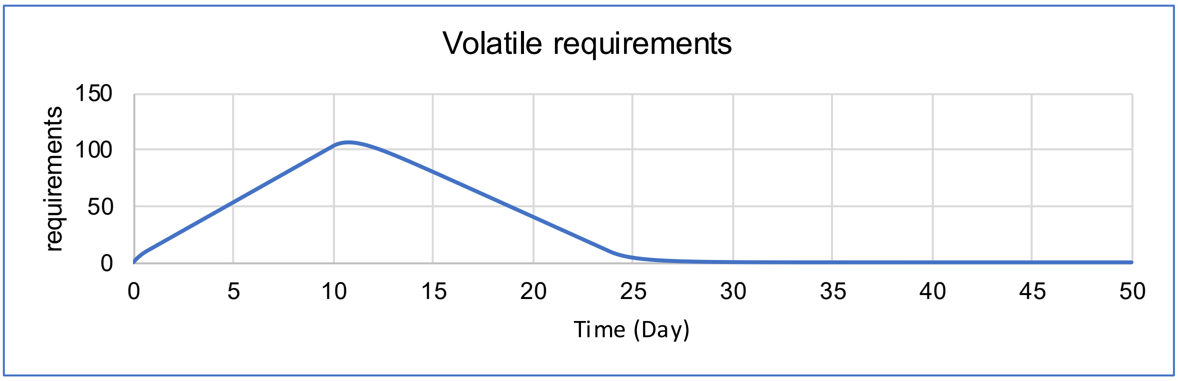


Figure 63: Volatile requirements from Verification Scenario 2

Another practical insight obtained from the model was that because the elicitation process was not perfect (error creation rate = 80%), it could be expected that the requirements that changed as a result of the imperfect elicitation process would peak at some point and then start to reduce. This knowledge can enable the requirements engineer to track the number of changed requirements in practice and use it to estimate when the elicitation process will be completed.

5.5.3.3 Verification Scenario 3

In the first two verification scenarios, the elicitation process was modelled as a constant rate. In the third verification scenario, the elicitation process was modelled as a variable that changed over time. This type of scenario is most likely to be seen in practice. The elicitation behaviour was modelled in such a way as to simulate a low elicitation rate at the start of the process, and then gradually increased to the maximum elicitation rate after a period. The effect was to slow the elicitation process initially before allowing it to reach the maximum value.

The simulation parameters for Verification Scenario 3 are shown in Table 31. The transfer function for the parameter elicitation behaviour lookup is shown in Figure 64. The simulation results for Verification Scenario 3 are shown in Figure 65 to Figure 67.

Verification Scenario 3 was expanded by adding a mechanism to explore non-linear behaviour in the simulation model. This non-linear behaviour was introduced in the elicitation behaviour lookup function using a lookup table (refer Figure 64) that modelled the effect of the parameter over time. The effect of the non-linearity can clearly be seen in the subsequent behaviour of the different stock parameters in that their behaviour is no longer linear.

Table 31: Simulation parameters for Verification Scenario 3

Parameter	Value
initial undiscovered needs	1000 requirements
elicitation rate	100 requirements/day
volatile requirements creation rate	0.8
volatile requirements detection rate	10 requirements/day
elicitation behaviour lookup	Lookup table implementing Figure 64
elicitation complete	Control variable
Simulation time step	0.0625
Simulation end time	50 days

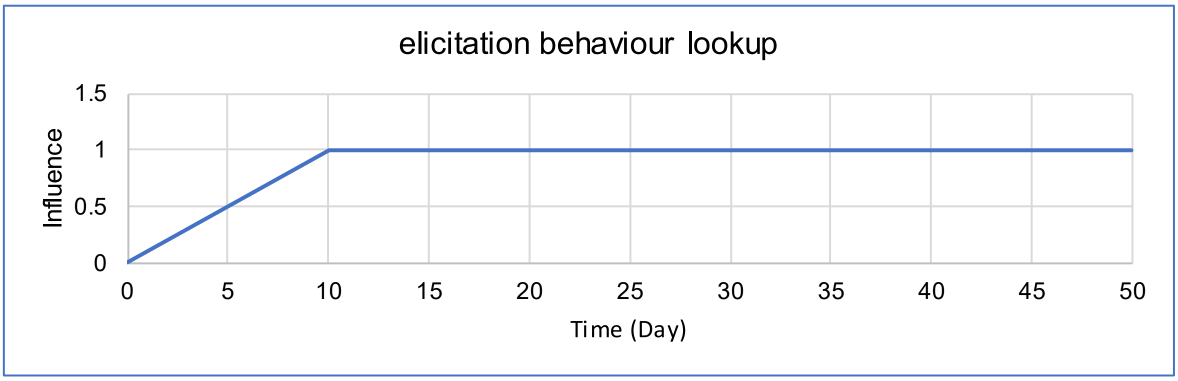


Figure 64: Lookup function for elicitation behaviour

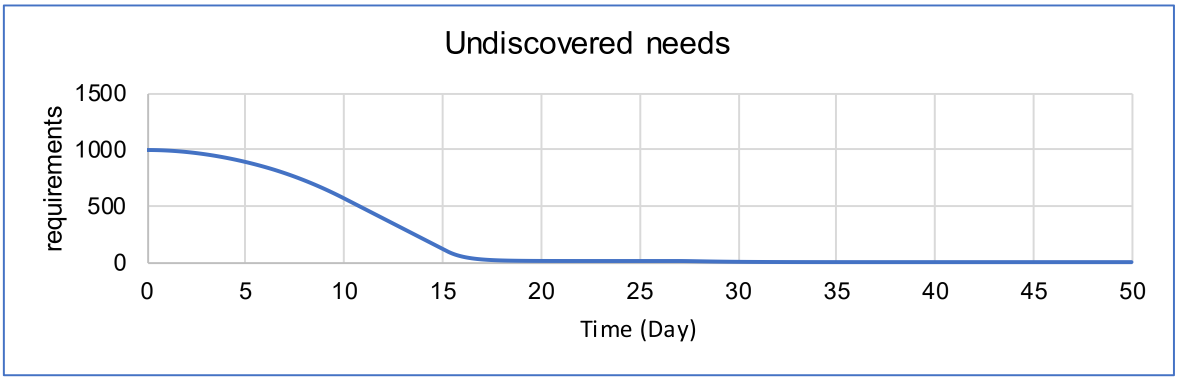


Figure 65: Undiscovered needs from Verification Scenario 3

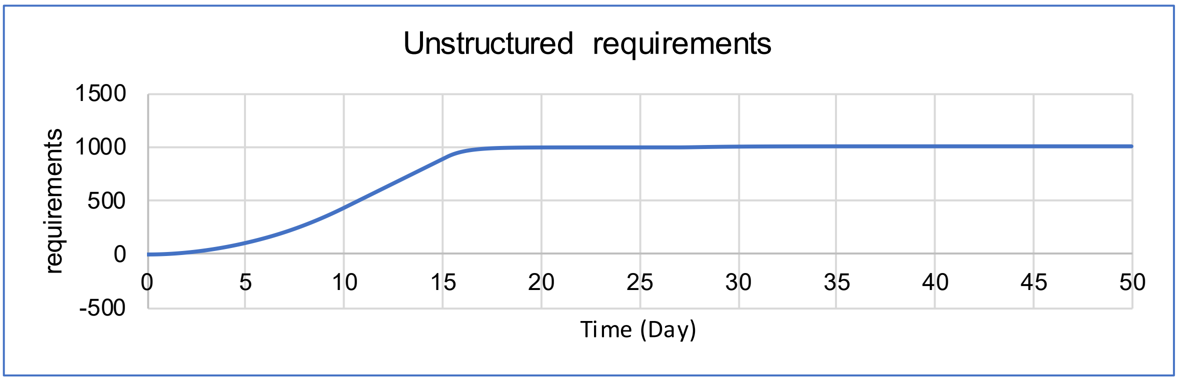


Figure 66: Unstructured requirements from Verification Scenario 3

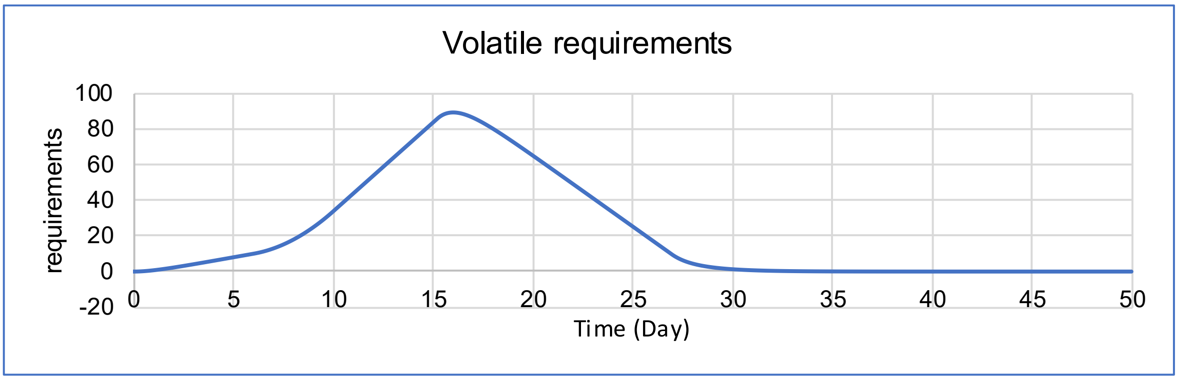


Figure 67: Volatile requirements from Verification Scenario 3

5.6 Evaluation of the verification of the research tool requirements

The following research tool requirements were identified in paragraph 4.2.2:

R1. The research tool shall provide a method to identify the parameters that can be used to describe the requirements engineering environment (real world of concern) that is being researched.

R2. The research tool shall provide a method to document the relationship between the different parameters that can be used to describe the requirements engineering environment (real world of concern) that is being researched.

R3. The research tool shall provide a method for the researcher to measure the current state or current behaviour of the parameters that can be used to describe the requirements engineering environment (real world of concern) that is being researched.

R4. The research tool shall provide a method to model an ideal-world scenario.

R5. The research tool shall provide a method to process the current state of the parameters of the requirements engineering environment (real world of concern) that is being researched into a model of the real-world situation.

R6. The research tool shall provide a method to compare different ideal models with the "real-world situation of concern".

R7. The research tool shall support an ethical research process.

R8. The research tool shall provide a method to evaluate different models of improvement in order to select the best alternative prior to implementation.

The verification process that was discussed in this chapter was aimed at verifying the aspects as is shown in Figure 68.

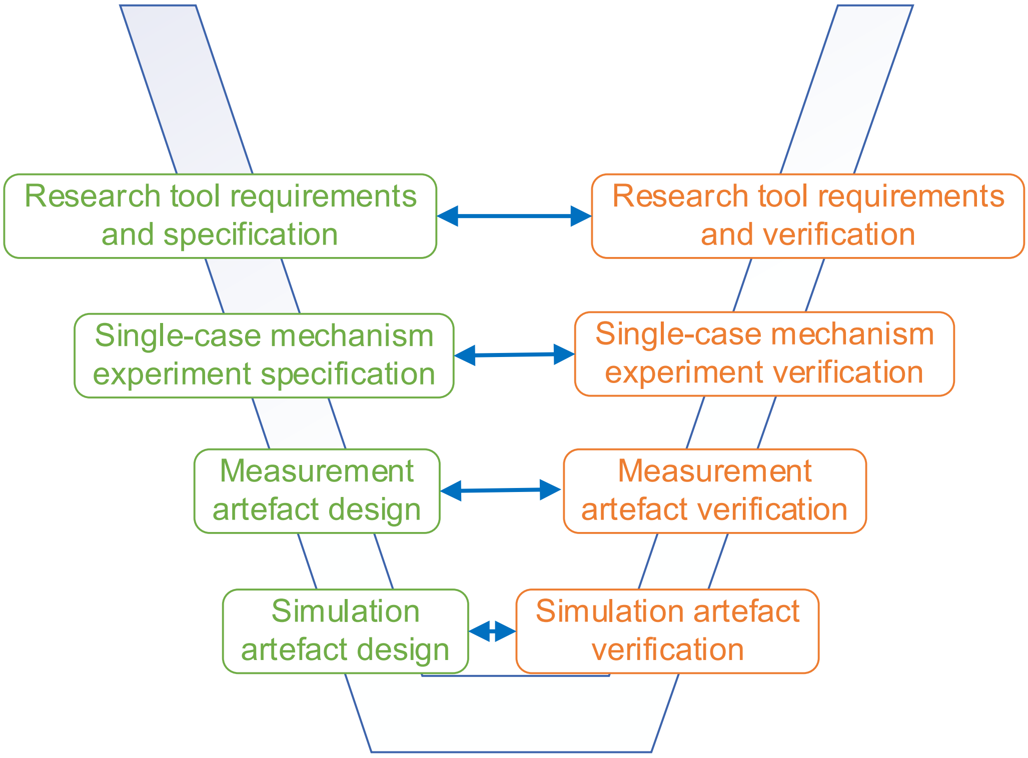


Figure 68: Research tool verification activities (own contribution)

The mapping of the research tool requirements presented in paragraph 4.2.2 and the single case mechanism specification presented in 5.2 compared to the verification activities is shown in Table 32.

Table 32: Mapping of the research tool requirements to the verification references

Research tool requirements (Rn) and single-case mechanism experiments (SC-n)	Verification reference
Measurement artefact (R1, R2, R3), (SC-1)	Paragraph 5.4.3
Simulation artefact (R4, R5, R6, R8), (SC-2, SC-4)	Paragraph 5.5.3

Research tool requirements (Rn) and single-case mechanism experiments (SC-n)	Verification reference
Ethical research (R7), (SC-3)	7.7Appendix A. 7.7Appendix B.

5.7 Chapter summary and conclusion

The purpose of this chapter was the design, implementation and verification of the requirements engineering research tool.

The design, implementation and verification activities of the research tool were discussed in this chapter. The research tool consists of two artefacts, namely, the measurement artefact and the simulation artefact. The design of the research tool was based upon the requirements established in Chapter 4.

The measurement artefact that was designed and implemented consisted of a questionnaire to be completed by the SME or stakeholder. The questionnaire was verified by reviewing it with the various SMEs as part of the semi-structured interview process.

The simulation artefact that was designed and implemented consisted of a systems dynamic simulation model that used as inputs the results of the semi-structured interview that will be conducted as part of the research tool validation phase. The verification of the simulation artefact that was performed was done by evaluating specific scenarios with a known outcome. The results of the verification activities confirmed that the simulation artefact perform to expectations on a mathematical level.

The final part of this chapter traced the verification activities back to the research tool requirements, thus proving that all the requirements were verified.

The next step in the overall research process will be to validate the research tool. This validation process will use the results of the semi-structured interview that were conducted with the different SMEs to perform actual single-case mechanism experiments. The validation process of the research tool will be discussed in Chapter 6.

CHAPTER 6. REQUIREMENTS ENGINEERING RESEARCH TOOL VALIDATION PHASE

"Learn from yesterday, live for today, hope for tomorrow. The important thing is to never stop questioning." – Albert Einstein

6.1 Introduction

The research tool will be validated by performing several research cycles as shown in Figure 49 using the results of the data obtained during the interviews of the SMEs. The research tool validation phase was also discussed in some detail in paragraph 1.4.5.

Figure 69 shows the relative position of the research tool validation phase in the overall research process.

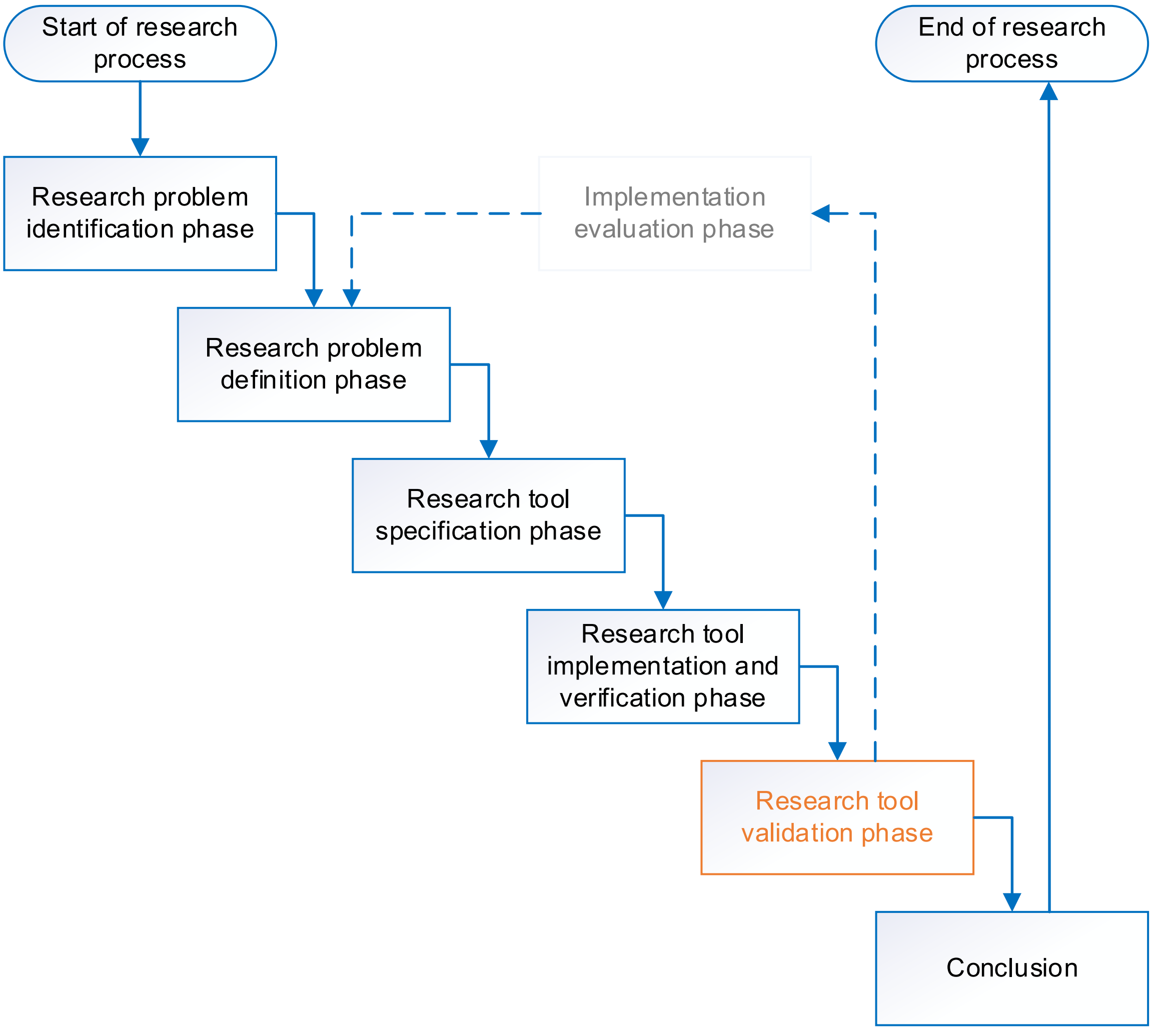


Figure 69: Relative position of research tool validation phase in overall research process

The purpose of the validation process is to provide proof that the research tool, when used within the research context identified previously, meets the stakeholder needs and requirements within the intended research environment (INCOSE, 2015; NASA, 2017). This will be done by observing and measuring how well the artefact supports the solution to the stated problem. This activity will involve comparing the objectives of the solution with the actual observed results by using the artefact in a demonstration. In order to achieve this, a knowledge of relevant metrics and analysis techniques identified as part of the inference design is required (Hevner *et al.*, 2004).

The solution validation phase can also be used to evaluate the performance of different research tool artefacts when more than one was identified initially, evaluate the sensitivity of the research tool, and quantify the contribution of the research tool to the knowledge goals and improvement goals as identified at the start of the study.

The results generated will be analysed and explained using causal, architectural, or rational reasoning methods. The general validity of the results will be examined to determine if the methods used and the results obtained are transferable to similar cases or populations (Wieringa, 2014). In the end, the results obtained should answer the knowledge questions posed during the research design process, including a summary of the conclusions as well as the limitations of the conclusions (Wieringa, 2014).

The validation process is shown in the form of an FFBD in Figure 70, which includes the following steps:

1. Defining and analysing reference modes to establish guidelines for analysing the results.
2. Performing single-case mechanism experiments with the identified SMEs.
3. Analysing the results using descriptive and abductive inference process to arrive at explanations of the phenomena being observed.
4. Evaluating the results against the identified goals, needs and requirements.

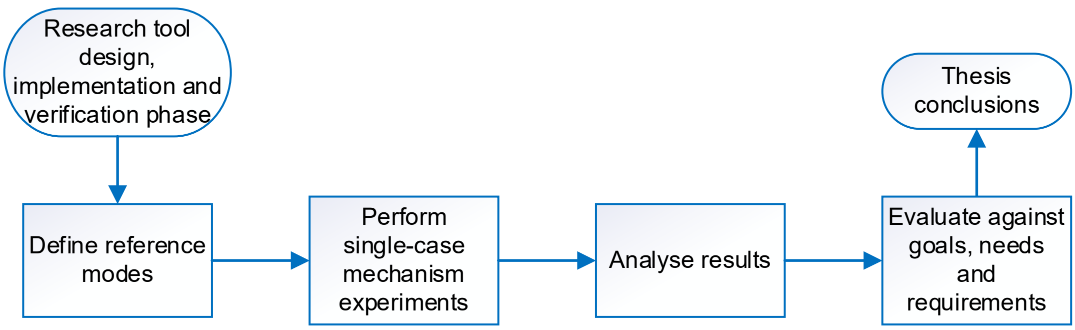


Figure 70: Research tool validation phase FFBD

6.2 Ideal-world behaviour and reference modes

In order to analyse the results of the parameters presented in Appendix C, a guideline is required that can serve as a baseline against which a comparison can be made. This baseline is also known as a reference mode when dealing with system dynamics models.

The reference mode that can serve as a baseline for the elicitation, error creation and error detection behaviour is one that is based on the growth models or susceptibility-infection models. These models are used to model a wide variety of situations such as technology adoption, diffusion and spread of infectious diseases (Duggan, 2017; Sterman, 2000; Thun, Größler and Milling, 2000).

The basic elicitation-diffusion model is shown in Figure 71. The model consists of two levels, namely, undiscovered needs and unstructured requirements. Undiscovered needs are converted to unstructured requirements via an elicitation process at a specific elicitation rate. In the elicitation-diffusion model, UR_0 represents the initial unstructured requirements and N the total population of requirements. P_r represents the probability that when an unstructured requirement is elicited from the undiscovered needs pool, the stakeholder(s) involved in the process will be triggered or reminded of an additional requirement.

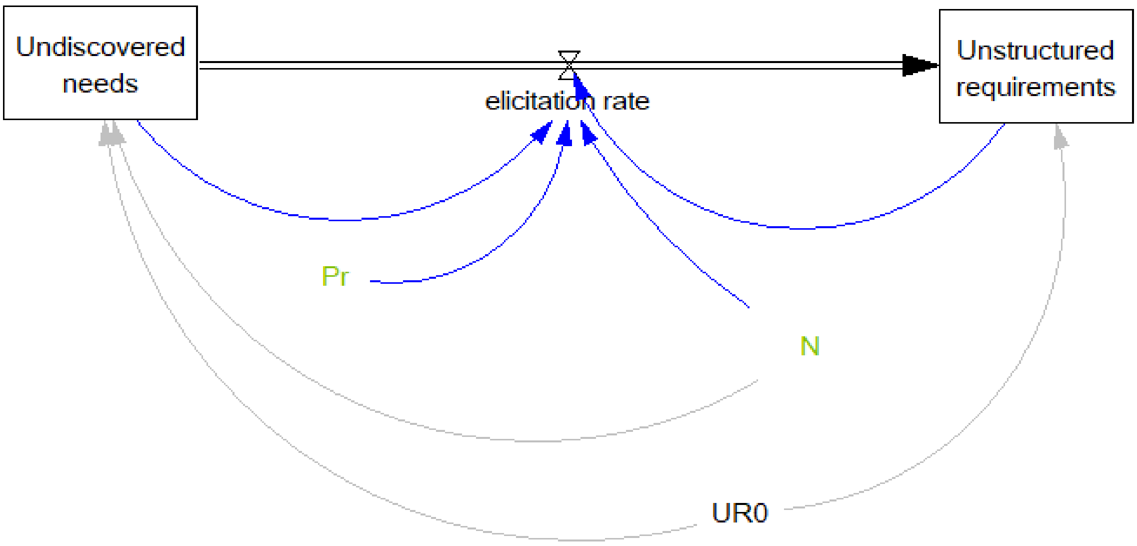


Figure 71: Elicitation-diffusion model (own contribution)

Table 33 shows the equations for the elicitation-diffusion model.

Table 33: Elicitation-diffusion model equations

Simulation model element	Element type	Equation/description
Undiscovered needs	Stock element	$\int(-elicitation\ rate)$ *initial value = N − UR₀*
Unstructured requirements	Stock element	= $\int(elicitation\ rate)$; *initial value* = UR_0
Elicitation rate	Flow element	P_r × *Undiscovered needs* × *(Unstructured requirements/N)*
UR_0	constant	*number of unstructured requirements at* T_0
N	Constant	*initial undiscovered needs*
P_r	Constant	*probability of reminding*

Using this model, insight into what the potential elicitation behaviour will be can be generated. The parameters that can be varied are the initial unstructured requirements, UR_0, the total population, N, and the probability of reminding, P_r. Figure 72 to Figure 75 show the results for different combination of these parameters. The primary interest lies with the shape of the graph, the relative time when the maximum elicitation rate occurs, and the relative peak of the elicitation rate.

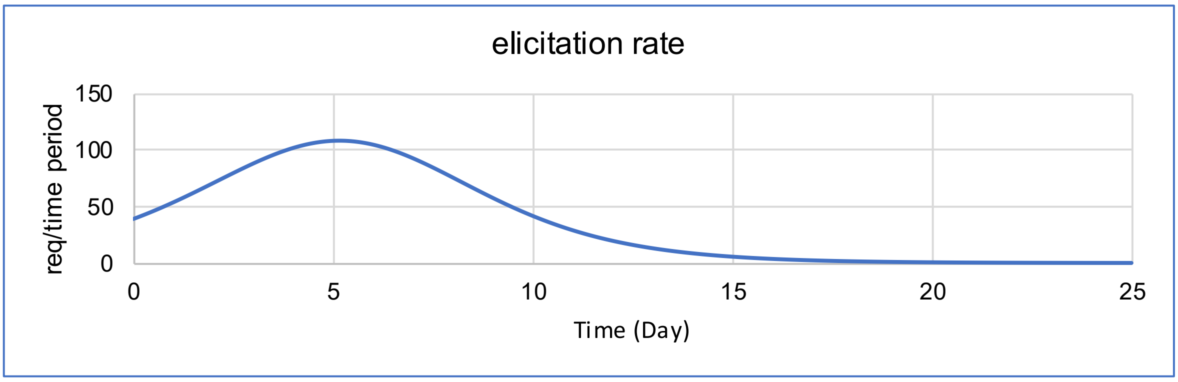


Figure 72: Elicitation behaviour for N = 1000, UR0 = 100 and Pr = 0.431

In all cases illustrated in Figure 72 to Figure 75, the total undiscovered needs population was kept constant at N = 1000. Figure 72 examined a hypothetical case in which UR_0 was set at a value of 100, i.e. 10% of the total requirements. This would typically be the case if a larger amount of prior experience or knowledge exists within the organisation, which means that the stakeholders can provide an initial set of requirements quicker.

The case illustrated in Figure 73 used the same parameters as previously with the exception that the initial set of requirements was $UR_0 = 10$. This implied that the elicitation process started at a lower base and, correspondingly, that the peak elicitation rate only occurred later in the overall elicitation process.

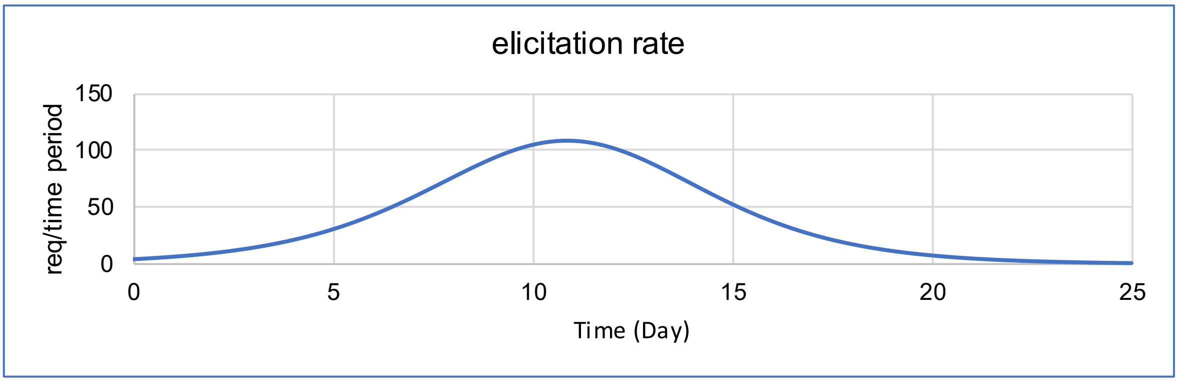


Figure 73: Elicitation behaviour for N = 1000, UR_0 = 10 and P_r = 0.431

The case illustrated in Figure 74 used the same parameters as for Figure 73 with the exception that the probability of reminding was lower at a value of $P_r = 0.261$. The implication was that the elicitation process was slower with the peak only occurring near the end of the process.

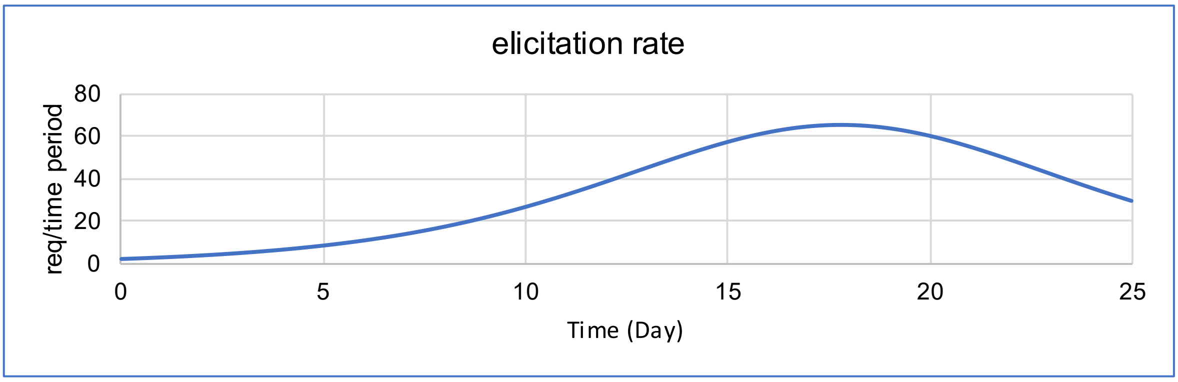


Figure 74: Elicitation behaviour for N = 1000, UR_0 = 10 and P_r = 0.261

The final example illustrated in Figure 75 used the same parameters as for Figure 74 with the exception that the probability of reminding was much higher at a value of $P_r = 0.961$. The implication was that the elicitation process was much quicker with the peak only occurring near the beginning.

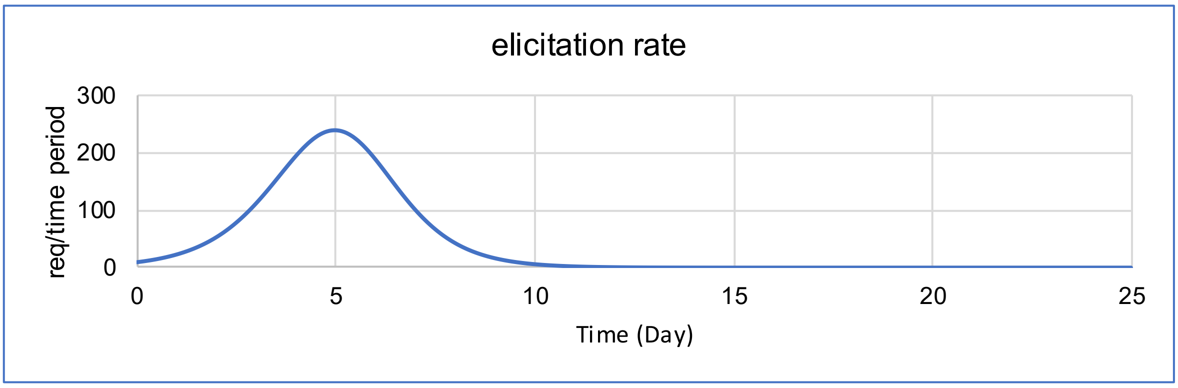


Figure 75: Elicitation behaviour for N = 1000, UR_0 = 10 and P_r = 0.961

It was postulated earlier that the errors that occur during the elicitation process are created at the same time as the requirements are elicited. These errors are in (a) the form of duplicate requirements that are elicited and will have to be deleted at some point in the future; (b) missed requirements that will have to be added at some point in the future; or (c) incorrectly elicited requirements that will have to be changed at some point in the future.

It can thus be postulated that the behaviour of the error creation process will have a similar behaviour although the level may be lower. In a similar way, it can be postulated that the behaviour of the error detection process will have a similar behaviour but, in this case, both the level and the timing of when this activity occurs will be affected.

One can formulate the following guidelines for the requirements engineer during the abductive inference process:

1. The relative location of the local peak for the elicitation behaviour, error creation and error detection, is an important aspect to consider. The later the peak occurs in the process, the more significant indication it is of a problem situation in the elicitation process. This is due to the less time available to complete the elicitation process.
2. The late occurrence of the local peak during the elicitation process can also be an indication of a delay that may exist within the overall requirements engineering process.
3. The relative height of the local peak is another essential factor to consider. The lower the peak is, the fewer requirements are being elicited. Once again, the time constraint can play an essential role in creating pressure during the requirements engineering process.

6.3 Single-case mechanism experiments

Paragraph 4.5.2 discussed the research process and the research tool FFBD. The sample industries identified included the healthcare infrastructure industry, financial industry (banking sector), human resources, and space technologies. The reason for selecting these industries was to investigate objects-of-study across the high-technology industry population and not to focus on one industry only.

Four SMEs were identified from within the sample objects-of-study (one per object-of-study), and the ethical clearance process was completed with them. The next step in the single-case mechanism experiment was to conduct the semi-structured interviews with the SMEs according to the questionnaire defined in paragraph 5.4. For the purposes of anonymity, these SMEs are identified in this research as SME-1 through to SME-4. The SMEs did not in all cases provide their insight and opinion to all the questions contained in the questionnaire. This was due to the response either being the same as a previous response or that they did not feel they had an opinion about the specific topic.

6.3.1 Summary of the interviewed results

A summary of the interview results is provided in Table 34 to Table 37 and with the detailed results provided in Appendix C.1 to Appendix C.4. The four SMEs provided around 55 response between them.

Table 34: Summary of SME-1 interview results

Nr	Group	Sub-group	Reference
1	Project related	Project type	Figure C-1 to Figure C-3
2	Project related	Project size	No response provided
3	Project related	Project complexity	No response provided
4	Project related	Number of stakeholders	No response provided
5	Project related	Existing or legacy stakeholders	No response provided
6	Project related	Project duration	Only a response for error creation provided in Figure C-4
7	Stakeholder characteristic	Domain knowledge	Figure C-5 to Figure C-7
8	Stakeholder characteristic	Technical knowledge	Behaviour expected to be the same as domain knowledge

Nr	Group	Sub-group	Reference
9	Stakeholder characteristic	Stakeholder misinformation	Figure C-8 to Figure C-9
10	Stakeholder characteristic	Stakeholder's nature	No response provided
11	Stakeholder characteristic	Resistance to change	No response provided
12	Stakeholder characteristic	Changes in technology	No response provided
13	Requirements engineer characteristics	Domain knowledge	Figure C-10 to Figure C-12
14	Requirements engineer characteristics	Technical knowledge	No response provided
15	Requirements engineer characteristics	Requirements engineer misinformation	No response provided
16	Requirements engineer characteristics	Experience	No response provided
17	Requirements engineer characteristics	Personality	No response provided
18	Stakeholder and requirements engineer interaction	Social nature	Figure C-13 to Figure C-15
19	Stakeholder and requirements engineer interaction	Communication	No response provided
20	Stakeholder and requirements engineer interaction	Conflict	No response provided
21	Stakeholder and requirements engineer interaction	Trust	No response provided

Table 35: Summary of SME-2 interview results

Nr	Group	Sub-group	Reference
1	Project related	Project type	Figure C-16 to Figure C-18
2	Project related	Project size	Figure C-19. Error creation and error detection behaviour as for Project Type.
3	Project related	Project complexity	No response provided
4	Project related	Number of stakeholders	No response provided
5	Project related	Existing or legacy stakeholders	Not applicable to this study
6	Project related	Project duration	Elicitation behaviour will be like the behaviour of Project Type. Figure C-20 for error creation behaviour and Figure C-21 for error detection behaviour.
7	Project related	Project budget	Figure C-22 to Figure C-24
8	Stakeholder characteristic	Domain knowledge	Figure C-25 to Figure C-27
9	Stakeholder characteristic	Technical knowledge	No response provided
10	Stakeholder characteristic	Stakeholder misinformation	Figure C-28 to Figure C-30
11	Stakeholder characteristic	Stakeholder's nature	Behaviour expected to be the same as for stakeholder misinformation
12	Stakeholder characteristic	Resistance to change	Figure C-31 to Figure C-33
13	Stakeholder characteristic	Changes in technology	No response provided
14	Requirements engineer characteristics	Domain knowledge	No response provided
15	Requirements engineer characteristics	Technical knowledge	No response provided

Nr	Group	Sub-group	Reference
16	Requirements engineer characteristics	Requirements engineer misinformation	No response provided
17	Requirements engineer characteristics	Experience	No response provided
18	Requirements engineer characteristics	Personality	No response provided
19	Stakeholder and requirements engineer interaction	Social nature	No response provided
20	Stakeholder and requirements engineer interaction	Communication	Figure C-34 to Figure C-36
21	Stakeholder and requirements engineer interaction	Conflict	No response provided
22	Stakeholder and requirements engineer interaction	Trust	Figure C-37 to Figure C-39

Table 36: Summary of SME-3 interview results

Nr	Group	Sub-group	Reference
1	Project related	Project type	Figure C-40 to Figure C-42
2	Project related	Project size	Figure C-43 to Figure C-45
3	Project related	Project complexity	Figure C-46 to Figure C-48
4	Project related	Number of stakeholders	Figure C-49 to Figure C-51
5	Project related	Existing or legacy stakeholders	Figure C-52 to Figure C-54
6	Project related	Project duration	Figure C-55 to Figure C-57
7	Stakeholder characteristic	Domain knowledge	Figure C-58 to Figure C-60
8	Stakeholder characteristic	Technical knowledge	Figure C-61 to Figure C-63

Nr	Group	Sub-group	Reference
9	Stakeholder characteristic	Stakeholder misinformation	Figure C-64 to Figure C-66
10	Stakeholder characteristic	Stakeholder’s nature	Figure C-67 to Figure C-69
11	Stakeholder characteristic	Resistance to change	Figure C-70 to Figure C-72
12	Stakeholder characteristic	Changes in technology	Figure C-73 to Figure C-75
13	Requirements engineer characteristics	Domain knowledge	Figure C-76 to Figure C-78
14	Requirements engineer characteristics	Technical knowledge	Figure C-79 to Figure C-81
15	Requirements engineer characteristics	Requirements engineer misinformation	Figure C-82 to Figure C-84
16	Requirements engineer characteristics	Experience	Figure C-85 to Figure C-87
17	Requirements engineer characteristics	Personality	Figure C-88 to Figure C-90
18	Stakeholder and requirements engineer interaction	Social nature	Figure C-91 to Figure C-93
19	Stakeholder and requirements engineer interaction	Communication	Figure C-94 to Figure C-96
20	Stakeholder and requirements engineer interaction	Conflict	Figure C-97 to Figure C-99
21	Stakeholder and requirements engineer interaction	Trust	Figure C-100 to Figure C-102
22	Ad-hoc	Agile / iterative vs waterfall development	Figure C-103 to Figure C-105

Table 37: Summary of SME-4 interview results

Nr	Group	Sub-group	Reference
1	Project related	Project type	Figure C-106 to Figure C-108
2	Project related	Project size	No response provided
3	Project related	Project complexity	No response provided
4	Project related	Number of stakeholders	No response provided
5	Project related	Existing or legacy stakeholders	No response provided
6	Project related	Project duration	Figure C-109 to Figure C-111
7	Stakeholder characteristic	Domain knowledge	Figure C-112 to Figure C-114
8	Stakeholder characteristic	Technical knowledge	Behaviour expected to be the same as for domain knowledge
9	Stakeholder characteristic	Stakeholder misinformation	Figure C-115 to Figure C-117
10	Stakeholder characteristic	Stakeholder’s nature	Figure C-118 to Figure C-120
11	Stakeholder characteristic	Resistance to change	Behaviour expected to be the same as for stakeholder misinformation
12	Stakeholder characteristic	Changes in technology	Figure C-121 to Figure C-123
13	Requirements engineer characteristics	Domain knowledge	Behaviour expected to be the same as for stakeholder domain knowledge
14	Requirements engineer characteristics	Technical knowledge	No response provided
15	Requirements engineer characteristics	Requirements engineer misinformation	No response provided
16	Requirements engineer characteristics	Experience	Figure C-124 to Figure C-126

Nr	Group	Sub-group	Reference
17	Requirements engineer characteristics	Personality	Behaviour expected to be the same as for stakeholder domain knowledge, technical knowledge and experience
18	Stakeholder and requirements engineer interaction	Social nature	Figure C-127 to Figure C-129
19	Stakeholder and requirements engineer interaction	Communication	Figure C-130 to Figure C-132
20	Stakeholder and requirements engineer interaction	Conflict	Figure C-133 to Figure C-135
21	Stakeholder and requirements engineer interaction	Trust	Behaviour expected to be the same as for requirement's engineer personality

6.3.2 Subject Matter Expert 1

Table 38 summarises the professional experience of SME-1. The raw data results of the interview conducted with SME-1 are contained in Appendix C.1. The highlighted area indicates the field of expertise for SME-1.

Table 38: Summary of data elicited for SME-1

Name SME-1	**Interview date** 13 April 2017
Field of experience	**Years in field**
High-technology commercial systems	
High-technology military systems	
Financial systems	
Commercial systems	
Medical system (infrastructure and health technologies)	20 years
Transport systems	
Estimation of the number of requirements and typical project duration	

Field:	**Medical systems (infrastructure and health technologies)**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects	300	±18 months	±4 months
Medium projects	1000	±4 years	±1 years
Large projects	2000	±10 years	±2 years
The split of requirements between the different sources			
Field:		**Health technologies**	
Source		**Distribution (%)**	**Comments**
Documentation		20%	
Legacy system requirements		5%	
Stakeholders		75%	

6.3.3 Subject Matter Expert 2

Table 39 summarises the professional experience of SME-2. The raw data results of the interview conducted with SME-2 are contained in Appendix C.2. The highlighted area indicates the field of expertise for SME-2.

Table 39: Summary of data elicited for SME-2

Name SME-2			**Interview date** 11 June 2017
Field of experience			**Years in field**
High-technology commercial systems			18 years
High-technology military systems			
Financial systems			
Commercial systems			
Medical system			
Transport systems			
Estimation of the number of requirements and typical project duration			
Field:	**High-technology commercial systems**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**

Small projects	40	±12 months	±3 months
Medium projects			
Large projects	200	±18 months	±3 months
The split of requirements between the different sources			
Field:		**High-technology commercial systems**	
Source		**Distribution (%)**	**Comments**
Documentation		30%	
Legacy system requirements		30%	
Stakeholders		40%	
General note:	The environment is less regulated and formal than some other companies or organisations.		

6.3.4 Subject Matter Expert 3

Table 40 summarises the professional experience of SME-3. The raw data results of the interview conducted with SME-3 are contained in Appendix C.3. The highlighted areas indicate the fields of expertise for SME-3.

Table 40: Summary of data elicited for SME-3

Name			**Interview date**
SME-3			11 June 2017
Field of experience			**Years in field**
High-technology commercial systems			
High-technology military systems			6 years
Financial systems (banking systems)			17 years
Commercial systems			4 years
Medical system			
Transport systems			4 years
Estimation of the number of requirements and typical project duration			
Field:	**Financial – Banking**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects		±3 months	±1 month
Medium projects		3–12 months	3–4 months

Large projects		> 10 months	4–6 months
Field:	**High-technology military systems**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects			
Medium projects			
Large project		> 12 months	±6 months
The split of requirements between the different sources			
Field:		**Financial – Banking**	
Source		**Distribution (%)**	**Comments**
Documentation		20%	
Legacy system requirements		60%	
Stakeholders		20%	
Field:		**High-technology military systems**	
Source		**Distribution (%)**	**Comments**
Documentation		30%	
Legacy system requirements		40%	
Stakeholders		60%	
Note: The spilt in requirements will depend on if it is a commercial off-the-shelf (COTS) product upgrade or a bespoke product development. A COTS product upgrade will have a higher percentage of legacy systems requirements.			

6.3.5 Subject Matter Expert 4

Table 41 summarises the professional experience of SME-4. The raw data results of the interview conducted with SME-4 are contained in Appendix C.4. The highlighted area indicates the field of expertise for SME-4.

Table 41: Summary of data elicited for SME-4

Name SME-4	**Interview Date** 16 March 2018
Field of experience	**Years in field**
High-technology commercial systems	
High-technology military systems	
Financial systems	

Commercial systems			
Medical system			
Space systems (satellite payloads – science instruments)		28 years	
Estimation of the number of requirements and typical project duration			
Field:	**Satellite payloads – Science Instruments – NASA**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects			
Medium projects	< 500	10 years development + 18 years operational	6 years
Large projects			
Estimation of the number of requirements and typical project duration			
Field:	**Satellite payloads – Science Instruments – ESA**		
Project size	**Requirements**	**Total project duration**	**Project definition and design duration**
Small projects			
Medium projects	< 500	12 years development + 10 years operational	7 years
Large projects			
Split of requirements between different sources			
Field:			
Source		**Distribution (%)**	**Comments**
Documentation			
Legacy system requirements			
Stakeholders			

6.4 Analysis and evaluation of selected results from semi-structured interviews

As stated in paragraph 6.1, the purpose of this chapter is to validate the design and implementation of the research tool, which was done by taking selected cases from the semi-structured interviews and processing them using the research tool. The cases were selected to demonstrate the validity of the design and implementation of the research tool over a spectrum of cases from different industries.

6.4.1 Validation Experiment 1

The first single-case mechanism experiment used the results obtained from SME-3 in evaluating the behaviour of the parameter *project type*. The digitised data obtained from the interview is available in Appendix C.3.1.

6.4.1.1 Descriptive inference process

The original raw data obtained during the interview consisted of freehand graphs captured in the hard copy of the interview questionnaire. These drawing are shown in Figure 76, Figure 78 and Figure 80. As can be seen from the drawings, certain corrections were made by SME-3 before arriving at the result. These drawings were digitised to provide numerical data that was used by the simulation artefact during the abductive inference process. The digitised results are shown in Figure 77, Figure 79 and Figure 81.

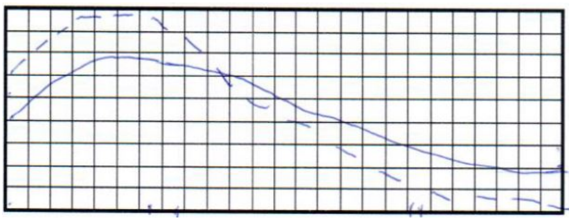


Figure 76: Raw results for SME-3's response for project type elicitation behaviour

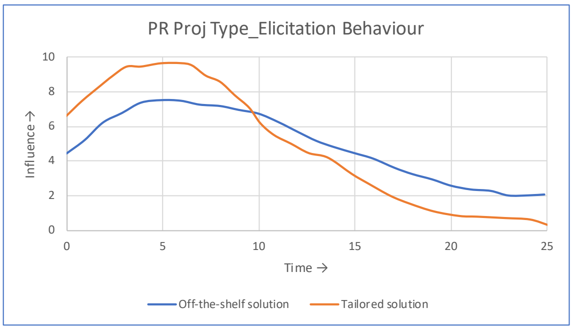


Figure 77: Digitised results for SME-3's response for project type elicitation behaviour

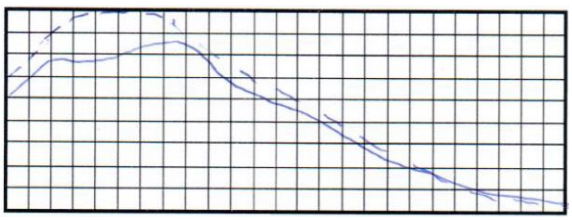


Figure 78: Raw results for SME-3's response for project type error creation behaviour

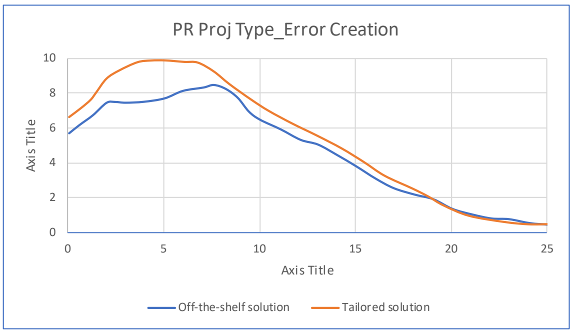


Figure 79: Digitised results for SME-3's response for project type error creation behaviour

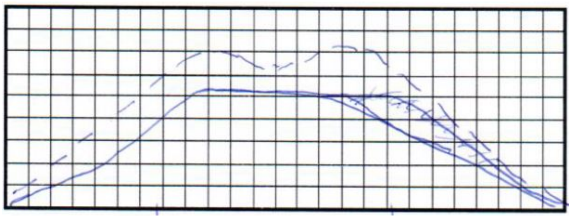


Figure 80: Raw results for SME-3's response for project type error detection behaviour

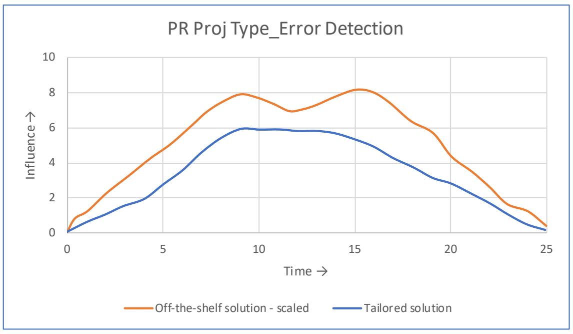


Figure 81: Digitised results for SME-3's response for project type error detection behaviour

The original data was digitised with a full-scale value for the *influence* of 10 and a full-scale value for the *time* of 25. The data was rescaled to represent a full-scale value for *influence* of 1 and a full-scale value for the *time* of 20 days to correspond to the typical parameters provided by SME-3. The parameters included that the project definition and design phase for a small project lasts approximately one month (20 working days), hence the rescale to 20 days. The data was also inspected to ensure that data exists for time=0 as well as time = 20 to prevent simulation errors. No additional comments were provided by the SME for this specific case.

The scaled results are shown in Figure 82 to Figure 84.

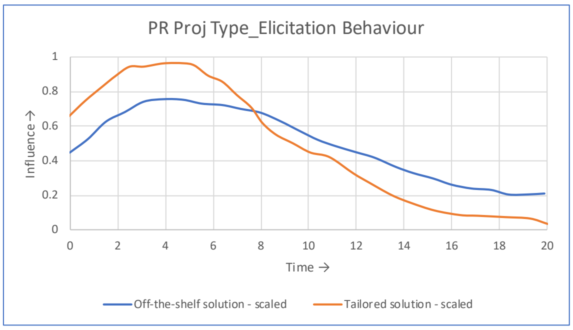


Figure 82: Scaled digitised results for SME-3's response for project type elicitation behaviour

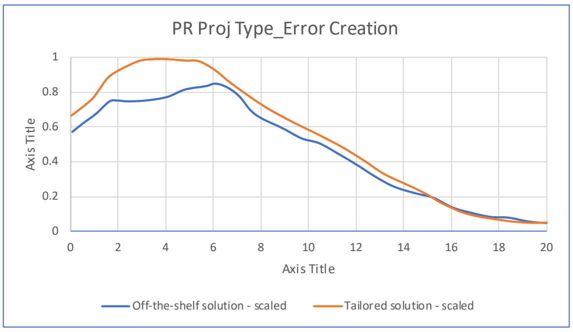


Figure 83: Scaled digitised results for SME-3's response for project type error creation behaviour

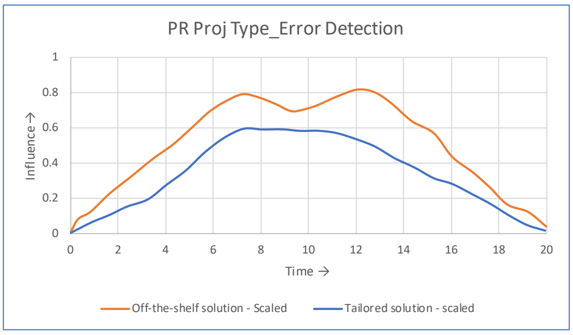


Figure 84: Scaled digitised results for SME-3's response for project type error detection behaviour

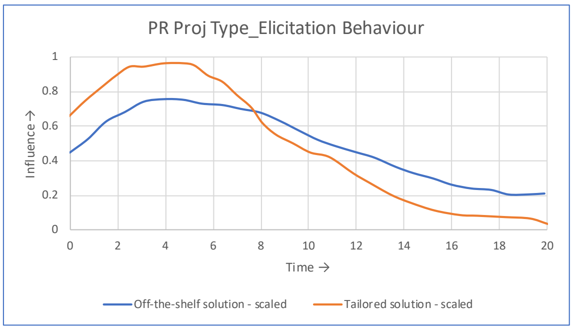
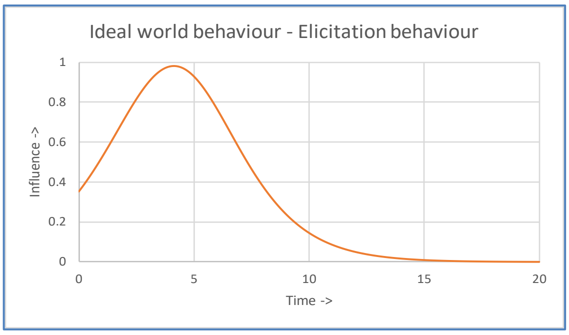
6.4.1.2 Abductive inference process

Paragraph 4.4.4.2 presented the purpose of the abductive inference process. The two main components of the abductive inference process are a causal explanation that seeks to explain the changes in a specific variable as the result of a previous change in a different variable. The second type of explanation commonly used in abductive inference is a mechanistic explanation of phenomena in terms of the interaction of the components of the object-of-study.

6.4.1.2.1 Comparing real-world behaviour and ideal-world behaviour

The real-world behaviour and the ideal-world behaviour for Analysis Sample 1 is shown in Table 42. The real-world behaviour of Analysis Sample 1 was compared with the ideal-world behaviour based on several evaluation points. The results of this comparison are shown in Table 43.

Table 42: Real-world behaviour and ideal-world behaviour for Validation Experiment 1

Real-world behaviour	Ideal-world behaviour
	

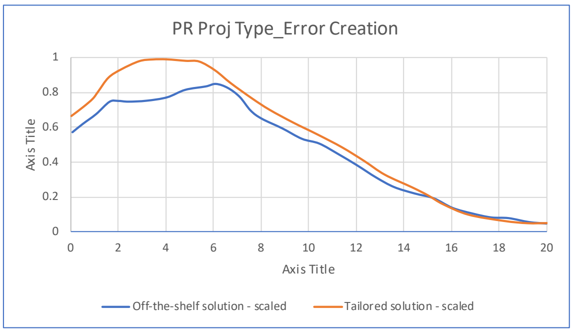
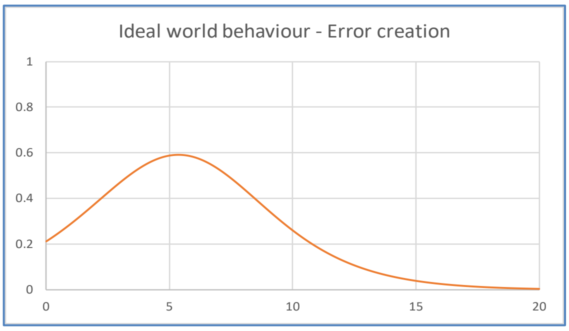
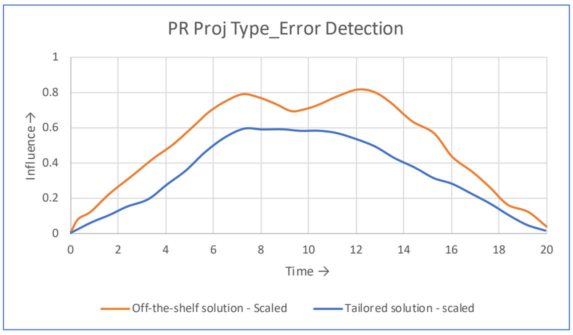
<table>
<tr><th>Real-world behaviour</th><th>Ideal-world behaviour</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Table 43: Comparison of real-world behaviour and ideal-world behaviour for Validation Experiment 1

No.	Comparison guideline	Comment
1	The relative location of the local peak in the elicitation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak can be observed in the 1st half of the timescale for the results of both the off-the-shelf and the tailored solution.*
2	The relative location of the local peak in the error creation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak can be observed in the 1st half of the timescale for the results of both the off-the-shelf and the tailored solution.*
3	The relative location of the local peak in the error detection behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak can be observed in the centre of the timescale for the results of both the off-the-shelf and the tailored solution.*
4	The relative height of the local peak in the elicitation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the tailored solution is at approximately 100% of the full-scale value, and the relative height of the off-the-shelf solution is at 80% of the full-scale value.*

No.	Comparison guideline	Comment
5	The relative height of the local peak in the error creation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the tailored solution is at approximately 100% of the full-scale value, and the relative height of the off-the-shelf solution is at 80% of the full-scale value.*
6	The relative height of the local peak in the error detection behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the tailored solution is at approximately 80% of the full-scale value, and the relative height of the off-the-shelf solution is at 60% of the full-scale value.*
7	Comparison of the relative height of the local peak of the elicitation behaviour vs the relative height of the local peak of the error creation behaviour.	*The relative height of the local peaks of both the elicitation behaviour and the error creation is similar.*
8	Any noticeable delay present in the observed response of the error detection behaviour.	*The local peak for the elicitation behaviour occurred at approximately 20% of the relative timescale. The local peak for the error detection behaviour occurred at approximately 50% of the relative time.*

6.4.1.2.2 Causal and mechanistic explanations

The following observations can be made resulting from the comparison:

1. Local peaks can be observed in the behaviour of all three parameters. This indicates that the activity reaches a peak after a specific time and then declines in an orderly manner, showing that all processes are making a natural progression to reach a conclusion.
2. The occurrence of the local peaks for both the elicitation behaviour and the error creation behaviour early in the overall process is an overall good indicator of a requirements engineering process that is under control.
3. A point of concern is that the elicitation behaviour and error creation behaviour show a similar trend in terms of the value of the local peak. This can be interpreted that a significant number of errors is being created while the elicitation takes place, depending on the overall error creation rate present in this situation.

4. The peak of the error detection behaviour occurs later in time than the elicitation behaviour and error creation behaviour. This is in line with the hypothesis that the detection of the errors will take place later in time when the requirements analysis process starts to detect inconsistent requirements.
5. When examining the effect of the observed behaviour of the project size, type and complexity on other elements of the overall causal structure, as shown in Figure 34, it can be seen that there are no elements that have an impact on the project size, type and complexity element.
6. It can, however, be observed that the *project size, type and complexity* parameter have a direct impact on the number of requirements, which in turn again links to requirements volatility. This relationship is shown in Figure 85.
7. It can further be seen that the *project size, type and complexity* have an impact on the number of stakeholders who are present in a project. In turn, this has a direct impact on the number of requirements element and the stakeholder misinformation element. It is expected that as the size, type or the complexity of a project increases, so will the number of stakeholders of the project, which in turn will increase the number of requirements and the stakeholder misinformation present in the project.

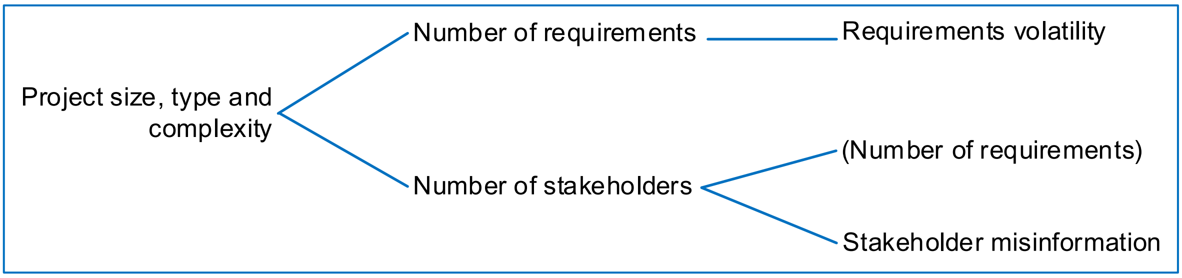


Figure 85: Impact of the project size, type and complexity element on other elements

6.4.1.2.3 Effect analysis

The following scenarios were evaluated using the simulation artefact as shown in Table 44 to Table 46. The results of the simulation artefact are shown in Figure 86 to Figure 88.

Table 44: Validation Experiment 1, Scenario 1 parameter set

Parameters for SME-3, Scenario 1	Value
Number of requirements for a small project	Not specified – work on an assumption of 100 requirements
Project duration	20 days
Elicitation rate	Calculated at 5 requirements/day (100 requirements/20 days)

Parameters for SME-3, Scenario 1	Value
Error creation rate	Assume 10% for Scenario 1
Error detection rate	Assume 10% of the elicitation rate for Scenario 1
Colour of line on results graph	Green

Table 45: Validation Experiment 1, Scenario 2 parameter set

Parameters for SME-3, Scenario 2	Value
Number of requirements for a small project	Not specified – work on an assumption of 100 requirements
Project duration	20 days
Elicitation rate	Calculated at 5 requirements/day (100 requirements/20 days)
Error creation rate	Assume 25% for Scenario 2
Error detection rate	Assume 10% of the elicitation rate for Scenario 2
Colour of line on results graph	Red

Table 46: Validation Experiment 1, Scenario 3 parameter set

Parameters for SME-3, Scenario 2	Value
Number of requirements for a small project	Not specified – work on an assumption of 100 requirements
Project duration	20 days
Elicitation rate	Calculated at 5 requirements/day (100 requirements/20 days)
Error creation rate	Assume 10% for Scenario 3
Error detection rate	Assume 25% of the elicitation rate for Scenario 3
Colour of line on results graph	Blue

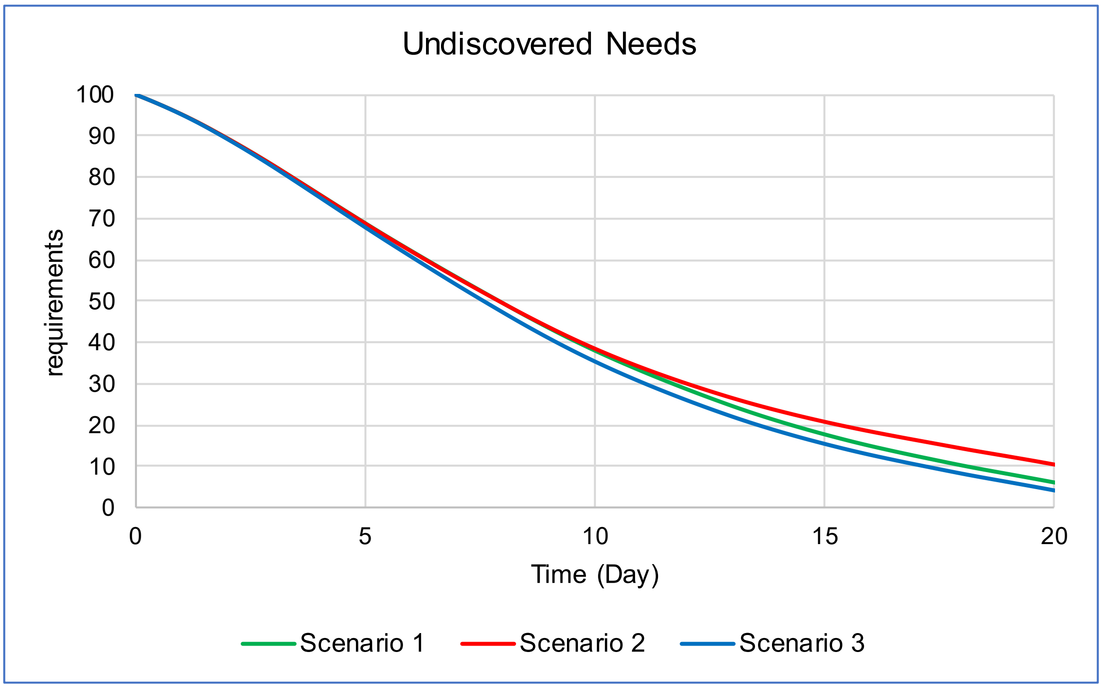


Figure 86: SME-3 undiscovered needs behaviour for Scenario 1 to Scenario 3

The effect of the different elicitation rates, error creation rates and error detection rates from the different scenarios on the undiscovered needs in the example is minuscule.

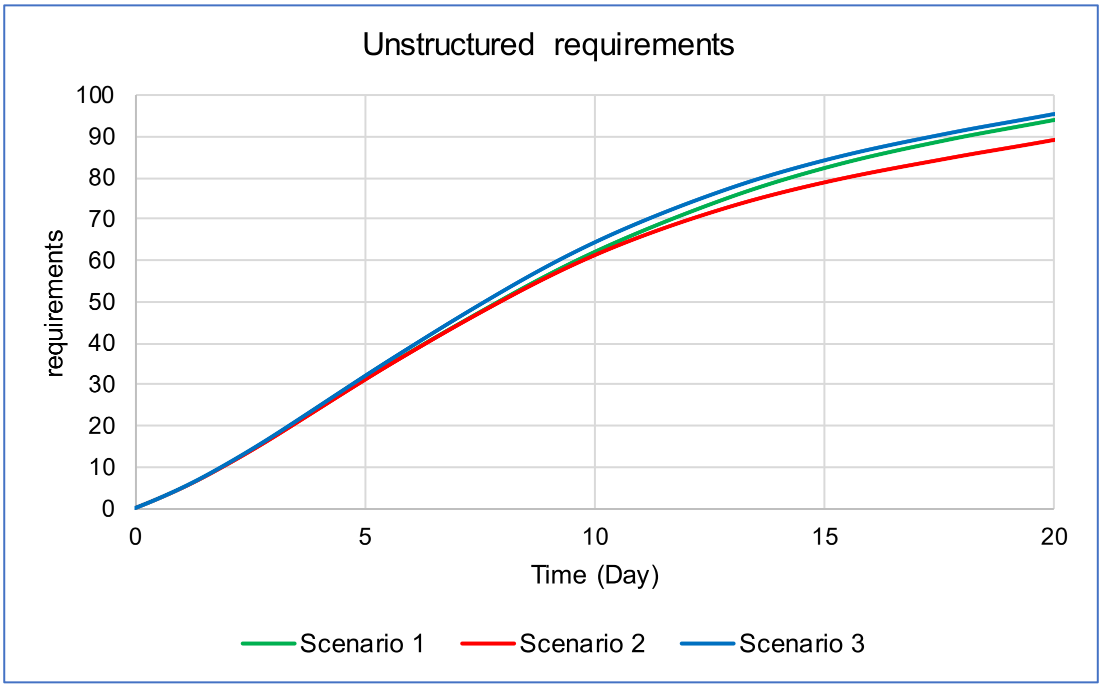


Figure 87: SME-3 unstructured requirements behaviour for Scenario 1 to Scenario 3

The effect of the different elicitation rates, error creation rates and error detection rates from the different scenarios on the unstructured requirements in the example is similarly insignificant.

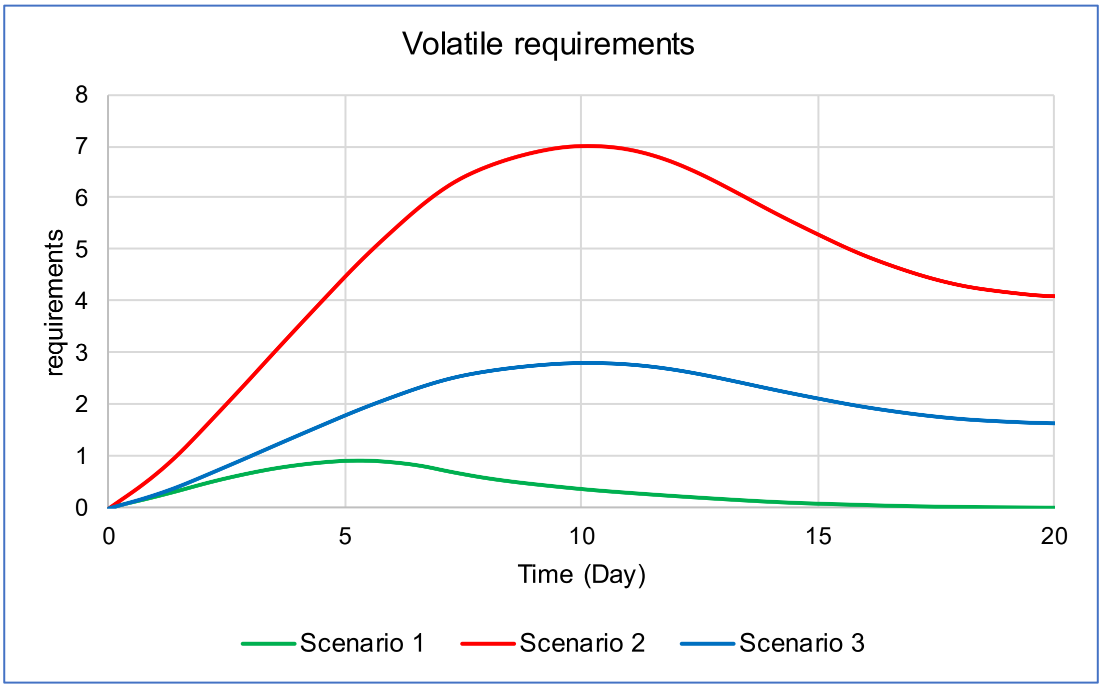


Figure 88: SME-3 volatile requirements behaviour for Scenario 1 to Scenario 3

The most significant effect of the different parameters from the different scenarios is visible in the behaviour of the volatile requirements parameter. Here, the effect of the different scenarios can be clearly seen. When the error creation rate is increased, but the error detection rate is kept stable, the volatile requirements end at a value of approximately four requirements that have not been detected at the end of the project definition and design phase. These requirements could potentially again surface towards the end of the project, in which case they may have a negative effect on the outcome of the project.

Another noticeable effect is that one would expect that with Scenario 3, in which the error detection rate is higher than for the other scenarios, the volatile requirements would be depleted earlier. However, what is observed is that the number of volatile requirements detected can never be more than the number of volatile requirements that exist at that point in time.

6.4.1.3 Other observations and comments

The sample selected for Validation Experiment 1 was one in which the apparent behaviour seemed to be that of a requirements engineering process which is nominally under control. The results presented supported this assumption. *Project* *size, type and complexity* is one of the core input parameters of a project and is as such not affected by any of the internal parameters in the project. It was however demonstrated that this parameter has a direct causal influence on several other parameters in the project. The main impact of the *project size, type and complexity* parameter is on the number of requirements that are present in the project. It is also self-explanatory that there is a direct relationship between the number of parameters and the inherent complexity in the project. The second impact of the *project size, type and complexity* parameter is on the number of stakeholder that are part of the project. The general view is that the more stakeholders are present, the higher factors such as conflict among the stakeholders will be. The effect of this will be to slow the learning loop when it comes to the detection of errors in the requirements set. This slow learning loop combined with a large requirement set will result in errors being detected late in the project and thus increasing the requirements volatility.

6.4.2 Validation Experiment 2

The second single-case mechanism experiment that will be analysed is the effect of a requirements engineer with low experience compared with a requirements engineer with high experience in the opinion of SME-4. The digitised data obtained from the interview is available in Appendix C.4.16.

6.4.2.1 Descriptive inference process

The original raw data obtained during the interview consisted of freehand graphs captured in the hard copy of the interview results. These drawings are shown in Figure 89, Figure 91 and Figure 93. The hand-drawn graphs were digitised to provide numerical data that was used by the simulation artefact during the abductive inference process. The digitised results are shown in Figure 90 to Figure 94.

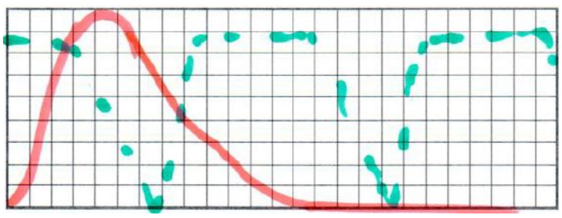


Figure 89: Raw results for SME-4's response for requirements engineer experience elicitation behaviour

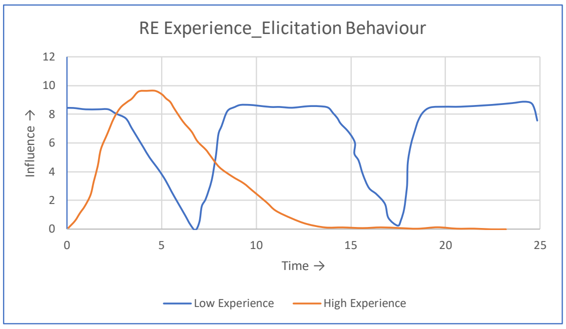


Figure 90: Digitised results for SME-4's response for requirements engineer experience elicitation behaviour

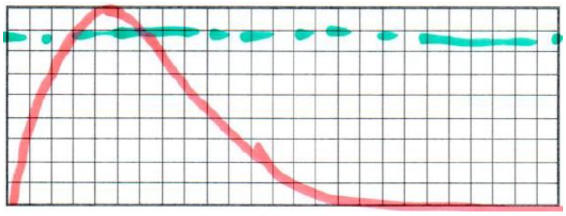


Figure 91: Raw results for SME-4's response for requirements engineer experience error creation behaviour

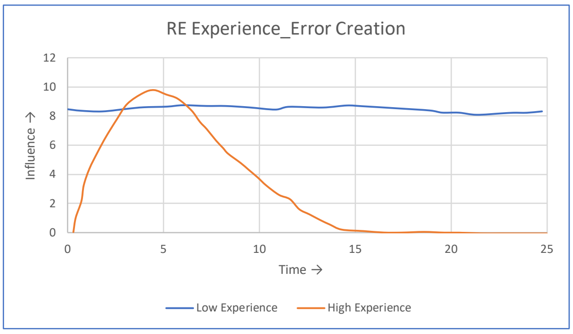


Figure 92: Digitised results for SME-4's response for requirements engineer experience error creation behaviour

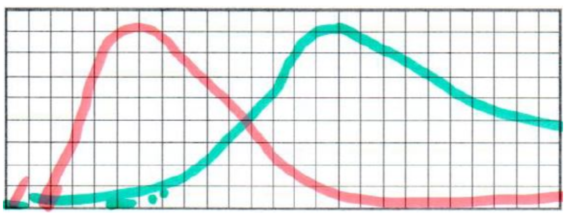


Figure 93: Raw results for SME-4's response for requirements engineer experience error detection behaviour

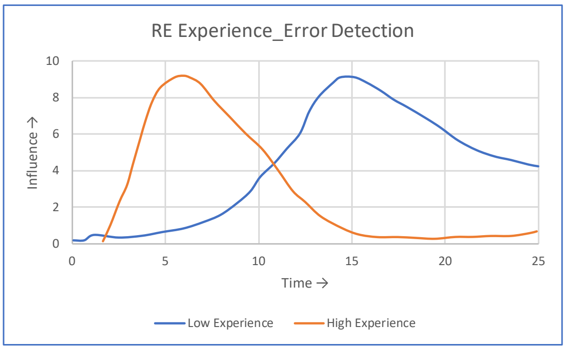


Figure 94: Digitised results for SME-4's response for requirements engineer experience error detection behaviour

The original data was digitised with a full-scale value for the *influence* of 10 and a full-scale value for the *time* of 25. The data was rescaled to represent a full-scale value for *influence* of 1 and a full-scale value for the time of 72 months to correspond to the typical parameters provided by SME-4. The parameters included that the project definition and design phase for a small project lasts approximately 6 years; hence the rescale to 72 months. The data was also inspected to ensure that data exists for time=0 as well as time = 72 to prevent simulation errors. No additional comments were provided by the SME for this specific case. The scaled results are shown in Figure 95 to Figure 97.

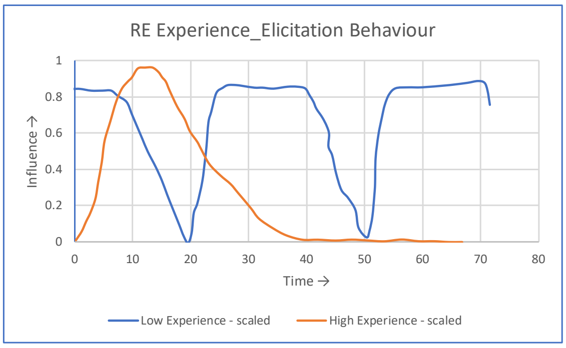


Figure 95: Scaled digitised results for SME-4's response for requirements engineer experience elicitation behaviour

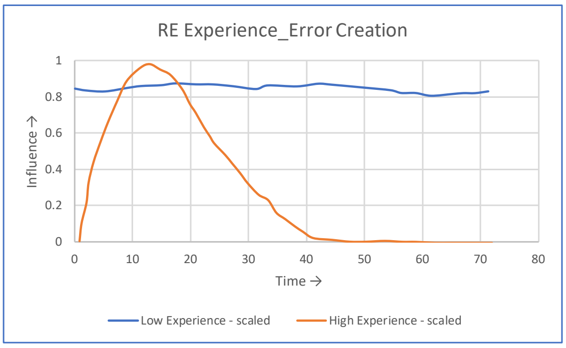


Figure 96: Scaled digitised results for SME-4's response for requirements engineer experience error creation

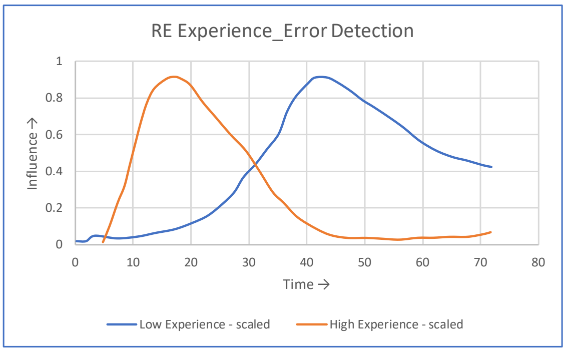


Figure 97: Scaled digitised results for SME-4's response for requirements engineer experience error detection

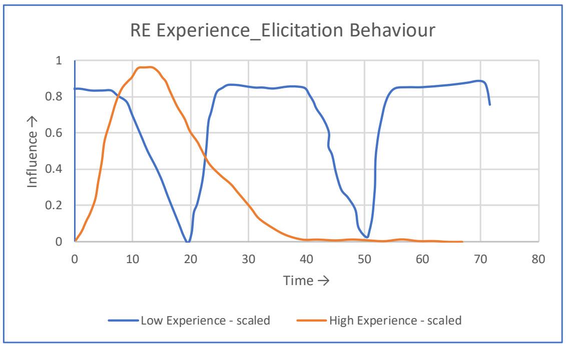
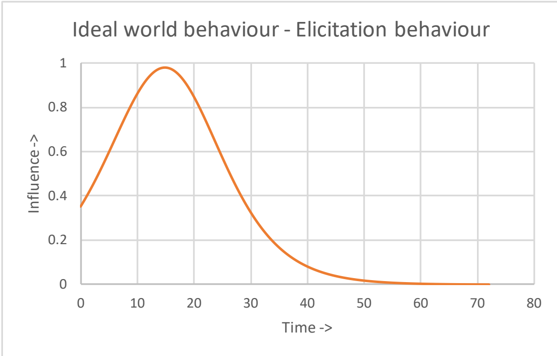
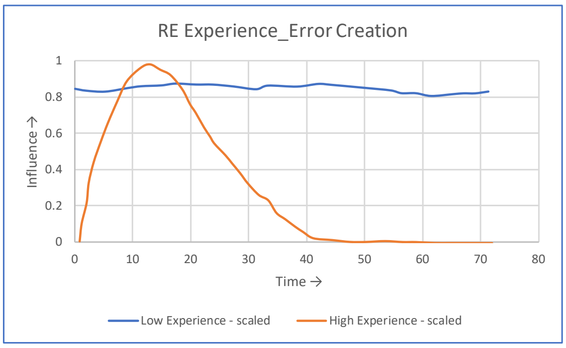
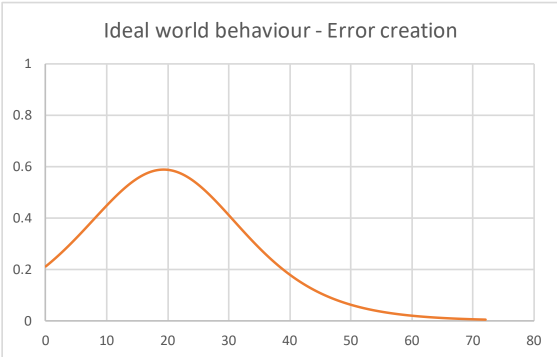
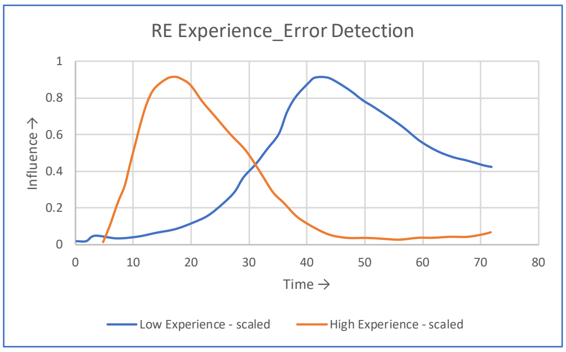
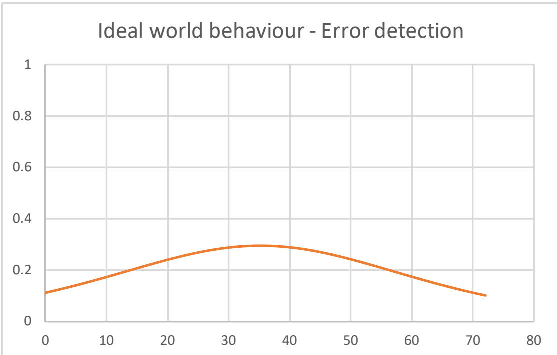
6.4.2.2 *Abductive inference process*

The abductive inference process will examine the data with the view of providing causal and mechanistic explanations for the observed behaviour.

6.4.2.2.1 Comparing real-world behaviour and ideal-world behaviour

The first step in identifying and describing the observed behaviour is to compare the real-world behaviour with an ideal-world behaviour. The real-world behaviour and the ideal-world behaviour for Analysis Sample 2 are shown in Table 47.

Table 47: Real-world behaviour and ideal-world behaviour for Validation Experiment 2

<table>
<tr><th>Real-world behaviour</th><th>Ideal-world behaviour</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

One of the first impression when examining the data is the complete disjointness between the behaviour of a requirements engineer with a low level of experience and a requirements engineer with a high level of experience. This behaviour is discussed in more detail in Table 48.

Table 48: Comparison of real-world behaviour with ideal-world behaviour for Validation Experiment 2

No.	Comparison guideline	Comment
1	The relative location of the local peak in the elicitation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak can be observed in the 1st half of the timescale for the high experience result.* *The low experience result, however, shows a very problematic behaviour in which the elicitation remains at a constant level with marked areas where the elicitation behaviour drops to 0.*
2	The relative location of the local peak in the error creation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak can be observed in the 1st half of the timescale for the high experience result.* *No peak can be observed for the low experience result. The behaviour indicates a continuous creation of elicitation errors that never decreases.*
3	The relative location of the local peak in the error detection behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local peak for the high experience result can be observed in the 1st third of the timescale.* *The peak for the low experience result is in the 2nd third of the timescale.*
4	The relative height of the local peak in the elicitation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the high experience result is at approximately 100% of the full-scale value.* *The relative height of the low experience result is at 80% of the full-scale value.*
5	The relative height of the local peak in the error creation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the high experience result is at approximately 100% of the full-scale value.* *The relative height of the low experience result is at 80% of the full-scale value.*

No.	Comparison guideline	Comment
6	The relative height of the local peak in the error detection behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local peak for the high experience and the low experience result 80% of the full-scale value.* *The relative height of the off-the-shelf solution is at 60% of the full-scale value.*
7	Comparison of the relative height of the local peak of the elicitation behaviour with the relative height of the local peak of the error creation behaviour.	*The relative height of the local peak for the high experience result and the low experience result are at approximately 100% of the full-scale value.*
8	Any noticeable delay present in the observed response of the error detection behaviour.	*The local peak for the high experience result occurred at approximately 20% of the relative timescale.* *The local peak for the low experience result occurred at approximately 60% of the relative time.*

The following observations can be made from the comparison:

1. The elicitation behaviour for the case in which the requirements engineer has a high level of experience compares remarkably well with the expected ideal-world behaviour. The elicitation reaches a peak early in the process. Thereafter, it decreases gradually to reach a particularly low level after about 50% of the time range.
2. On the other hand, the elicitation behaviour of the low or inexperienced requirements engineer shows a behaviour in which the elicitation process starts at a high level and then suddenly drops off to 0 a quarter of the way into the process. This cycle seems to repeat itself repeatedly. This type of behaviour indicates that the requirements engineering process does not seem to reach a natural conclusion. The dips observed in the elicitation behaviour can probably be explained by review activities taking place at this time, thus pausing the elicitation process.
3. The error creation behaviour observed for the requirements engineer with a high level of experience also seems to be a reasonable behaviour. The one area of concern is, however, with the peak of the behaviour graph. This can be interpreted that the error creation is at an equivalent rate as the elicitation behaviour.

4. The indicated error creation behaviour for the requirements engineer with the low experience reflects the behaviour observed with the elicitation behaviour. The elicitation process is not under control with errors being created at a constant rate throughout the process.
5. The error detection behaviour highlights the difference between the requirements engineer with the high experience and the requirements engineer with the low experience, which is visible in the time delay when the created errors are again detected.
6. The effect of the level of experience of the requirements engineer is shown in Figure 98. The direct impact is on important aspects such as communication between the stakeholders and the requirements engineer, conflict between stakeholders, the ability to manage cultural differences among different stakeholders, and trust among the stakeholders. This also has a direct influence on the volatility of the requirements elicited.

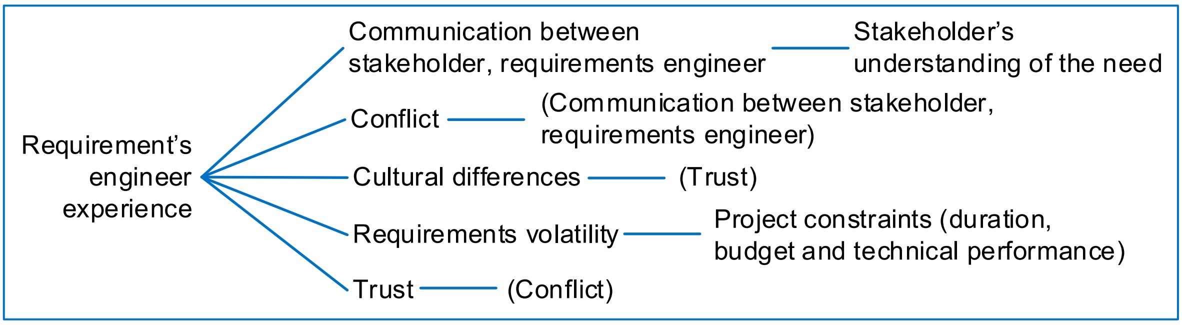


Figure 98: Effect of the requirements engineer's experience on the overall requirements engineering process

6.4.2.2.2 Effect analysis

The following scenarios were evaluated using the simulation artefact as shown in Table 49 to Table 51. The results of the simulation artefact are shown in Figure 99 Figure 103.

Table 49: Single-case Experiment 2, Scenario 1 parameters set

Parameters for SME-4, Scenario 1	Value
Number of requirements for a small project	< 500
Project definition and design duration	72 months
Elicitation rate	Calculated at 7.94 requirements/day (500 requirements/72 months)
Error creation rate	Assume 10% for Scenario 1

Error detection rate	Assume 10% of the elicitation rate for Scenario 1
Colour of line on results graph	Red

Table 50: Single-case Experiment 2, Scenario 2 parameter set

Parameters for SME-4, Scenario 2	Value
Number of requirements for a small project	< 500
Project duration	72 months
Elicitation rate	Calculated at 7.94 requirements/day (500 requirements/72 months)
Error creation rate	Assume 25% for Scenario 2
Error detection rate	Assume 10% of the elicitation rate for Scenario 2
Colour of line on results graph	Blue

Table 51: Single-case Experiment 2, Scenario 3 parameter set

Parameters for SME-4, Scenario 3	Value
Number of requirements for a small project	< 500
Project duration	72 months
Elicitation rate	Calculated at 7.94 requirements/day (500 requirements/72 months)
Error creation rate	Assume 10% for Scenario 3
Error detection rate	Assume 25% of the elicitation rate for Scenario 3
Colour of line on results graph	Green

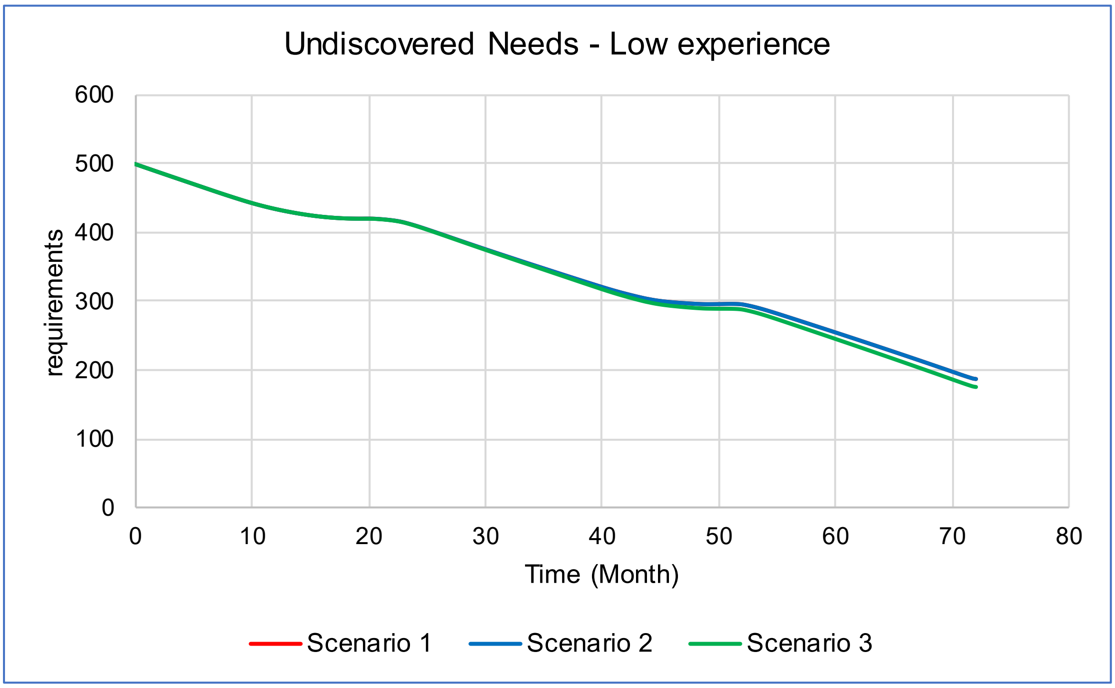


Figure 99: Single-case Experiment 2, SME-4 undiscovered needs behaviour for Scenario 1 to Scenario 3 – Low requirements engineer experience

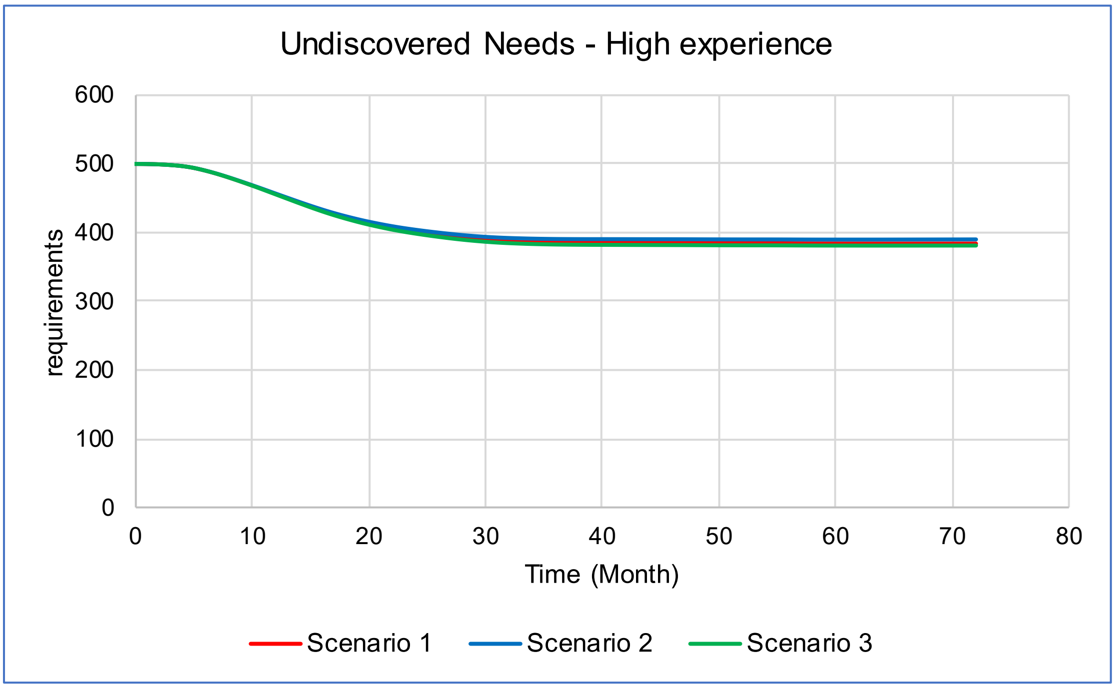


Figure 100: Single-case Experiment 2, SME-4 undiscovered needs behaviour for Scenario 1 to Scenario 3 – High requirements engineer experience

There is no discernible difference among the behaviour of the different scenarios. What is of interest, however, is that more than a quarter of the requirements have not yet been elicited after a period of 72 months.

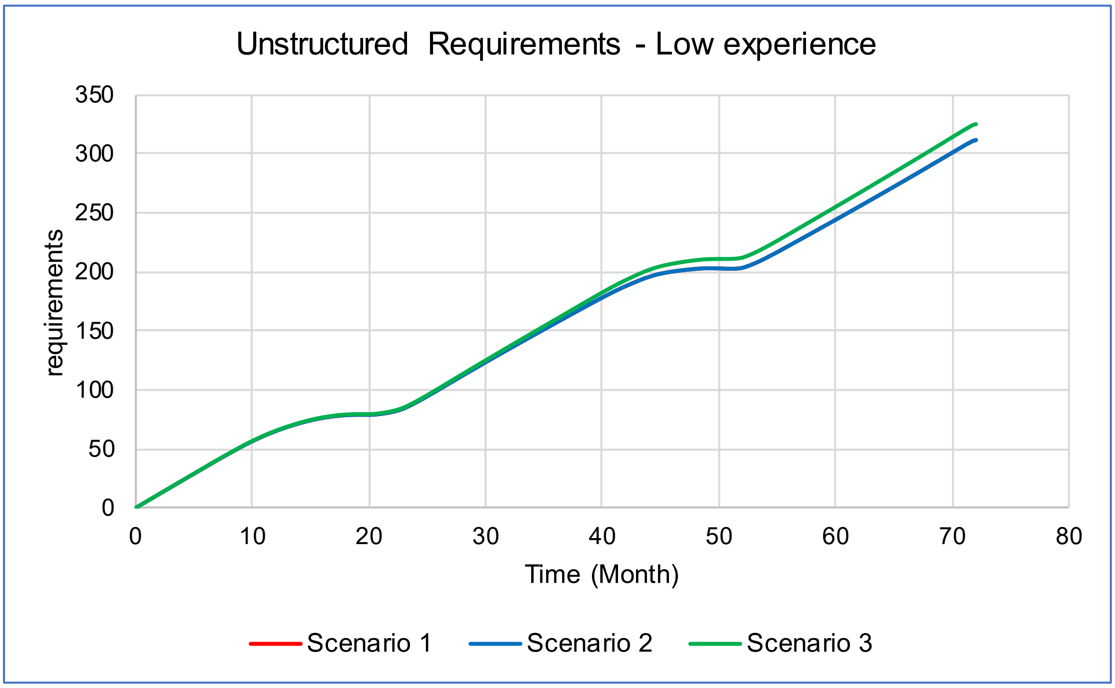


Figure 101: Single-case Experiment 2, SME-4 unstructured requirements behaviour for Scenario 1 to Scenario 3 – Low requirements engineer experience

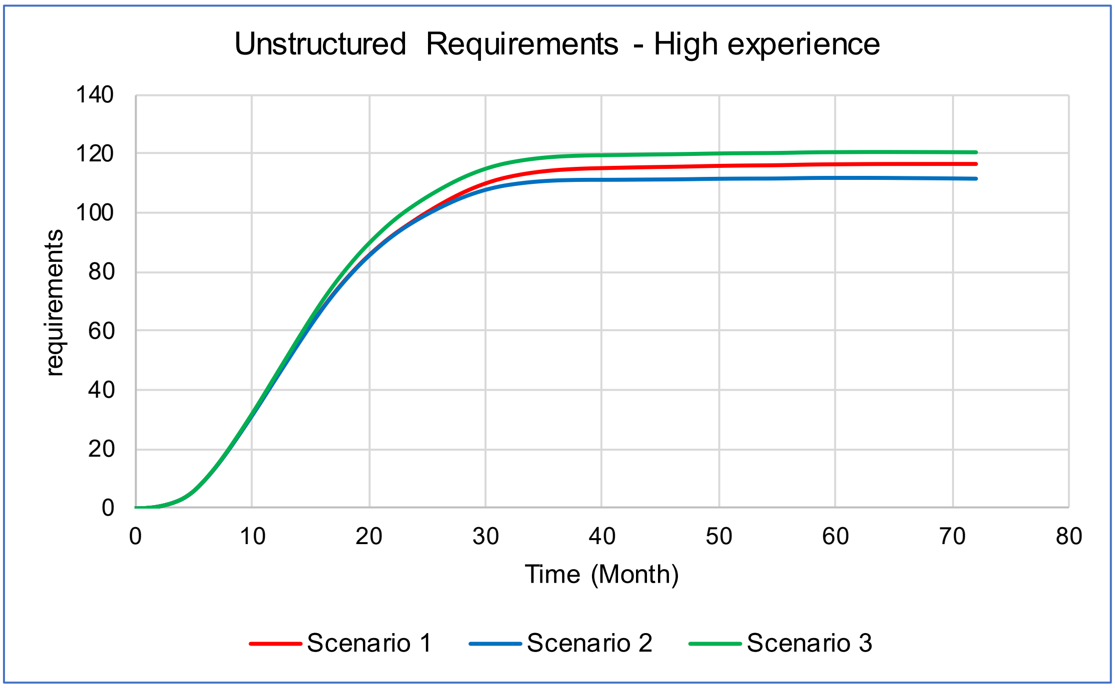


Figure 102: Single-case Experiment 2, SME-4 unstructured requirements behaviour for Scenario 1 to Scenario 3 – High requirements engineer experience

The difference in behaviour between the various cases is again minimal.

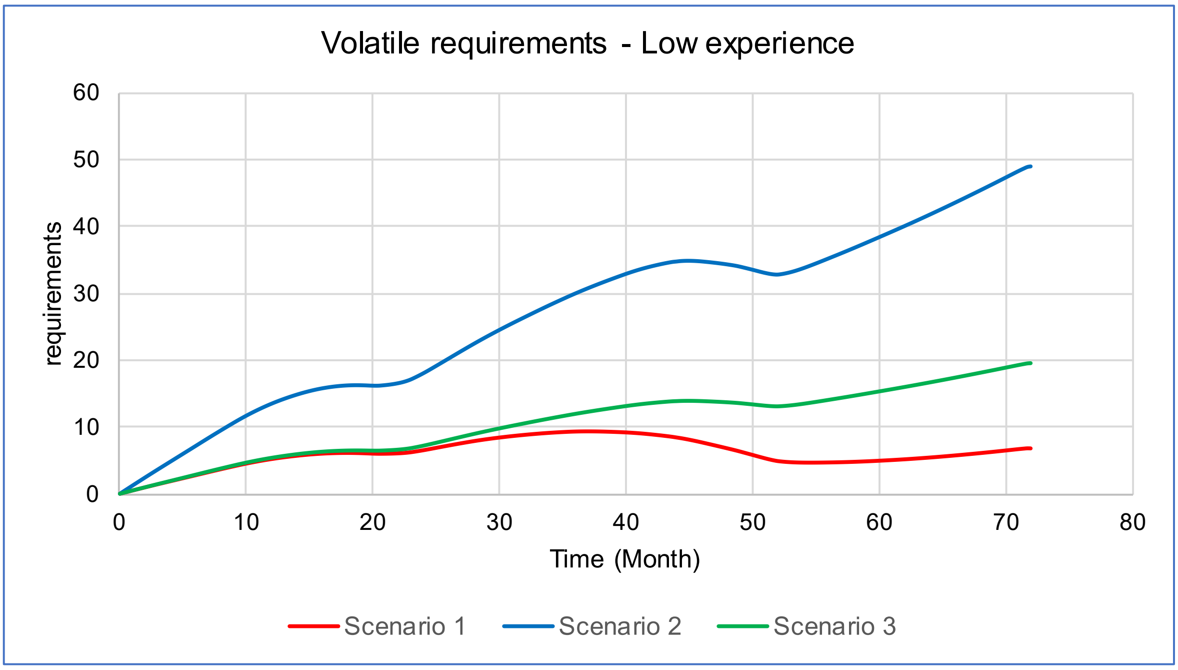


Figure 103: Single-case Experiment 2, SME-4 volatile requirements behaviour for Scenario 1 to Scenario 3 – Low requirements engineer experience

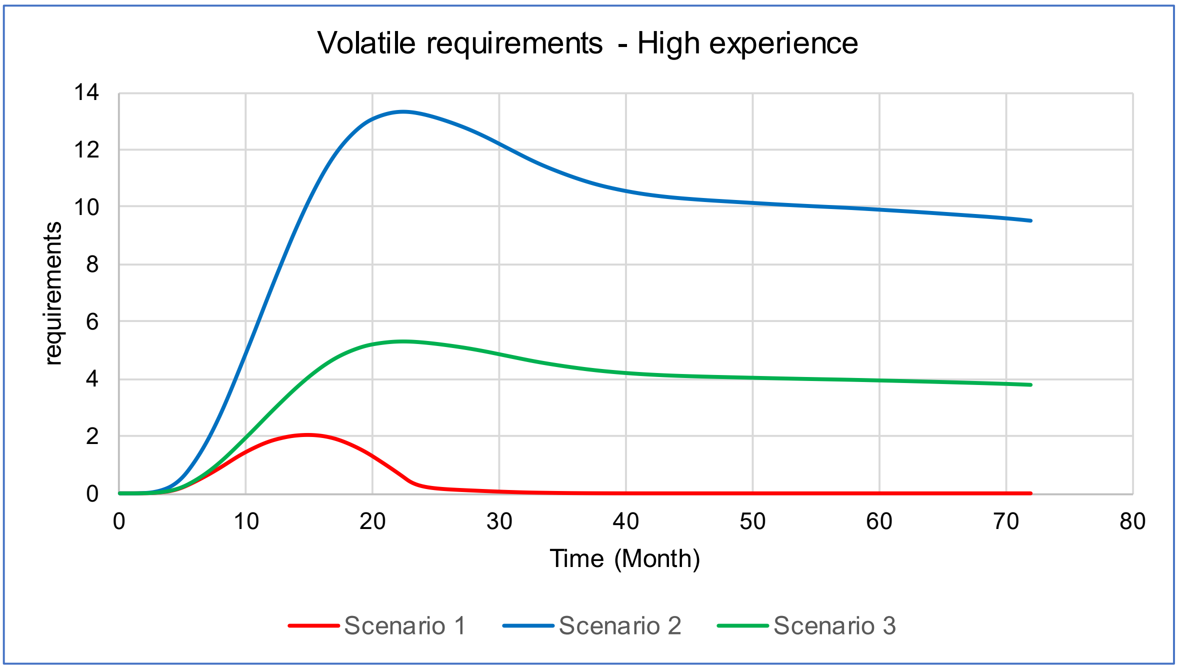


Figure 104: Single-case Experiment 2, SME-4 volatile requirements behaviour for Scenario 1 to Scenario 3 – High requirements engineer experience

The most significant effect elicitation behaviour and the error creation behaviour can be seen in the results of the volatile requirements. For both Scenario 2 and Scenario 3, there is a definite increase in the rate with which volatile requirements are generated towards the end of the process.

6.4.2.3 Other observations and comments

The sample selected for Validation Experiment 2 provided an opportunity to compare the impact of the requirements engineer's experience on the requirements engineering process. This effect can clearly be seen in the behaviour of the volatile requirements that decreases and stabilises in the case of the requirements engineer with a high experience versus the case in which the volatile requirements grow in the case of the requirements engineer with the low experience.

The experience of the requirements engineer becomes apparent in the error detection behaviour. The less experienced the requirements engineer is, the later in the process the error in the requirements set will be detected. The examples presented in Table 47, clearly illustrates (and was confirmed by the SME) that requirements were only discovered or elicited during project review events such as a preliminary design review or a critical design review. Again, this delay in the discovery of the requirements resulted in a higher requirements volatility which in the end results in a failed project for this specific instance.

6.4.3 Validation Experiment 3

The third case that will be examined considers what was reported by SME-1 regarding the effect of the domain knowledge of the stakeholder on the elicitation behaviour, error creation behaviour and the error detection behaviour.

6.4.3.1 Descriptive inference process

The original raw data obtained during the interview consisted of freehand graphs captured in the hard copy of the interview results. These drawings are shown in Figure 105, Figure 107 and Figure 109. The hand-drawn graphs were digitised to provide numerical data that will be used by the simulation artefact during the abductive inference process. The digitised results are shown in Figure 106, Figure 108 and Figure 110.

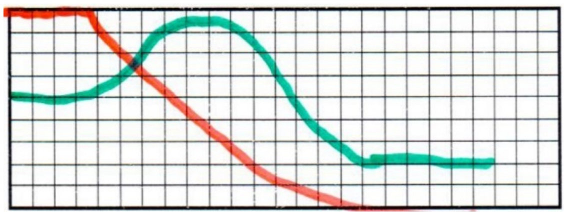


Figure 105: Raw results for SME-1’s response for stakeholder’s domain knowledge elicitation behaviour

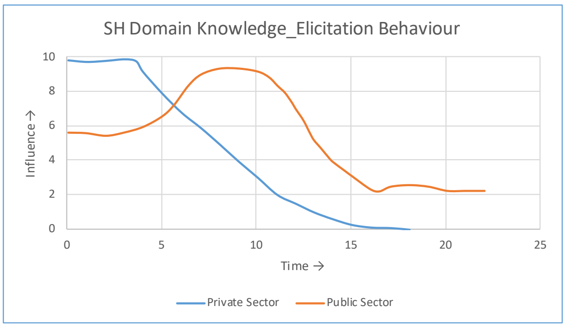


Figure 106: Digitised results for SME-1’s response for stakeholder’s domain knowledge elicitation behaviour

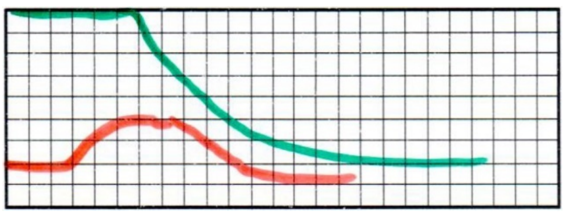


Figure 107: Raw results for SME-1’s response for stakeholder’s domain knowledge error creation behaviour

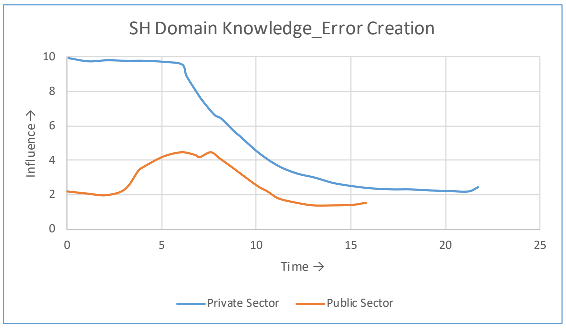


Figure 108: Digitised results for SME-1’s response for stakeholder’s domain knowledge error creation behaviour

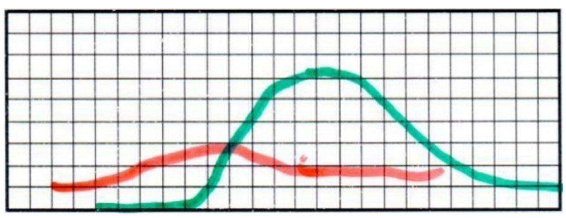


Figure 109: Raw results for SME-1’s response for stakeholder’s domain knowledge error detection behaviour

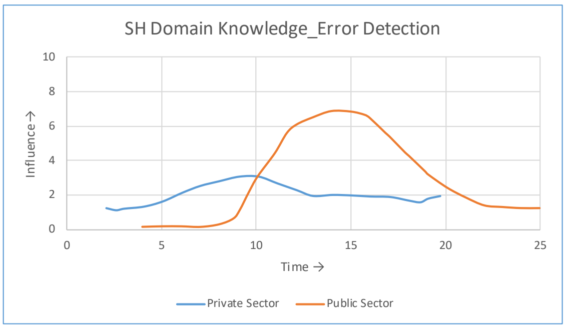


Figure 110: Digitised results for SME-1’s response for stakeholder’s domain knowledge error detection behaviour

The original data was digitised with a full-scale value for the *influence* of 10 and a full-scale value for the *time* of 25. The data was rescaled to represent a full-scale value for *influence* of 1 and a full-scale value for the *time* of 20 days to correspond to the typical parameters provided by SME-1. The parameters included that the project definition and design phase for a small project lasts approximately four months (80 working days); hence the rescale to 112 days. The data was also inspected to ensure that data exists for time=0 as well as time = 112 to prevent simulation errors. No additional comments were provided by the SME for this specific case. The scaled results are shown in Figure 111 to Figure 113 .

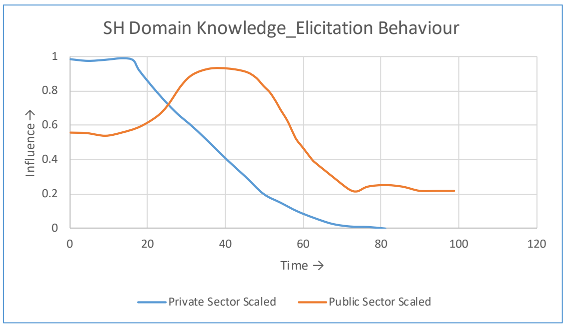


Figure 111: Scaled digitised results for SME-1's response for stakeholder domain knowledge elicitation behaviour

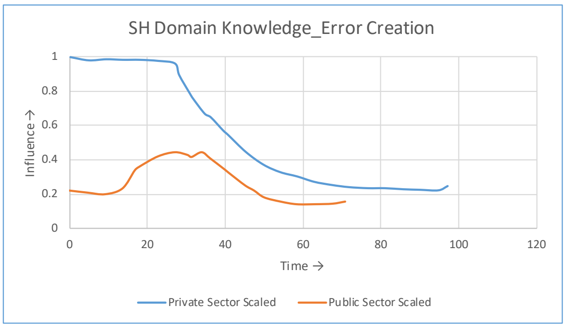


Figure 112: Scaled digitised results for SME-1's response for stakeholder domain knowledge error creation

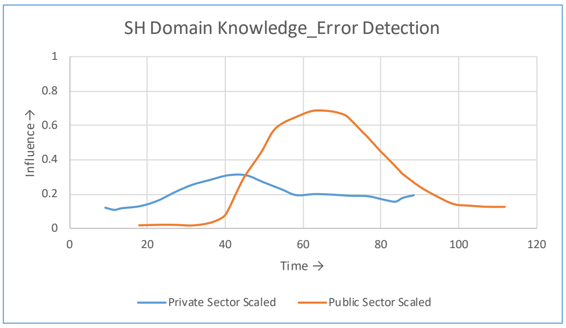


Figure 113: Scaled digitised results for SME-1's response for stakeholder domain knowledge error detection

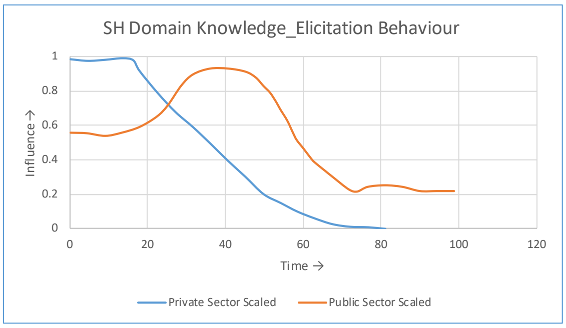
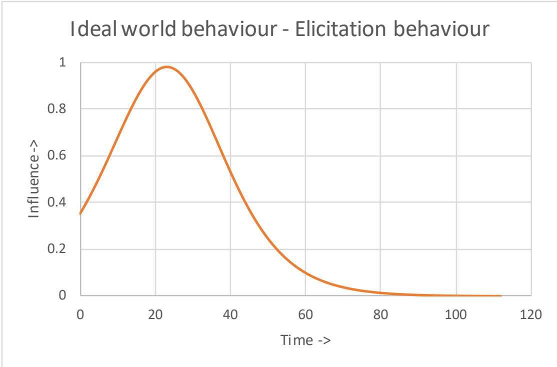
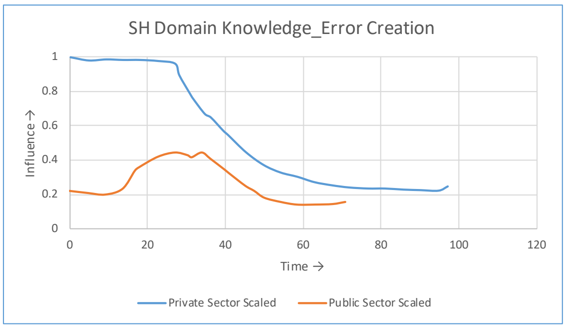
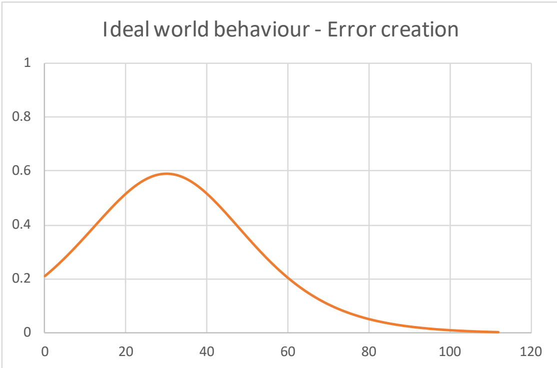
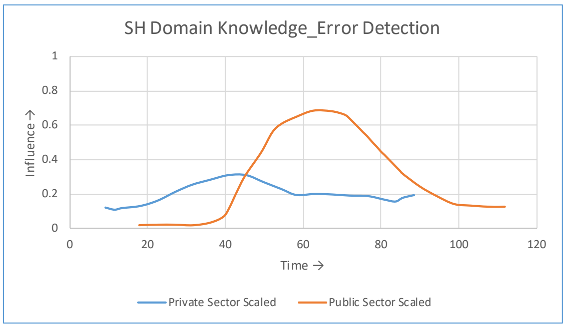
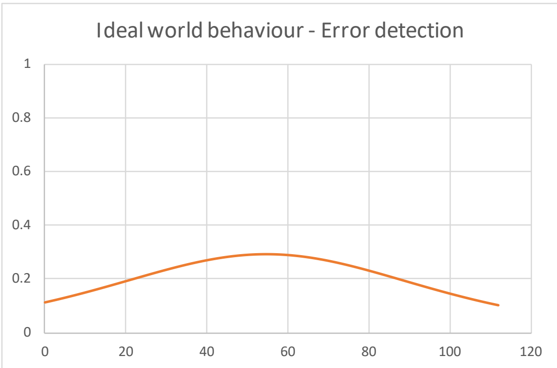
6.4.3.2 Abductive inference process

The abductive inference process will examine the data with the view of providing causal and mechanistic explanations for the observed behaviour.

6.4.3.2.1 Comparing real-world behaviour and ideal-world behaviour

The first step in identifying and describing the observed behaviour is to compare the real-world behaviour with an ideal-world behaviour. The real-world behaviour and the ideal-world behaviour for Analysis Sample 2 are shown in Table 52.

Table 52: Real-world behaviour and ideal-world behaviour for Validation Experiment 3

Real-world behaviour	Ideal-world behaviour
	
	
	

One of the first impression when examining the data is the complete disjointness between the behaviour of a requirements engineer with a low level of experience and a requirements engineer with a high level of experience. This behaviour is discussed in more detail in Table 53.

Table 53: Comparison of real-world behaviour with the ideal-world behaviour for Validation Experiment 3

No.	Comparison guideline	Comment
1	The relative location of the local peak in the elicitation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local plateau can be observed in the 1st half of the timescale for the private sector behaviour result.* *A relative peak for the public sector behaviour can be observed in the 1st half of the timescale.*
2	The relative location of the local peak in the error creation behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local plateau can be observed in the 1st half of the timescale for the private sector behaviour result.* *A relative peak for the public sector behaviour can be observed in the 1st half of the timescale.*
3	The relative location of the local peak in the error detection behaviour (no observable peak, 1st half, 2nd half of timescale).	*A local plateau can be observed in at approximately 50% of the timescale for the private sector result.* *A local peak for the public sector result can be observed at approximately 50% of the timescale.*
4	The relative height of the local peak in the elicitation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local plateau for the private sector result is at approximately 100% of the full-scale value for 20% of the time and gradually declines to 0 thereafter.* *The relative height of the public sector result is at 90% of the full-scale value.*
5	The relative height of the local peak in the error creation behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local plateau for the private sector result is at approximately 100% of the full-scale value for 20% of the time and gradually declines to 0 thereafter.* *The relative height of the public sector result is at 40% of the full-scale value.*

No.	Comparison guideline	Comment
6	The relative height of the local peak in the error detection behaviour (flat response, below 50% level, above 50% level).	*The relative height of the local plateau for the private sector result is at approximately 30% of the full-scale value.* *The relative height of the public sector result is at 70% of the full-scale value.*
7	Comparison of the relative height of the local peak of the elicitation behaviour vs the relative height of the local peak of the error creation behaviour.	*The relative height of the local peak for the public sector result is at approximately 100% for both the elicitation behaviour and the error creation behaviour.*
8	Any noticeable delay present in the observed response of the error detection behaviour.	*The local peak for the public sector result occurred at approximately 8% of the relative timescale, and the local peak for the public sector result occurred at approximately 16% of the relative time.*

The following observations can be made resulting from the comparison:

1. The private sector behaviour starts with a plateau for approximately 25% of the time after which it starts to decline over the remaining time period. The public sector behaviour shows a local peak that already starts at a level of 60% of the full-scale value.
2. The error creation behaviour is like that of the elicitation behaviour, with the private sector level at 100% and the public sector behaviour at 40%.
3. The peak of the error detection behaviour occurs later in time than the elicitation behaviour and error creation behaviour. This is in line with the hypothesis that the detection of the errors will take place later in time.

Figure 114 shows the effect of the observed behaviour of the stakeholder's domain knowledge on other elements of the overall causal structure. The stakeholder's domain knowledge has a direct impact on the requirements engineer misinformation as well as the stakeholder's understanding of the need. Both these elements can again be linked to the requirements volatility element.

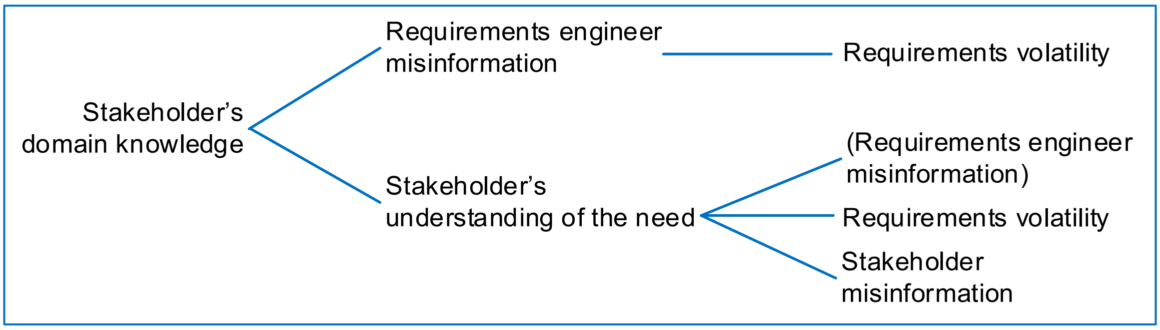


Figure 114: Impact of the stakeholder's domain knowledge on other elements

6.4.3.2.2 Effect analysis

The following scenarios were evaluated using the simulation artefact as shown in Table 44 to Table 46. Figure 86 to Figure 88 show the results of the simulation artefact.

Table 54: Validation Experiment 3, Scenario 1 initial values

Parameters for SME-1, Scenario 1	Value
Number of requirements for a small project	300 requirements
Project duration	Four months, equal to approximately 112 days
Elicitation rate	Calculated at 2.7 requirements/day (300 requirements/112 days)
Error creation rate	Assume 10% for Scenario 1
Error detection rate	Assume 10% of the elicitation rate for Scenario 1
Colour of line on results graph	Red

Table 55: Validation Experiment 3, Scenario 2 initial values

Parameters for SME-1, Scenario 2	Value
Number of requirements for a small project	300 requirements
Project duration	Four months, equal to approximately 112 days
Elicitation rate	Calculated at 2.7 requirements/day (300 requirements/112 days)
Error creation rate	Assume 25% for Scenario 2

Parameters for SME-1, Scenario 2	Value
Error detection rate	Assume 10% of the elicitation rate for Scenario 2
Colour of line on results graph	Blue

Table 56: Validation Experiment 3, Scenario 3 initial values

Parameters for SME-1, Scenario 3	Value
Number of requirements for a small project	300 requirements
Project duration	Four months, equal to approximately 112 days
Elicitation rate	Calculated at 2.7 requirements/day (300 requirements/112 days)
Error creation rate	Assume 10% for Scenario 3
Error detection rate	Assume 25% of the elicitation rate for Scenario 3
Colour of line on results graph	Green

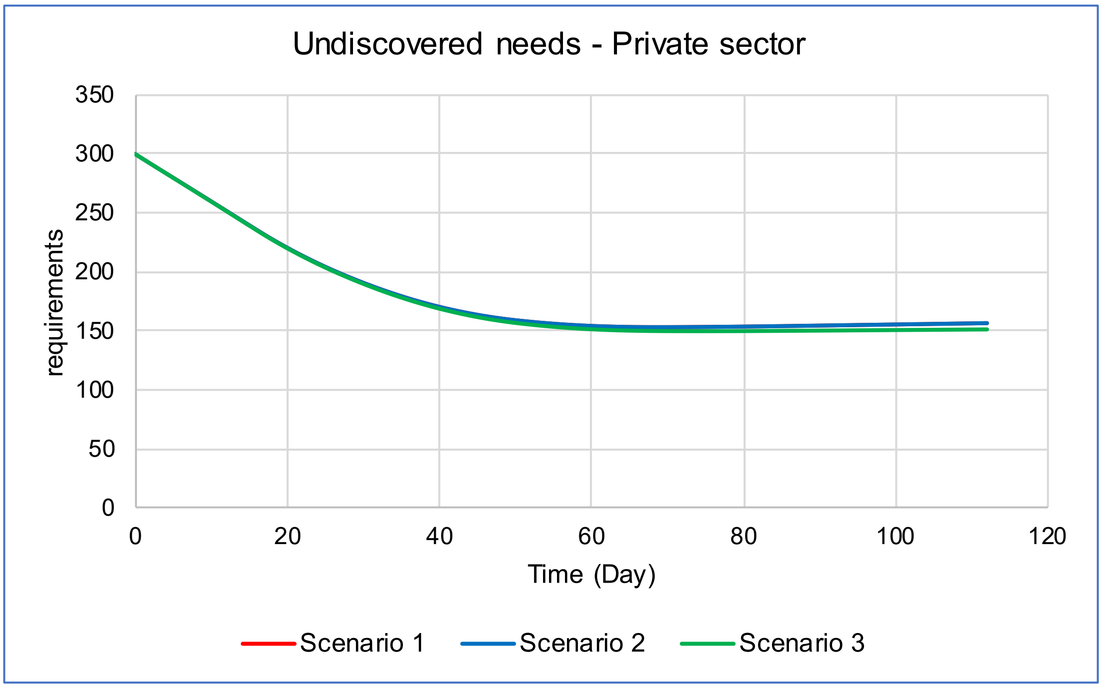


Figure 115: Single-case experiment 3, SME-1 undiscovered needs behaviour for Scenario 1 to Scenario 3 – Private sector

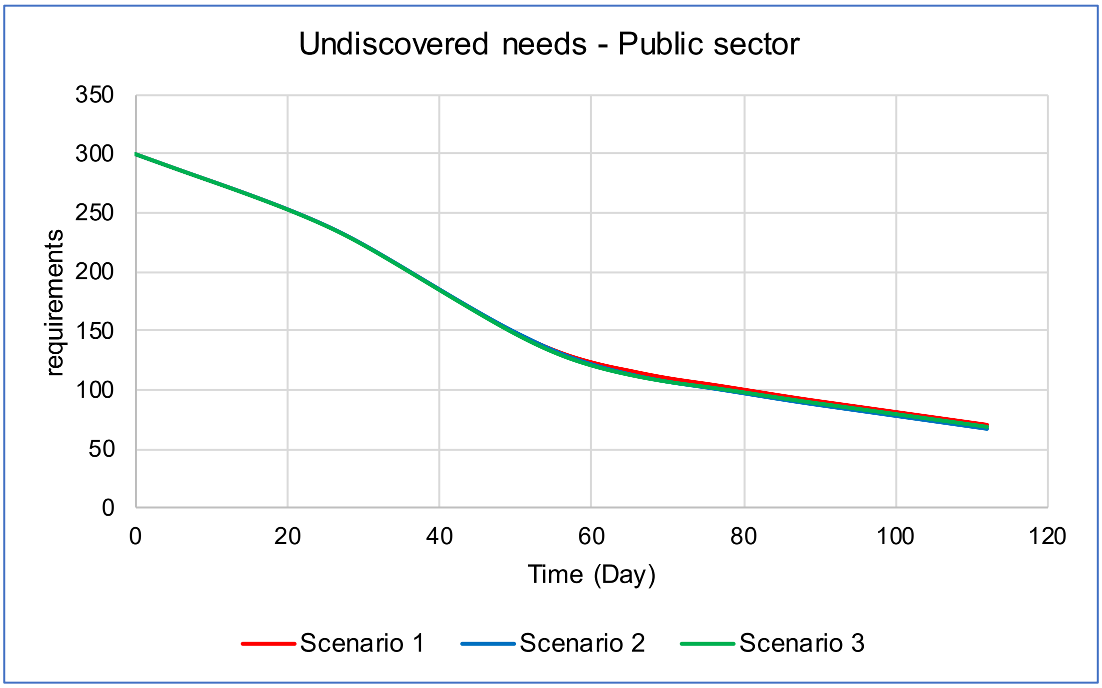


Figure 116: Single-case Experiment 3, SME-1 undiscovered needs behaviour for Scenario 1 to Scenario 3 – Public sector

The behaviour of the undiscovered needs when the stakeholders originate from the private sector is compared with the behaviour when stakeholders originate from the public sector. This comparison shows that behaviour of the undiscovered needs stabilises in the case of the private sector but is still declining at the end of the time period in the case of the public sector. This can be taken as an indication that stakeholders in the public sector exhibit a better domain knowledge.

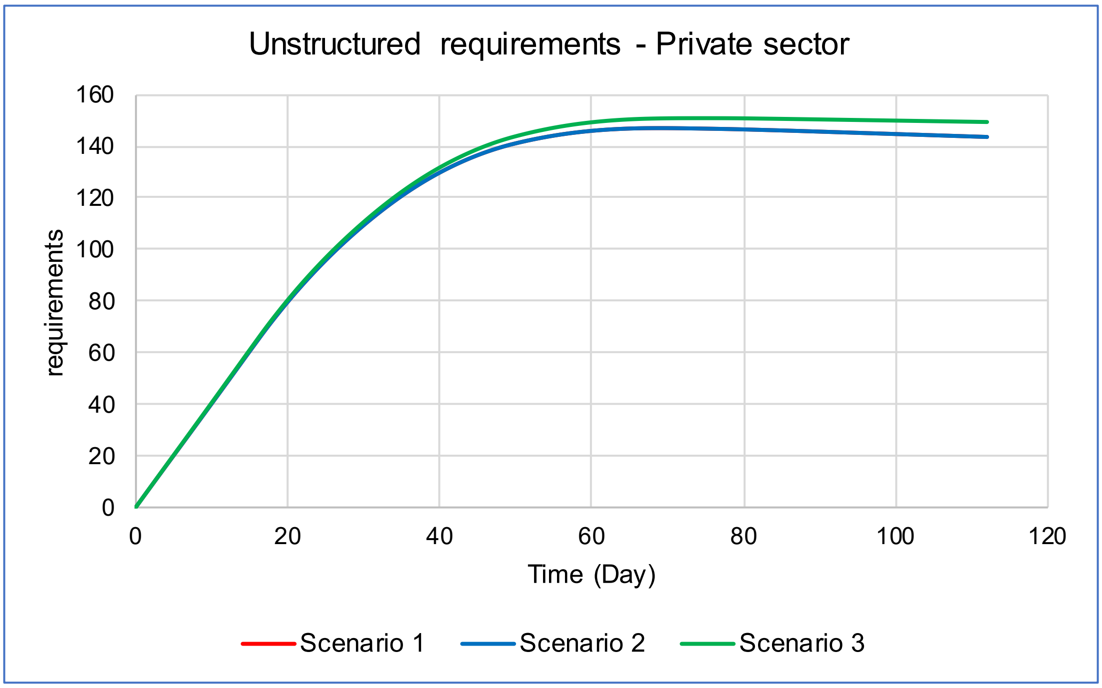


Figure 117: Single-case Experiment 3, SME-1 unstructured requirements behaviour for Scenario 1 to Scenario 3 – Private sector

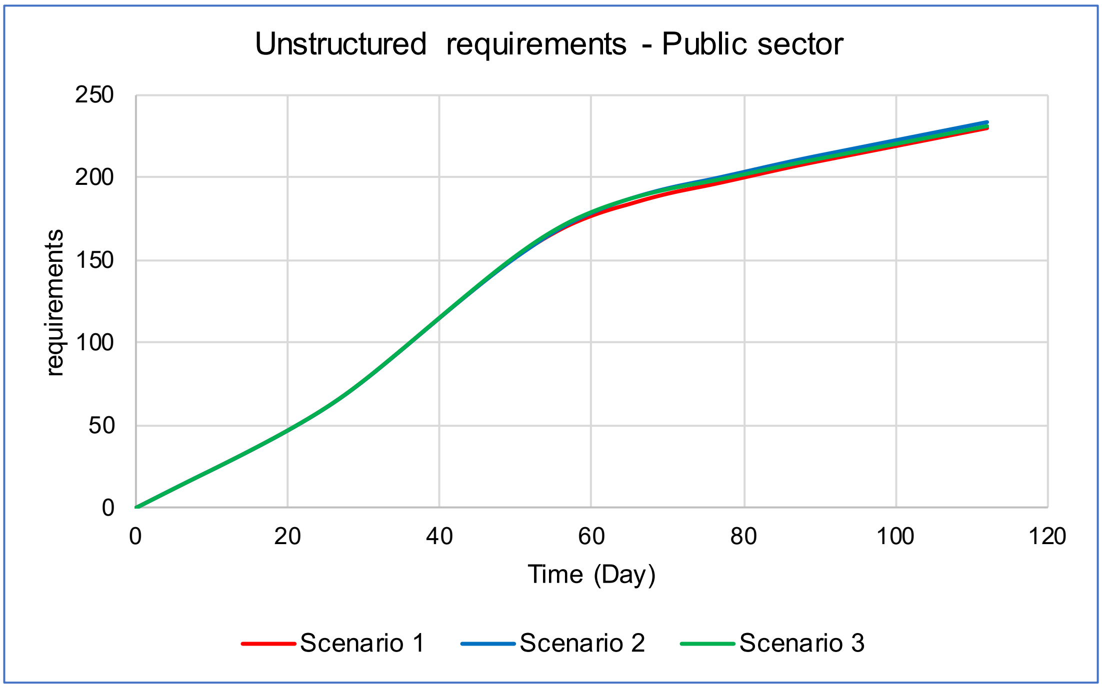


Figure 118: Single-case Experiment 3, SME-1 unstructured requirements behaviour for Scenario 1 to Scenario 3 – Public sector

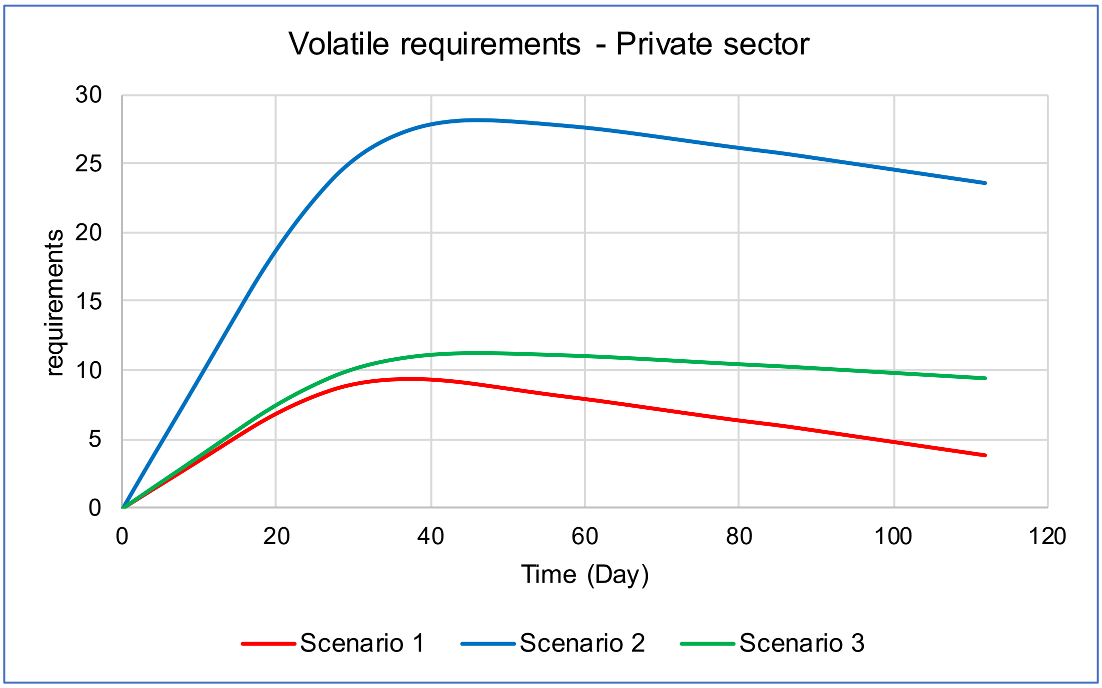


Figure 119: Single-case Experiment 3, SME-1 volatile requirements behaviour for Scenario 1 to Scenario 3 – Private sector

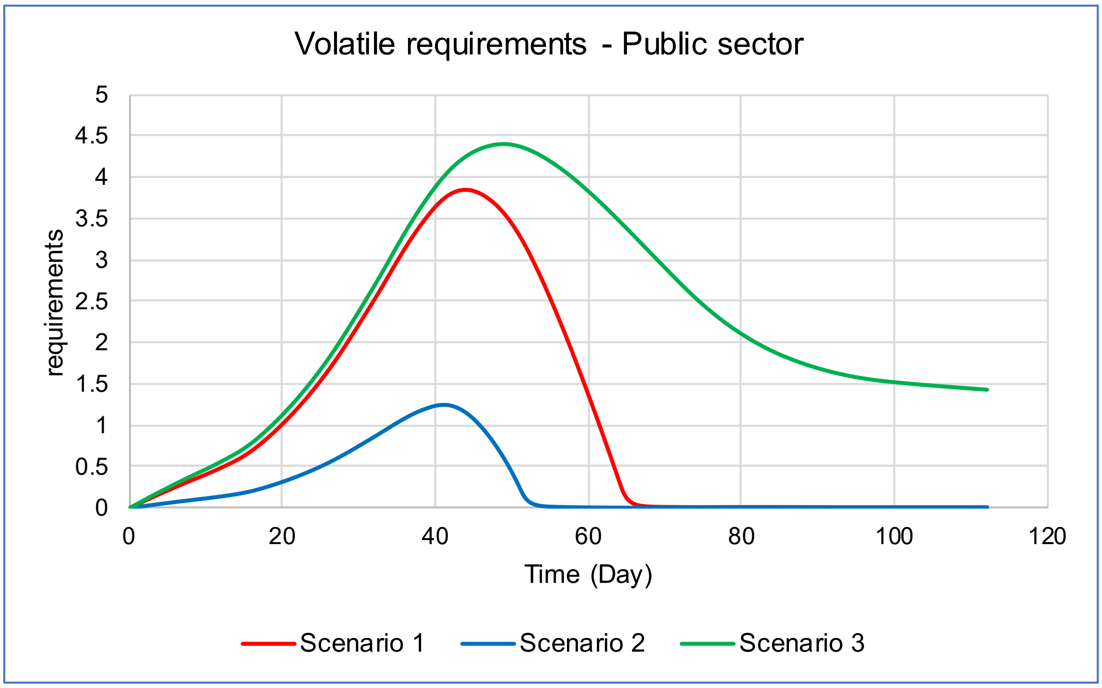


Figure 120: Single-case Experiment 3, SME-1 Volatile requirements behaviour for Scenario 1 to Scenario 3 – Public sector

6.4.3.3 Other observations and comments

When comparing the volatile requirements at the end of the time period, the maximum value for the public sector stakeholders are less than 0.5% whereas for the stakeholders from the private sector, this is in the order of 8%. This can be traced back to the elicitation behaviour where the private sector starts of quickly and at a high rate, but this quickly declines. The error creation process is also significant whereas the error detection process is slow and at a low activity rate.

In comparison, the requirements elicitation behaviour of the stakeholders from the public sector starts at a lower rate but then increase significantly. The error creation rate is also lower in comparison to the private sector, and where errors are made, they are detected earlier than in the private sector.

The result of this is that the stakeholders in the public sector seems to be more informed as to their needs and requirements in comparison to the private sector.

6.5 Evaluation of stakeholder needs and requirements versus the results of the validation process

The design of the requirements engineering research tool was approached from a systems engineering perspective based on the technical processes identified in the INCOSE Systems Engineering Handbook (INCOSE, 2015). During this process, the following stakeholder requirements or user requirements were identified (refer Table 15):

SH1 The requirements engineering research tool shall provide a mechanism that can be used to investigate behaviour in the requirements engineering domain.

SH2 The requirements engineering research tool shall provide a mechanism to evaluate potential solutions under simulated conditions without requiring implementation of these solutions in the real world.

SH3 The requirements engineering research tool shall support an ethical research process.

The solution validation process is shown in Figure 121 and address the validation of the stakeholder requirements SH1 to SH4. These stakeholder requirements were incorporated in the design and implementation of the requirements engineering research tool and were validated by means of three validation experiments, the results of which were presented in paragraph 6.4.1 to paragraph 6.4.3.

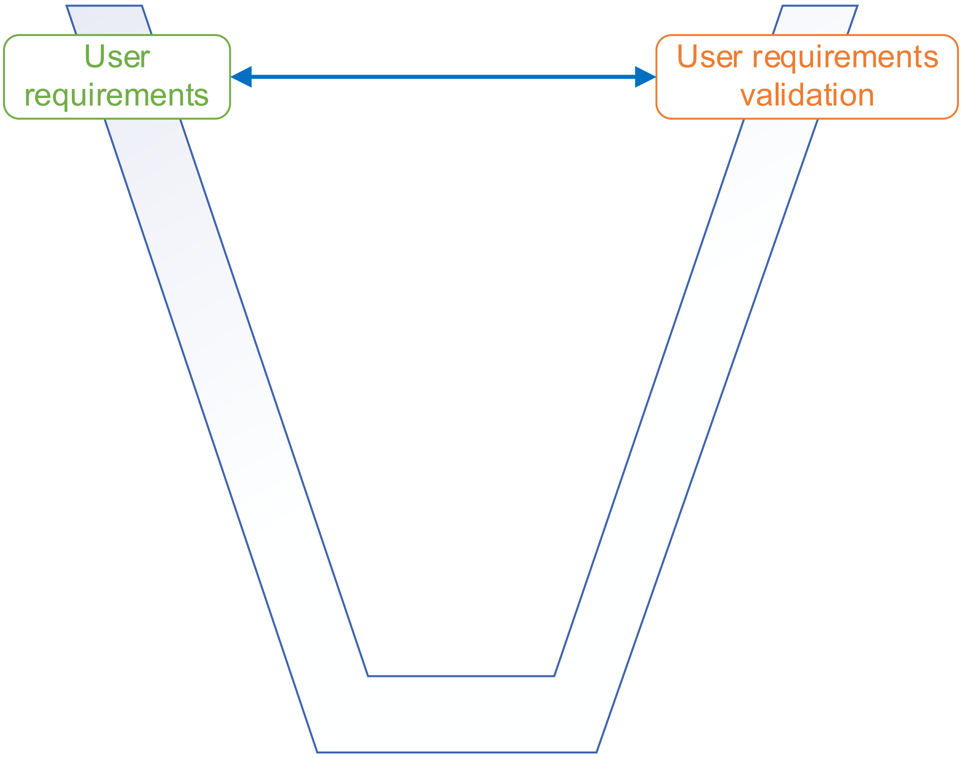


Figure 121: Stakeholder requirements validation (own contribution)

6.6 Chapter summary and conclusion

The aim of this chapter was to present the validation process as was used during the design cycle of the research tool. It was already established in Chapter 5, that the research tool consists of a measurement artefact and a simulation artefact. It was also further established that the research tool will be used to implement a single-case mechanism experiment. The validation process thus must address all these aspect.

The first aspect that validated was the measurement artefact. The measurement artefact consists of a questionnaire that was completed by the researcher together with the SMEs in an action research setting. An ideal reference mode was defined by creating a novel "*elicitation-diffusion"* model. These reference modes were used as guide against which the responses of the SMEs could be compared against in order to develop a first-order measurement of the health of the organisation. Aspects that could be determined from this comparison includes aspects such as the level of effort present in the requirements elicitation process as well as the delay times and behaviours for the error detection behaviour.

The second part of the research tool consisted of the simulation artefact. Here the simulation artefact that was design and verified in Chapter 5 was used. The qualitative data was prepared using a descriptive inference process. Once the data was prepared, it was used in the simulation model, and the output was evaluated in an abductive inference process.

CHAPTER 7. THESIS SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

"Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience." – Roger Bacon

7.1 Introduction

The purpose of this chapter is to summarise the research presented in this thesis, draw conclusions on the research questions posed as well as to make recommendations for future areas of research that have been identified but falls outside the scope of this thesis. During the research phase of this thesis a bibliography of more than 1000 articles were identified and reviewed. From this bibliography a selection of approximately 200 directly applicable books, articles and conference papers were used as references in order to form the theoretical basis for this thesis.

7.2 Summary of the research presented in this thesis

The point of departure for this thesis is that the world that we live in is an artificial world where the inhabitants are the creators of the artefacts. These artefacts will only be useful if they exhibit a particular utility which is required for a specific reason. The utility required is captured by the stakeholder needs and wants.

The process of discovering and eliciting this required utility falls in the domain of requirements engineering. It was argued that this process of discovering and eliciting is notoriously difficult and is one of the most common reasons for projects that are aimed at producing the required utility for the stakeholder, often fail or are challenged. I During the initial research process, it became clear that the system engineering domain and, within it, the requirements engineering domain that is primarily involved with the discovery and elicitation of requirements along with the other elements of the requirements engineering process, is a complex socio-technical system.

This realisation lead to the posing of the following research problem statement:

It is difficult to identify the correct requirements in a complex socio-technical environment (Paragraph 1.2)

Novel research contribution
The use of a design science research methodology to perform research in the requirements engineering

Solving problems outright in such a complex socio-technical system is not always achievable. Some researchers even go as far as to postulate that it is not possible to find a solution that will solve the problem completely. The literature suggests that an approach to problem investigation to improve the problem situation should be followed instead.

Novel research contribution
The integration of the systems engineering process as defined by INCOSE.

It was established from the literature that one of the possible research methodologies that can be applied to a complex socio-technical environment is that of a design science research methodology (paragraph 1.3). A design science research methodology is based upon the design and investigation of an artefact within its problem context. During the evaluation of the design science research process as was proposed by different authors, it was observed that while the empirical research effort was discussed in detail, the design aspects of the artefact were glossed over. The researcher at this stage decided to enhance the design cycle inherent in the design science research methodology by tailoring the systems engineering process as defined by INCOSE (paragraph 1.4).

Novel research contribution
The definition of a new object-of-study consisting of the original artefact and problem context as the new problem context and the research tool as the new artefact.

One of the central aspects of the design science research methodology is the concept of the object-of-study. This object of study consists not only of the artefact that is being designed nor the problem context to which this artefact is being applied but rather both the artefact as well as the problem context and most importantly, the interaction between the artefact and the problem context. Since it was identified that a research tool would be required, a new object-of-study was defined where the original artefact/problem context became the new problem context, and the research tool became the new artefact (Paragraph 1.4.1.6).

The first step in the research problem identification phase was to define the research goal that reflected the problem statement. The identified research goal is the following:

> *To explain the cause-effect behaviour observed within the requirements engineering domain that results in requirements volatility (Paragraph 2.3.1).*

The following instrument design goal was identified to support the research goal:

To improve the understanding of the requirements engineering process by designing and implementing a requirements engineering research tool that can be used to investigate and explain the phenomena observed in the requirements engineering process (Paragraph 2.3.1)

The following knowledge goal was identified to support the research goal:

To identify and describe the observed phenomena, their interactions, and the effect of these interactions as observed in the complex requirements engineering domain that may result in the incorrect set of requirements being used in the project implementation (Paragraph 2.3.1).

Once the research problem was identified the next step was to define the research problem in greater detail. This was done posing the following knowledge questions (refer to paragraph 3.2):

1. What are the observed phenomena?
2. What are the potential causes of these phenomena?
3. What are the mechanisms that cause these effects and what system components produced this event?
4. What are the contributions of these phenomena to the observed problems?
5. Is it possible to identify any specific sensitivity to any of the parameters?
6. Is there any outside motivation for the stakeholders or organisations that could explain the behaviour?

Consequential research contribution
The use of the structure of a causal loop diagram to identify the contributing factors to observed phenomena in the requirements engineering domain.

The next step was to define the research stakeholders (paragraph 3.3) as well as the research stakeholder goals and needs (paragraph 3.4). The observed phenomena were identified and described (paragraph 3.5.2). A causal loop diagram was used to identify and document the relationship between the different parameters identified that contributed to the observed phenomena (Figure 34). The structure of the causal loop diagram was further analysed to identify the primary and secondary contributing factors towards requirements volatility (Figure 35 to Figure 37). The structure of the causal loop diagram was further used to identify feedback loops that are present in the structure. Three feedback loops were identified and documented (Figure 38 to Figure 40). It was further established that all the feedback loops were reinforcing meaning that no single parameter could be used to control the behaviour. This confirmed the literature where it was stated that the best method to treat a complex, socio-technical system is to approach it as an improvement problem since there will not be any singular solutions.

The final activity of the research problem definition phase was the identification of the research stakeholder requirements (paragraph 3.9)

Novel research contribution
The application of Soft Systems Methodology to the design of a research tool.

The next phase in the research process related to establishing the specification of the research tool. This was done by deriving the research tool requirements. To facilitate this process, a mental model was based on the Soft Systems Methodology as defined by Checkland (1999). The application of this process led to identifying requirements related to the real-world of concern, the perceived real-world situation and the action needed to improve the situation (paragraph 4.2.2).

The research tool architecture was defined based on the research tool requirements (paragraph 4.4) as well as the research tool design specification (paragraph 4.5). A requirements traceability was performed to confirm that the research tool requirements could be linked back to the original research goals (Table 18).

The research tool was designed and implemented according to the requirements. This was done by firstly establishing the design specifications for the single-case mechanism experiment (paragraph 5.2) and the requirements engineering research tool (paragraph 5.3). It was identified that the research tool should consist of two artefacts, that of a measurement artefact and that of a simulation artefact. These two artefacts were designed, implemented and verified (paragraph 5.4 and 5.5). The measurement artefact consists of a research questionnaire that is to be completed by the researcher and the SME during a semi-structured interview in an action research setting (paragraph 5.4). The simulation artefact consists of a system dynamics simulation model that used the output of the measurement artefact to calculate the requirements volatility present in the system (paragraph 5.5). The output of this design phase was a verified research tool (paragraph 5.4.3 and paragraph 5.5.3).

The final phase in the research process was to validate the research tool. The validation process was presented in Chapter 6 and consisted of using the research tool as it was implemented and verified to perform three single-case mechanism experiment based on interviews with various SMEs.

Unique research contribution
The development and the application of an elicitation-diffusion model for use in requirements engineering.

The first step in the validation process was to define a reference model that could be used as an ideal sample that could be used as a template against which the results could be compared to as the first step in the abductive inference process. These reference models were generated using a novel elicitation-diffusion model (paragraph 6.2).

Three single-case mechanism experiments were conducted on a possible 55 data set. The important aspect to consider here is that the process intended to validate the research tool (paragraph 6.3 and 6.4). A brief analysis of the results was also provided (paragraph 6.4.1.3, 6.4.2.3 and 6.4.3.3). The final activity was to evaluate the stakeholder needs and requirements against the results of the validation process (paragraph 6.5)

7.3 Self-assessment

The main aim of the research presented in this thesis was firstly to gain a better understanding of why the requirements engineering process does not seem to work as intended. During the research process, the researcher realised that no single "silver bullet solution will be found, but that the problem should rather be approached from an improvement point of view. In order to be able to do this, the researcher set out to design a research tool that could be used firstly to establish a current performance baseline in the organisation that is being investigated, secondly to identify specific improvement actions, thirdly to provide an environment that can be used to evaluate potential improvements in a simulated environment prior to implementing these improvements in a real, live organisation.

These objects were met by applying a novel implementation of a design science research methodology resulting in a validated research tool. During the research process, several novel contributions were made to the research body of knowledge as was detailed in paragraph 7.2.

Unique research contribution
The research methodology and method presented is not only applicable to the requirements engineering domain but also the larger complex research domain as well as the complex socio-technical research domain.

The application of the research presented in this thesis is not unique to only the requirements engineering domain. It is the opinion of the researcher that the research methodology and the research process that was applied in this thesis can also be applied to other complex or complex socio-technical situations.

In conclusion of the self-assessment of the research presented in this thesis, it is the opinion of the researcher that the identified research problem was addressed and that in doing so, that the stated research goals, instrument design goals and knowledge goals were also successfully achieved.

7.4 Summary of the main conclusion researched in this thesis

The following main conclusions were researched in this thesis:

1. The system engineering domain and, within it, the requirements engineering domain, is a complex socio-technical system.
2. Conducting research in such a domain is problematic since describing the behaviour of the social element and generalising it are not always possible.
3. When conducting research in a socio-technical system to solve a problem, finding a complete and final solution may not always be feasible. In such a situation it is better to aim for a problem improvement approach.
4. The use of a design science research methodology as the base research methodology proved to be a valid approach.
5. Enhancing the design cycle by using the INCOSE systems engineering technical process as basis added value to the overall design.
6. The research methodology and research approach can be applied to the wider complex problem domains as well as the complex socio-technical problem domains.

7.5 Recommendations

1. The engineering curriculum, in general, has limited space for subjects other than those related to the core of engineering science. The fact is, however, that engineers deal with the needs and requirements of stakeholders and end-users daily. The ability and skill of engineers to deal with this complex socio-technical element do not currently form part of the standard engineering curriculum. Specialised training in aspects such as communication and conflict management can add great value to the toolkit that an engineer assembles to take into the workplace.
2. The use of a design science research methodology can be used not only to solve theoretical problems in the academic environment but can also potentially be applied with great success to real-life practical problems in industry.

7.6 Further research opportunities

The following two areas of further research can be identified:

1. Closing the loop by taking the results of the theoretical process back into the real-world environment and evaluating the results.
2. Expanding the elicitation-diffusion model to provide a tool for the requirements engineer to simulate different ideal-world scenarios.

7.7 Concluding comments

The parameters or elements that were identified to influence requirements volatility as well the relationship between them are not absolute and is but one of multiple scenarios that may occur. It is the responsibility of the research or requirements engineer to identify the correct set of parameter or elements as well as the relationship between them for the specific circumstances that are being researched.

REFERENCES

"There is more treasure in books than in all the pirate's loot on Treasure Island." – Walt Disney

Achimugu, P., Selamat, A., Ibrahim, R. and Mahrin, M. N. 2014. A Systematic Literature Review of Software Requirements Prioritization Research. *Information and Software Technology*. 56(6):568–585.

Adams, J., Khan, H. T. A., Raeside, R. and White, D. 2007. *Research Methods for Graduate Business and Social Science Students*. Thousands Oaks, CA: SAGE.

Ahmad, S. 2008. Negotiation in the Requirements Elicitation and Analysis Process. In: *Proceedings of the 19th Australian Software Engineering Conference*.

Al-Zawahreh, H. and Almakadmeh, K. 2015. Procedural Model of Requirements Elicitation Techniques. In: *IPAC '15 Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*. Batna, Algeria.

Alexander, I. F. 2005. A Taxonomy of Stakeholders. *International Journal of Technology and Human Interaction*. 1(1).

Aljafari, R. and Khazanchi, D. 2013. On the Veridicality of Claims in Design Science Research. In: *2013 46th Hawaii International Conference on System Sciences*.

Alspaugh, T. A. and Antón, A. I. 2008. Scenario support for effective requirements. *Information and Software Technology*. 50(3):198–220.

Amber, A., Bajwa, I. S., Naweed, M. S. and Bashir, T. 2011. Requirements Elicitation Methods. In: *2011 2nd International Conference on Mechanical, Industrial and Manufacturing Technologies*.

Anwar, F. and Razali, R. 2012. A Practical Guide to Requirements Elicitation Techniques Selection - An Empirical Study. *Middle-East Journal of Scientific Research*. 11(8):1059–1067.

Appan, R. and Browne, G. J. 2012. The Impact of Analyst-Induced Misinformation on the Requirements Elicitation Process. *MIS Quarterly*. 36(1):85–106.

Aranda, G. N., Vizcaíno, A. and Piattini, M. 2010. A framework to improve communication during the requirements elicitation process in GSD projects. *Requirements Engineering*. 15(4):397–417.

Aranda, P. J. V. 2007. *A Requirements Engineering Approach for the Development of Web Applications*. PhD thesis, Technical University of Valencia, Spain.

Arnold, R. D. and Wade, J. P. 2017. A Complete Set of Systems Thinking Skills. *Insight*. 20(3):9–17.

Ary, D., Jacobs, L. C., Sorenson, C. and Razavieh, A. 2010. *Introduction to Research in Education*. 8th edn. Belmont, CA: Wadsworth Cengage Learning.

Attarzadeh, I. and Ow, S. H. 2008. Project Management Practices: The Criteria for Success or Failure. *Communications of the IBIMA*. 1(28):234–241.

Baiyere, A., Hevner, A. R., Rossi, M., Gregor, S. and Baskerville, R. L. 2015. Artifact and / or Theory? Publishing Design Science Research in IS. In: *Thirty Sixth International Conference on Information Systems*. Fort Worth, Texas.

Bar-Yam, Y. 1997. *Dynamics of Complex Systems*. Redmond, WA: CRC Press.

Barlas, Y. 1994. Model Validation in System Dynamics. In: *Proceedings of the 1994 International System Dynamics Conference*.

Baskerville, R. L. 2008. What design science is not. *European Journal of Information Systems*. 17(5).

Baskerville, R. L., Baiyere, A., Gergor, S., Hevner, A. R. and Rossi, M. 2018. Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems*. 19(5):358–376.

Baskerville, R. L., Pries-Heje, J. and Venable, J. R. 2009. Soft Design Science Methodology. In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*.

Berenbach, B., Paulish, D. J., Kazmeier, J. and Rudorfer, A. 2009. *Software & Systems Requirements Engineering: In Practice*. New York, NY: McGraw-Hill, Inc.

Blanchard, B. S. and Fabrycky, W. J. 1990. *System Engineering and Analysis*. 2nd edn. Englewood Cliffs, NJ: Prentice-Hall.

Bochmann, G. 2009. Requirements Elicitation Techniques. Lecture Notes, University of Ottawa, Ottawa, Canada.

Bostrom, R. P. and Heinen, J. S. 1977a. MIS Problems and Failures : A Socio-Technical Perspective, Part I: The Causes. *MIS Quarterly*. (September):17–32.

Bostrom, R. P. and Heinen, J. S. 1977b. MIS Problems and Failures : A Socio-Technical Perspective, Part II : The Application of Theory. *MIS Quarterly*. (December):11–28.

Boulila, N., Hoffmann, A. and Herrmann, A. 2011. Using Storytelling to Record Requirements: Elements for an Effective Requirements Elicitation Approach. In: *2011 4th Int. Workshop on Multimedia and Enjoyable Requirements Engineering. Beyond Mere Descriptions and with More Fun and Games (Co-located with the 19th IEEE Int. Requirements Eng. Conf.)*. :9–16.

Broll, G., Hussmann, H., Rukzio, E. and Wimmer, R. 2007. Using Video Clips to Support Requirements Elicitation in Focus Groups - An Experience Report. In: *2nd International Workshop on Multimedia Requirements Engineering: Conference on Software Engineering*.

Brooks, F. P. 1986. No silver bullet: Essence and Accidents of Software Engineering. In: *Proceedings of the International Federation for Information Processing 10th World Computing Congress*. Dublin, Ireland.

Bubenko, J. A. 1995. Challenges in Requirements Engineering. In: *Proceedings of 1995 IEEE International Symposium on Requirements Engineering*.

Bunge, M. 2004. How Does it Work? The Search for Explanatory Mechanisms. *Philosophy of the Social Sciences*. 34(2):182–210.

Burgelman, R. A. and Maidique, M. A. 1988. *Strategic Management of Technology and Innovation*. 1st ed. Homewood, IL: Richard D. Irwin.

Burnay, C. 2016. Are Stakeholders the Only Source of Information for Requirements Engineers? Toward a Taxonomy of Elicitation Information Sources. *ACM Transactions on Management Information Systems*. 7(3).

Carlshamre, P. 2001. *A Usability Perspective on Requirements Engineering*. PhD thesis, Linköpings Universitet, Linköping, Sweden.

Carrizo, D., Dieste, O. and Juristo, N. 2014. Systematizing Requirements Elicitation Technique Selection. *Information and Software Technology*. 56(6):644–669.

Cash, P. J. 2018. Developing theory-driven design research. *Design Studies*. 56:84–119.

Chakraborty, S., Sarker, Saonee and Sarker, Suprateek. 2010. An Exploration into the Process of Requirements Elicitation : A Grounded Approach. *Journal of the Association for Information Systems*. 11(4):212–249.

Charette, R. N. 2005. Why software fails. *IEEE Spectrum*. (September).

Checkland, P. B. 1999. *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. Chichester, West Sussex: John Wiley & Sons.

Checkland, P. B. and Holwell, S. 2007. Action Research: Its Nature and Validity. In: Kock, N. F. (ed.) *Information Systems Action Research An Applied View of Emerging Concepts and Methodologies*. New York, New York, USA: Springer Science+Business Media, LLC. :3–17.

Christel, M. G. and Kang, K. C. 1992. *Issues in Requirements Elicitation*. *CMU/SEI-92-TR-12*. Pittsburgh, PA: Carnegie-Mellon University.

Chua, B. B., Bernardo, D. V. and Verner, J. 2010. Understanding the Use of Elicitation Approaches for Effective Requirements Gathering. In: *Proceedings of the 5th International Conference on Software Engineering Advances*. :325–330.

Coulin, C. R. 2007. *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation*. PhD Thesis, Université Paul Sabatier, Toulouse, France.

Cui, Z. and Loch, C. H. 2014. A Rational Framework on the Causes and Cures of Collaborative Projects Failure. In: *Proceedings of Portland International Center for Management of Engineering and Technology Conference: Infrastructure and Service Integration*. Kanazawa, Japan.

Daniels, C. B. and Lamarsh, W. J. 2007. Complexity as a Cause of Failure in Information Technology Project Management. In: *2007 IEEE International Conference on System of Systems Engineering*.

Davis, A. M. 2003. The art of requirements triage. *Computer*. 36(3):42–49.

Davis, C. J., Fuller, R. M., Tremblay, M. C. and Berndt, D. J. 2006. Communication Challenges in Requirements Elicitation and the Use of the Repertory Grid Technique. *Journal of Computer Information Systems*. International Association for Computer Information Systems. 46(5):78–86.

Delater, A. 2013. *Tracing Requirements and Source Code during Software Development*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, Germany.

Dickson, A., Hussein, E. K. and Agyem, J. A. 2018. Theoretical and conceptual framework: Mandatory ingredients of a quality research. *International Journal of Scientific Research*. 7(1):438–441.

Dieste, O., Lopez, M. and Ramos, F. 2008. Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques. In: *11th Workshop on Requirements Engineering*. :96–103.

Dorsey, P. 2005. Top 10 Reasons Why Systems Projects Fail. Woodbridge, NJ: Dulcian, Inc.

Duggan, J. 2017. Implementing a Metapopulation Bass Diffusion Model using the R Package deSolve. *The R Journal*. 9(1):153–163.

Easterbrook, S. M. 1991a. *Elicitation of Requirements from Multiple Perspectives*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, London.

Easterbrook, S. M. 1991b. Handling Conflict Between Domain Descriptions with Computer-supported Negotiation. *Knowledge Acquisition*. 3(3):255–289.

Elster, J. 2015. *Explaining Social Behavior: More Nuts and Bolts for the Social Sciences*. Cambridge, England: Cambridge University Press.

Emery, F. E. and Trist, E. L. 1960. Sociotechnical Systems. In: Churchman, C. and Verhulst, M. (eds) *Management Sciences, Models and Techniques: Vol. 2.* New York, NY: Pergamon.

Ernst, N. A. 2012. *Software Evolution: a Requirements Engineering Approach*. PhD thesis, University of Toronto, Toronto, Canada.

Faulk, S. R. 1997. Software requirements: A tutorial. In: Thayer, R. and Dorfman, M. (eds) *Software Requirements Engineering 2nd Edition*. Washington, DC: IEEE Computer Society Press.

Ferreira, S., Collofello, J., Shunk, D. and Mackulak, G. 2009. Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation. *Journal of Systems and Software*. Elsevier Inc. 82(10):1568–1577.

Flood, R. L. and Jackson, M. C. 1991. *Creative Problem Solving - Total Systems Intervention*. West Sussex, England: Wiley.

Forrester, J. W. 1986. Lessons From System Dynamics Modeling. In: *1986 International Conference of the System Dynamics Society*. Sevilla, Spain.

Forsberg, K., Mooz, H. and Cotterman, H. 2005. *Visualizing Project Management*. 3rd ed. Hoboken, New Jersey: John Wiley & Sons Inc.

Franklin, R. D., Allison, D. B. and Gorman, B. S. 1996. *Design and Analysis of Single-Case research*. East Sussex, England: Psychology Press.

Frese, R. and Sauter, V. 2006. *Project Success and Failure: What is Success, What* *is Failure, and how Can You Improve Your Odds for Success*. St. Louis, MO: University of Missouri-St. Louis.

Frost, J. and Osterloh, M. 2003. Dialogue Devices : Bridging between ‘ Mode 1 ’ and ‘ Mode 2 ’ Knowledge Production. In: Müller, A. and Kieser, A. (eds) *Communication in organizations: Structures and practices*. Frankfurt am Main, Germany: Peter Lang.

Fuentes-Fernández, R., Gómez-Sanz, J. J. and Pavón, J. 2010. Understanding the Human Context in Requirements Elicitation. *Requirements Engineering*. 15(3):267–283.

Fuentes, J. M., Fraga, A., Génova, G., Parra, E., Alvarez, J. M. and Llorens, J. 2016. Applying INCOSE Rules for writing high-quality requirements in Industry. In: *26th Annual INCOSE International Symposium (IS2016)*. Edinburgh, Scotland. :1875–1889.

Geerts, G. L. 2011. A Design Science Research Methodology and its Application to Accounting Information Systems Research. *International Journal of Accounting Information Systems*. 12(2):142–151.

Geisser, M. and Hildenbrand, T. 2006. A Method for Collaborative Requirements Elicitation and Decision-supported Requirements Analysis. *IFIP International Federation for Information Processing*. 219:108–122.

Geneca LLC. 2011. *Why a Majority of Business and IT Teams Anticipate Their Software Development Projects Will Fail*.

Glinz, M. and Wieringa, R. J. 2007. Stakeholders in requirements engineering. *IEEE Software*. 24(2):312–335.

Goguen, J. A. and Linde, C. 1993. Techniques for requirements elicitation. *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*.

Goodman, M. 2017. *From causal loop diagram to computer models - Part I*. *The Systems Thinker*.

Gregg, D. G., Kulkarni, U. R. and Vinzé, A. S. 2001. Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. *Information Systems Frontiers*. 3(2):169–183.

Gregor, S. and Jones, D. 2007. The Anatomy of a Design Theory. *Journal of the Association for Information Systems*. 8(5):312–335.

Hadar, I., Soffer, P. and Kenzi, K. 2014. The Role of Domain Knowledge in Requirements Elicitation via Interviews: An Exploratory Study. *Requirements Engineering*. 19(2):143–159.

Halligan, R. J. 2012. *Requirements Analysis that Works*. report no. PPI-005261. Victoria, Australia: Project Performance International.

Hamid, S.-R., Chew, B.-C. and Halim, S. 2012. What’s the Principles of Technology Management–Eliciting Technology Management Principles through Expert Opinion. *International Journal of Innovation, Management and Technology*. 3(5):631–636.

Harris, H. J. 2006. *Requirements Dilema*. PhD dissertation, Brunel University;

London England.

Haskins, B. 2016. Implementing a Structured Verification Framework to Improve Verification Requirements Quality. In: *26th Annual INCOSE International Symposium (IS2016)*. Edinburgh, Scotland. :877–891.

Hastie, S. and Wojewoda, S. 2015. *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. Available from: http://www.infoq.com/articles/standish-chaos-2015 [Accessed 20 December 2015].

Haumer, P. 2000. *Requirements Engineering with Interrelated Conceptual Models and Real World Scenes*. PhD dissertation, Rheinisch-Westfälischen Technischen Hochschule, Aachen, Germany.

Hengstler, M. 2016. Innovation ecosystems: how do orchestrating firms build trust and control? In: *International Association for Management of Technology IAMOT 2016 Conference Proceedings*. :1832–1852.

Hevner, A. R., March, S. T., Park, J. and Ram, S. 2004. Design Science Research in Information Systems. *MIS quarterly*. 28(1):75–105.

Hevner, A. R. 2007. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*. 19(2):87–92.

Hevner, A. R. and Chatterjee, S. 2010. *Design Research in Information Systems*. London, England: Springer Science + Business Media.

Heydari, B. and Pennock, M. J. 2018. Guiding the Behavior of Socio-Technical Systems : The Role of Agent-Based Modeling. *System Engineering*. 21(3):1–17.

Hickey, A. M. and Davis, A. M. 2003a. Elicitation technique selection: how do experts do it? In: *Proceedings of the 11th IEEE International Requirements Engineering Conference 2003*.

Hickey, A. M. and Davis, A. M. 2003b. Requirements Elicitation and Elicitation Technique Selection : A Model for Two Knowledge-Intensive Software Development Processes. In: *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, (HICSS'03).*

Hickey, A. M. and Davis, A. M. 2004. A Unified Model of Requirements Elicitation. *Journal of Management Information Systems*. M.E. Sharpe Inc. 20(4):65–84.

Honour, E. C. 2018. A historical perspective on systems engineering. *Systems Engineering*. 21(3):148–151.

Hooks, I. F. 1993. Writing Good Requirements: A Requirements Working Group Information Report. In: *Proceedings of the Third International Symposium of the NCOSE - Volume 2*.

Iivari, J. 2007. A Paradigmatic Analysis of Information Systems As a Design Science. *Scandinavian Journal of Information Systems*. 19(2):39–64.

Iivari, J. 2015. Distinguishing and Contrasting Two Strategies for Design Science Research. *European Journal of Information Systems*. 24(1):107–115.

INCOSE. 2015. *Systems engineering handbook*. 4th ed. San Diego, CA: John Wiley

& Sons.

INCOSE. 2017. *Guide for writing requirements*. report no. INCOSE-TP-2010-006-02.1. INCOSE: San Diego, CA.

ISO/IEC/IEEE. 2011. *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*. report no. ISO/IEC/IEEE 29148. International Organization for Standardization, International Electrotechnical Commission, IEEE.

Jabareen, Y. 2009. Building a Conceptual Framework: Philosophy, Definitions, and Procedure. *International Journal of Qualitative Methods*. 8(4):49–62.

Jackson, M. C. 2002. *Systems Approaches to Management*. New York, NY: Kluwer Academic.

Jackson, M. C. 2003. *Systems Thinking: Creative Holism for Managers*. West Sussex, England: John Wiley & Sons.

Jarke, M. and Pohl, K. 1994. Requirements Engineering in 2001: (Virtually) Managing a Changing Reality. *Software Engineering Journal*. 9(6):257–266.

Järvinen, P. 2007. Action research is similar to design science. *Quality and Quantity*. 41(1):37–54.

Javed, T., Maqsood, M. E. and Durrani, Q. S. 2004. A Study to Investigate the Impact of Requirements Instability on Software Defects. *ACM Software Engineering Notes*. 29(4):1–7.

Jeary, S. P. 2010. *The Derivation of a Pragmatic Requirements Framework for Web Development*. PhD thesis, Bournemouth University, Poole, England.

Johannesson, P. and Perjons, E. 2014. *An Introduction to Design Science*. Kista, Sweden: Springer.

Johnson, J. and Mulder, H. 2016. CHAOS Chronicles , Focusing on Failures and Possible Improvements in IT Projects. *The 10th International Multi-Conference on Society, Cybernetics and Informatics: IMSCI 2016*. 14(5).

Jones, C. 2006. Why Projects Fail - Social and Technical Reasons for Software Project Failures. *Crosstalk - The Journal of Defense Software Engineering*. 19(6):4–9.

Jrad, R. B. N., Ahmed, M. D. and Sundaram, D. 2014. Insider Action Design Research: A multi-methodological Information Systems research approach. In: *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*. IEEE.

Kaiya, H. and Saeki, M. 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation. In: *14th IEEE International Requirements Engineering Conference (RE'06)*. :189–198.

Kappelman, L. A., McKeeman, R. and Zhang, L. 2006. Early Warning Signs of it Project Failure: The Dominant Dozen. *Information Systems Management*. 23(4):31–36.

Kerzner, H. R. 2013. *Project management: A Systems Approach to Planning,* *Scheduling, and Controlling*. 11th edn. New York, NY: John Wiley & Sons.

Khan, N. U. and McLucas, A. C. 2008. Application of Systems Engineering Vee Model to Enhance System Dynamics Modelling Methodology. In: *The 2008 International Conference of the System Dynamics Society*. Athens, Greece.

Kim, D. 2017. *Guidelines for drawing casual loop diagrams*. *The Systems Thinker*.

Laplante, P. 2009. *Requirements Engineering for Software and Systems*. Boca Raton, FL: CRC Press.

Lee, A. S. 2007. Action is an Artifact: What Action Research and Design Science Offer to Each Other. In: Kock, N. F. (ed.) *Information Systems Action Research An Applied View of Emerging Concepts and Methodologies*. New York, New York, USA: Springer Science+Business Media, LLC. :43–60.

Lempia, D. L., Schindel, B., Mcgill, S. and Graber, M. 2016. Using Visual Diagrams and Patterns for Consistent and Complete Requirements. In: *26th Annual INCOSE International Symposium*. Edinburgh, Scotland. :415–429.

Lewis, A. 2019. WordWeb Pro Software.

Lim, S. L. 2010. *Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation*. Phd dissertation, University of New South Wales, Sidney, Australia.

March, S. T. and Smith, G. F. 1995. Design and Natural Science Research on Information Technology. *Decision Support Systems*. 15(4):251–266.

Marnewick, A. 2013. *A Socio-technical View of the Requirements Engineering Process*. PhD thesis, University of Johannesburg, Johannesburg, South Africa.

Maxwell, J. A. 2013. *Qualitative Research Design - An Interactive Approach*. 3rd edn. Edited by L. Bickman and D. J. Rog. Sage Publications.

Méndez Fernández, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., Conte, T., Christiansson, M. T., Greer, D., Lassenius, C., *et al.* 2017. Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice. *Empirical Software Engineering*. 22(5):2298–2338.

Midgley, G. 2016. Four Domains of Complexity. *Emergence: Complexity and Organization*. 18(2).

Mohamadi, M. and Ranjbaran, T. 2013. Effective Factors on the Success or Failure of the Online Payment System, Focusing on Human Factors. In: *7th International Conference on e-Commerce in Developing Countries with a focus on e-Security*. Kish Island, Iran.

Morisio, M., Egorova, E. and Torchiano, M. 2007. Why Software Projects Fail? Empirical Evidence and Relevant Metrics. In: *Proceedings of the IWSM - Mensura 2007*. :299–308.

Mouton, J. 2001. *How to Succeed in Your Master's & Doctoral Studies*. 1st ed. Pretoria, South Africa: Van Schaik.

NASA. 2017. *NASA Systems Engineering Handbook*. *NASA/SP-2016-6105 Rev 2*.

Washington, DC: NASA.

Nunamaker Jr., J. F., Chen, M. and Purdin, T. D. M. 1990. Systems development in information systems research. *Journal of Management Information Systems*. 7(3).

Nurmuliani, N., Zowghi, D. and Williams, S. P. 2006. Requirements Volatility and Its Impact on Change Effort: Evidence-based Research in Software Development Projects. In: *Proceedings of the Eleventh Australian Workshop on Requirements Engineering*.

Oosthuizen, R. 2014. *Modelling methodology for assessing the impact of new technology on complex sociotechnical systems*. Phd thesis, University of Pretoria, Pretoria, South Africa.

Pajerek, L. A. 1998. Requirements Management Starter Kit. In: *System Engineering Society of Australia / INCOSE SE98*.

Panchal, J. H. and Szajnfarber, Z. 2017. Experiments in systems engineering and design research. *Systems Engineering*. 20(6):529–541.

Pearce II, J. A. and Robinson Jr, R. B. 1991. *Strategic Management - Formulation, Implementation, and Control*. 4th ed. Boston, MA: Richard D. Irwin.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. 2006. The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. *Proceedings of Design Research in Information Systems and Technology DESRIST'06*. 24.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*. 24(3):45–77.

Peffers, K., Tuunanen, T. and Niehaves, B. 2018. Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems*. 27(2):129–139.

Pinto, J. K. and Mantel, S. J. 1990. The causes of project failure. *IEEE Transactions on Engineering Management*. 37(4):269–276.

PMBoK. 2008. *A Guide to the Project Management Body of Knowledge*. 4th ed. Newtown Square, PA: Project Management Institute.

PPI. 2007. *Types of Requirements*. P007-XBIO-001717–8.

Pruyt, E. 2013. *Small system dynamics models for big issues: Triple Jump towards Real-World Complexity*. 1st ed. Delft, The Netherlands: TU Delft Library.

Rahi, S. 2017. Research Design and Methods: A Systematic Review of Research Paradigms, Sampling Issues and Instruments Development. *International Journal of Economics & Management Sciences*. 06(02):1–5.

Ratsiepe, K. B. and Yazdanifard, R. 2011. Poor risk management as one of the major reasons causing failure of project management. In: *International Conference on Management and Service Science, MASS 2011*.

Ravitch, S. M. and Riggan, M. 2017. *Reason & Rigor*. 2nd ed. Thousand Oaks, CA:

SAGE.

Rayson, P., Garside, R. and Sawyer, P. 1999. Recovering legacy requirements. In: *5th International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'99)*.

Reifer, D. J. 2000. Requirements Management : The Search for Nirvana. *IEEE Software*. 17(3):45–47.

Richards, D. 2000. A Process Model for Requirements Elicitation. In: *Proceedings of The 11th Australasian Conference on Information Systems (ACIS)*. Brisbane, Australia. :1–12.

Richardson, K. A. 2004. Systems theory and complexity: Part 1. *Emergence: Complexity and Organization*. 6(3).

Ropohl, G. 1997. Knowledge Types in Technology. *International Journal of Technology and Design Education*. 7(1–2):65–72.

Rost, J. 2004. Political Reasons for Failed Software Projects. *IEEE Software*. 21(6):102–104.

Rutherford, A. 2018. *The Systems Thinker: Essential Thinking Skills For Solving Problems, Managing Chaos, and Creating Lasting Solutions in a Complex World*. 1st ed. Kindle Direct Publishing.

Ryan, M. J. 2013. An Improved Taxonomy for Major Needs and Requirements Artifacts. In: *Proceedings of the 23rd Annual INCOSE International Symposium*. Philadelphia, PA.

Ryan, M. J. and Wheatcraft, L. S. 2017. On a Cohesive Set of Requirements Engineering Terms. *Systems Engineering*. 20(2):118–130.

Sauer, C., Gemino, A. and Reich, B. H. 2007. The Impact of Size and Volatility on IT Project Performance. *Communications of the ACM*. 50(11):79–84.

Schurz, G. 2008. Patterns of abduction. *Synthese*. 164(2):201–234.

Scribante, N., Pretorius, L. and Benade, S. J. 2015. In the Beginning...-Challenges in Requirements Elicitation. In: Sparrius, A. (ed.) *Proceedings of the 11th SA INCOSE Conference*. Pretoria, South Africa.

Scribante, N., Pretorius, L. and Benade, S. J. 2016a. A Conceptual System Dynamics Model for Requirements Engineering. In: Musango, J. K. and Brent, A. C. (eds) *Proceeding of the 4th Annual System Dynamics Conference in South Africa*. Stellenbosch, South Africa.

Scribante, N., Pretorius, L. and Benade, S. J. 2016b. Conflict in the Requirements Engineering Process. In: Sparrius, A. (ed.) *Proceedings of the 12th INCOSE SA Systems Engineering Conference*. Pretoria, South Africa.

Scribante, N., Pretorius, L. and Benade, S. J. 2016c. Elements of a System Dynamics Model for Requirements Elicitation. In: *International Association for Management of Technology IAMOT 2016 Conference Proceedings*. Orlando, FL.

Scribante, N., Pretorius, L. and Benade, S. J. 2017a. A Single Stage System

Dynamics Simulation Model for Requirements Engineering. In: Sparrius, A. (ed.) *Proceedings of the 13th INCOSE SA Systems Engineering Conference*. Pretoria: International Council on Systems Engineering (INCOSE) South Africa.

Scribante, N., Pretorius, L. and Benade, S. J. 2017b. Requirements Engineering Principles Applicable to Technology and Innovation Management. In: Kocaoglu, D. F. (ed.) *2017 Proceedings of PICMET '17: Technology Management for Interconnected World*.

Scribante, N., Pretorius, L. and Benade, S. J. 2018a. Applying a Design Science Research Methodology to the design of a research instrument. In: Sparrius, A. (ed.) *Proceedings of the 14th INCOSE SA Systems Engineering Conference*. Pretoria, South Africa.

Scribante, N., Pretorius, L. and Benade, S. J. 2018b. Elements of a Systems Dynamics Simulation Model for Requirements Engineering. In: Nunes, B. et al. (eds) *27th Annual Conference of the International Association for Management of Technology*. Birmingham, UK.

Sendall, S. and Strohmeier, A. 2002. *Requirements elicitation with use cases. Lecture Notes in Software Engineering Course 2001-2002*. Springer-Verlag.

Serna, E., Bachiller, O. and Serna, A. 2017. Knowledge meaning and management in requirements engineering. *International Journal of Information Management*. Elsevier Ltd. 37(3):155–161. Available at: http://dx.doi.org/10.1016/j.ijinfomgt.2017.01.005.

Sharma, S. and Pandey, S. K. 2013. Revisiting Requirements Elicitation Techniques. *International Journal of Computer Applications (0975 – 8887)*. 75(12):35–39.

Sharp, H., Finkelstein, A. and Galal, G. 1999. Stakeholder identification in the requirements engineering process. In: *Proceedings of the 10th International Workshop on Database and Expert Systems Applications. DEXA 99*.

Sheard, S. A. 2013. Systems Engineering Complexity in Context. In: *INCOSE International Symposium*. :1145–1158.

Sheard, S. A. and Mostashari, A. 2009. Principles of complex systems for systems engineering. *Systems Engineering*. 12(4):295–311.

Simon, H. A. 1996. *The Sciences of the Artificial*. 3rd ed. Cambridge, MA: MIT Press.

Skyttner, L. 2001. *General systems theory: ideas & applications*. River Edge, NJ: World Scientific Publishing.

Smit, A., Brent, A. C. and Musango, J. K. 2014. Engineering Education – A Systems Dynamics View. In: *INCOSE International Symposium*. :382–397.

Sommerville, I. 2005. Integrated Requirements Engineering: A Tutorial. *IEEE Software*. 22(1):16–23.

Sommerville, I. 2011. *Software Engineering*. 9th ed. Boston, MA: Addison-Wesley.

Sommerville, I. and Sawyer, P. 1997. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of System Engineering*.

3(1):101–130.

Sommerville, I. and Sawyer, P. 2003. *Requirements Engineering: A Good Practice Guide*. Chichester, England: John Wiley & Sons.

Sterman, J. D. 2000. *Business Dynamics - Systems Thinking and Modeling for a Complex World*. Boston, MA: Irwin McGraw-Hill.

Sterman, J. D. 2017. *Fine-tuning your causal loop diagram - Part I. The Systems Thinker*.

Strang, K. D. 2015. *The Palgrave Handbook of Research Design in Business and Management*. 1st ed. New York, NY: Palgrave Macmillan.

Szajnfarber, Z. and Gralla, E. 2017. Qualitative methods for engineering systems: Why we need them and how to use them. *Systems Engineering*. 20(6):497–511.

Ter_Mors, A. 2009. *The world according to MARP*. PhD thesis, Technische Universiteit Delft, Delft, The Netherlands.

Thun, J.-H., Größler, A. and Milling, P. M. 2000. The Diffusion Of Goods Considering Network Externalities : A System Dynamics-Based Approach. In: *Proceedings of the 18th International Conference of the System Dynamics Society*. :6–10.

Trist, E. L. 1980. The Evolution of Socio-technical Systems. In: *Conference on Organizational Design and Performance*.

Twiss, B. 1990. *Managing Technological Innovation*. 3rd ed. London, England: Longman Group UK.

Vaishnavi, V. and Kuechler, B. 2004. *Design Science Research in Information Systems*. *Design Science Research in Information Systems*. Available at: http://www.desrist.org/design-research-in-information-systems/ (Accessed: 15 November 2015).

Vaishnavi, V. and Kuechler, W. 2015. *Design Science Research Methods and Patterns*. 2nd ed. Boca Raton, FL: CRC Press.

Van_Aken, J. E., Chandrasekaran, A. and Halman, J. 2016. Conducting and publishing design science research: Inaugural essay of the design science department of the Journal of Operations Management. *Journal of Operations Management*. 47–48:1–8.

Van_Dyk, D. J. 2013. *Sustainability if quality improvement programs in a heavy engineering manufacturing environment: A systems dynamics approach*. PhD thesis, University of Pretoria, Pretoria, South Africa.

Van_Lamsweerde, A. 2000. Requirements Engineering in the Year 00: A Research Perspective. In: *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*. Limerick, Ireland: ACM Press. :5–19.

Venable, J. R. 2006. The Role of Theory and Theorising in Design Science Research. In: *Proceedings of the 1st International Conference on Sesign Science in Information Systems and Technology (DESRIST 2006)*. :1–18.

Venable, J. R., Pries-Heje, J. and Baskerville, R. L. 2016. FEDS: a Framework for

Evaluation in Design Science Research. *European Journal of Information Systems*. 25(1):77–89.

Verner, J., Sampson, J. and Cerpa, N. 2008. What factors lead to software project failure? In: *Proceedings of the 2nd International Conference on Research Challenges in Information Science, RCIS 2008*. IEEE. :71–80.

Wessels, A. 2012. *The Development of Complex Systems : an Integrated Approach To Design Influencing*. PhD thesis, University of Pretoria, Pretoria, South Africa.

Whitworth, B. 2012. A Social Environment Model of Socio-technical Performance. *International Journal of Networking and Virtual Organisations*. 11(1):1–24.

Wiegers, K. E. 2000. When Telepathy Won't Do: Requirements Engineering Key Practices. *Cutter IT Journal*. 13(5):9–15.

Wieringa, R. J. 2014. *Design Science Methodology for Information Systems and Software Engineering*. New York, NY: Springer-Verlag.

Williams, D. 2001. Towards A System Dynamics Theory of Requirements Engineering Process. *Proceedings of the 19th International Conference of the System Dynamics Society*. :1–24.

Williams, D. and Kennedy, M. 2000. Towards a Model of Decision-Making for Systems Requirements Engineering Process Management. In: *International System Dynamics 2000 Conference*. Bergen, Norway.

Williams, T., Klakegg, O. J., Walker, D. H. T., Andersen, B. and Magnussen, O. M. 2012. Identifying and Acting on Early Warning Signs in Complex Projects. *Project Management Journal*. 43(2):37–53.

Williamson, J. 2011. Mechanistic Theories of Causality Part I & II. *Philosophy Compass*. 6(6):421–444.

Yilmaz, D. and Kılıçoğlu, G. 2013. Resistance to change and ways of reducing resistance in educational organizations. *European Journal of Research on Education*. 1(1):14–21.

Young, L. Z., Farr, J. V. and Valerdi, R. 2010. The role of complexities in system engineering cost estimate process. In: *Conference on System Engineering Research*. Hoboken, NJ.

Young, R. R. 2004. *The requirements engineering handbook*. Norwood, MA: Artech House.

Zowghi, D. and Coulin, C. R. 2005. Requirements elicitation: A survey of techniques, approaches, and tools. In: *Engineering and Managing Software Requirements*. Heidelberg, Germany: Springer.

Zowghi, D. and Nurmuliani, N. 2002. A study of the impact of requirements volatility on software project performance. In: *9th Asia-Pacific Software Engineering Conference*. IEEE. :3–11.

Appendix A. RESEARCH ETHICAL CLEARANCE

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering,
Built Environment and Information Technology

1956 – 2016
60
years of
Engineering Education

Reference number: EBIT/38/2016 27 July 2016

Mr NP Scribante
Graduate School of Technology Management
University of Pretoria
Pretoria
0028

Dear Mr Scribante,

FACULTY COMMITTEE FOR RESEARCH ETHICS AND INTEGRITY

Your recent application to the EBIT Research Ethics Committee refers.

Approval is granted for the application with reference number that appears above.

1. This means that the research project entitled "An investigation into requirements volatility using system dynamics modeling" has been approved as submitted. It is important to note what approval implies. This is expanded on in the points that follow.
2. This approval does not imply that the researcher, student or lecturer is relieved of any accountability in terms of the Code of Ethics for Scholarly Activities of the University of Pretoria, or the Policy and Procedures for Responsible Research of the University of Pretoria. These documents are available on the website of the EBIT Research Ethics Committee.
3. If action is taken beyond the approved application, approval is withdrawn automatically.
4. According to the regulations, any relevant problem arising from the study or research methodology as well as any amendments or changes, must be brought to the attention of the EBIT Research Ethics Office.
5. The Committee must be notified on completion of the project.

The Committee wishes you every success with the research project.

Prof JJ Hanekom
Chair: Faculty Committee for Research Ethics and Integrity
FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

EBIT Research Ethics Committee
Room 15-6, Level 15, Engineering Building 1
University of Pretoria, Private Bag X20
Hatfield 0028, South Africa
Tel +27 (0)12 420 3736
Email mari.ferreira@up.ac.za

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie
Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

Appendix B. BLANK INFORMED CONSENT FORM

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Graduate School of Technology Management

An investigation into
requirements volatility using
system dynamics modeling

INFORMED CONSENT FORM FOR PARTICIPATING AS AN INFORMANT IN A RESEARCH PROJECT

1. **Title of research project**: An investigation into requirements volatility using system dynamics modeling.
2. **Researcher:** Naudé Scribante
3. I ………………………………………………………………………………………..
hereby voluntarily grant my permission for participation in the research.
4. The nature and objectives of the research and my role therein have been explained to me and I understand them.
5. I understand my right to choose whether to participate in the research project and that the information furnished will be handled confidentially. I am aware that the results of the investigation may be used for the purposes of publication.
6. I will be provided with a copy of this form after it has been completed and signed.
7. Signed: ____________________ Date: ______________
8. Witness: ____________________ Date: ______________
9. Researcher: ____________________ Date: ______________

Informed consent form
V0.0
04 July 2016

Appendix C. RESULTS OF SME INTERVIEWS

C.1 Subject Matter Expert 1

C.1.1 Project related – Project type

Interview notes: Results valid for a bespoke project with stakeholders as the requirements source.

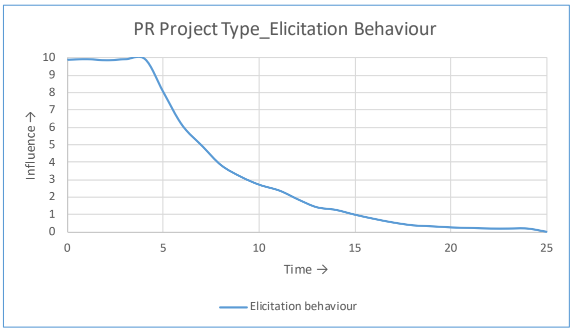


Figure C-1: SME-1: Project type – elicitation behaviour

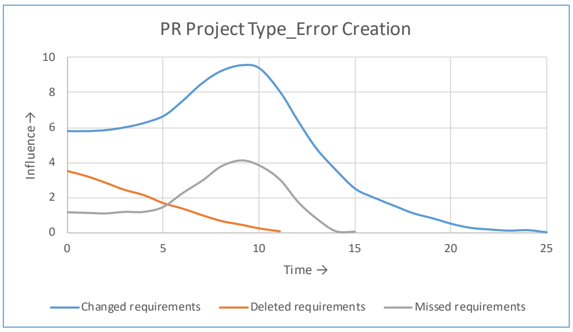


Figure C-2: SME-1: Project type – error creation

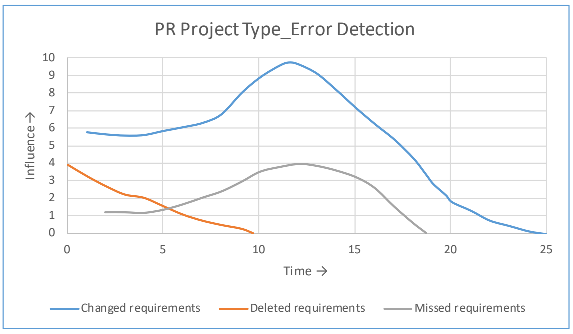


Figure C-3: SME-1: Project type – error detection

C.1.2 Project related – Project size

Interview notes: No response.

C.1.3 Project related – Project complexity

Interview notes: No response.

C.1.4 Project related – Number of stakeholders

Interview notes: No response.

C.1.5 Project related – Existing or legacy stakeholders

Interview notes: No response.

C.1.6 Project related – Project duration

Interview notes: Only provided a response to the error creation behaviour vs time. The behaviour indicated is for a case in which requirements change over the project duration. SME-1 commented that the error creation behaviour could potentially be due to the stakeholders taking a long time to change their minds.

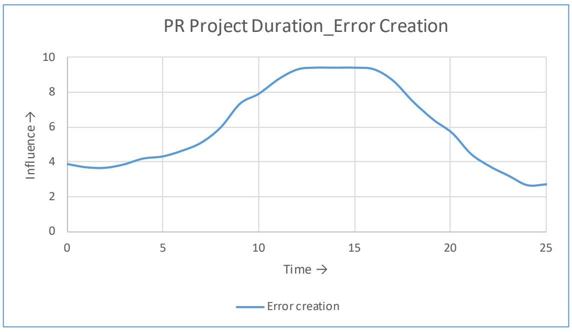


Figure C-4: SME-1: Project duration – error creation

C.1.7 Stakeholder characteristics – Domain knowledge

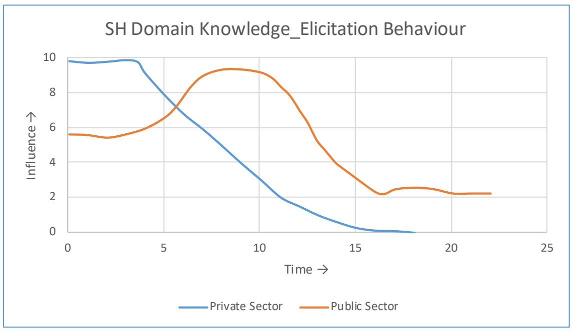


Figure C-5: SME-1: Stakeholder's domain knowledge – elicitation behaviour

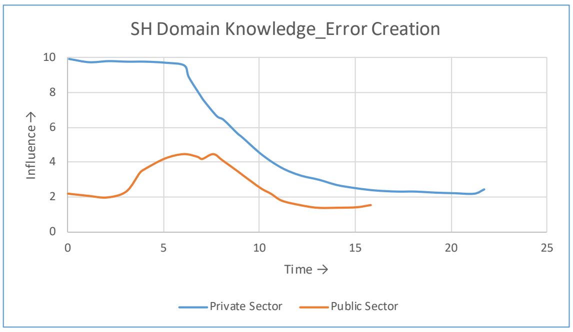


Figure C-6: SME-1: Stakeholder's domain knowledge – error creation

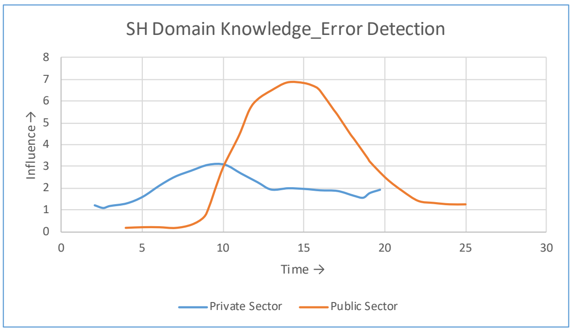


Figure C-7: SME-1: Stakeholder's domain knowledge – error detection

C.1.8 Stakeholder characteristics – Technical knowledge

Interview notes: Behaviour will be the same as for domain knowledge.

C.1.9 Stakeholder characteristics – Stakeholder misinformation

Interview notes: No response for the expected elicitation behaviour vs time.

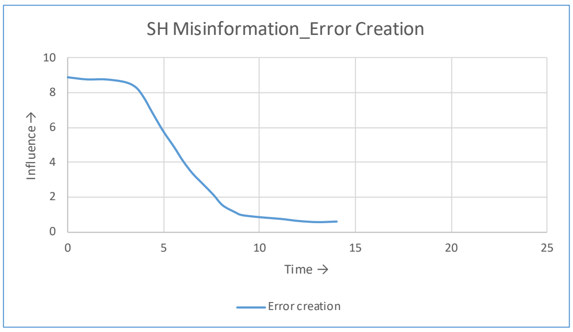


Figure C-8: SME-1: Stakeholder misinformation – error creation

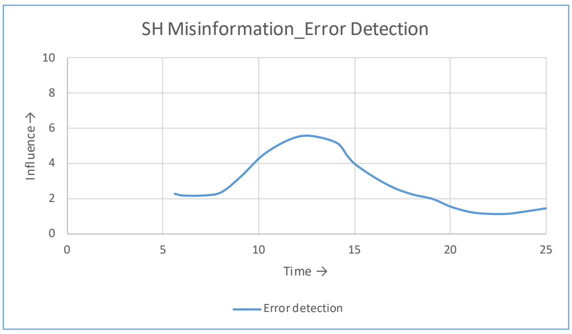


Figure C-9: SME-1: Stakeholder misinformation – error detection

C.1.10 Stakeholder characteristics – Stakeholder's nature

Interview notes: No response provided.

C.1.11 Stakeholder characteristics – Resistance to change

Interview notes: No response provided.

C.1.12 Stakeholder characteristics – Changes in technology

Interview notes: No response provided.

C.1.13 Requirements engineer characteristics – Domain knowledge

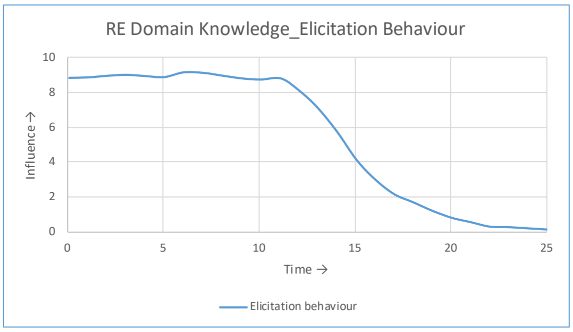


Figure C-10: SME-1: Requirements engineer's domain knowledge – elicitation behaviour

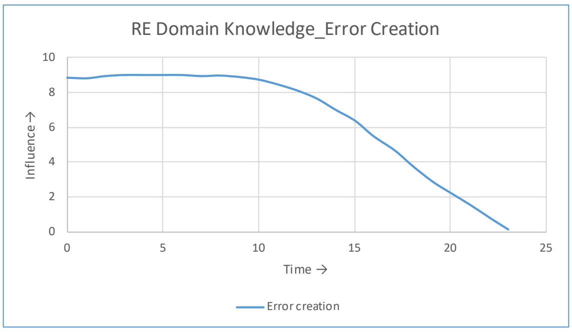


Figure C-11: SME-1: Requirements engineer's domain knowledge – error creation

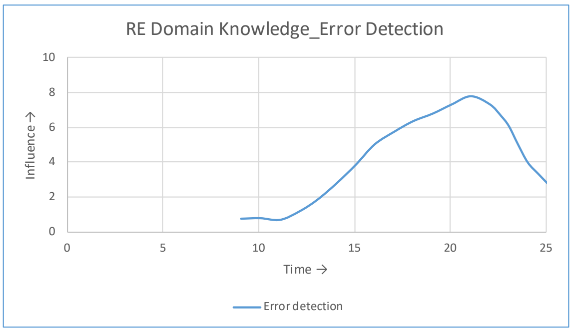


Figure C-12: SME-1: Requirements engineer's domain knowledge – error detection

C.1.14 Requirements engineer characteristics – Technical knowledge

Interview notes: No response provided.

C.1.15 Requirements engineer characteristics – Requirements engineer misinformation

Interview notes: No response provided.

C.1.16 Requirements engineer characteristics – Experience

Interview notes: No response provided.

C.1.17 Requirements engineer characteristics – Personality

Interview notes: No response provided.

C.1.18 Stakeholder and requirements engineer interaction – Social nature

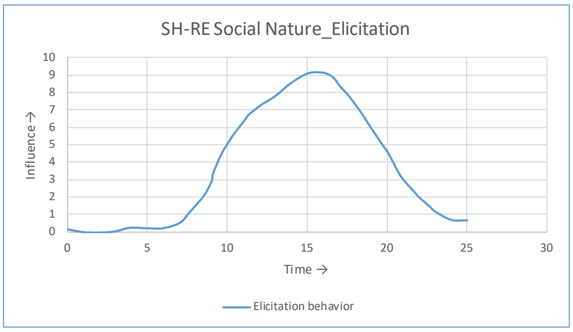


Figure C-13: SME-1: Stakeholder and requirements engineer interaction – social nature – elicitation behaviour

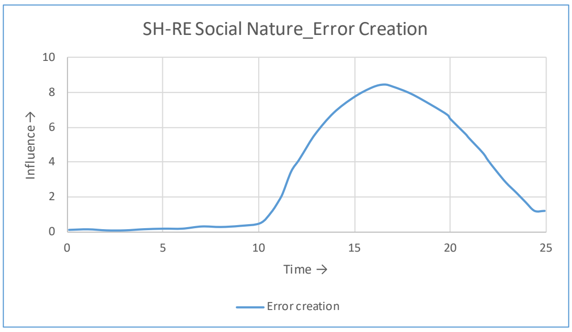


Figure C-14: SME-1: Stakeholder and requirements engineer interaction – social nature – error creation

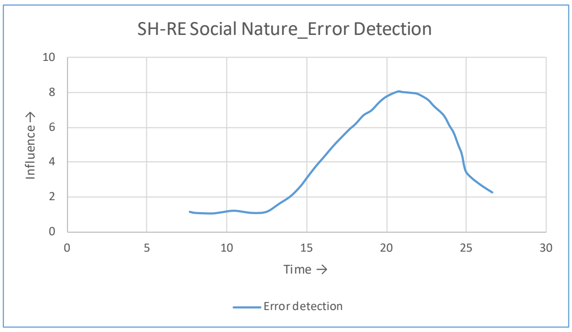


Figure C-15: SME-1: Stakeholder and requirements engineer interaction – social nature – error detection

C.1.19 Stakeholder and requirements engineer interaction – Communication

Interview notes: No response provided.

C.1.20 Stakeholder and requirements engineer interaction – Conflict

Interview notes: No response provided.

C.1.21 Stakeholder and requirements engineer interaction – Trust

Interview notes: No response provided.

C.2 Subject Matter Expert 2

C.2.1 Project related – Project type

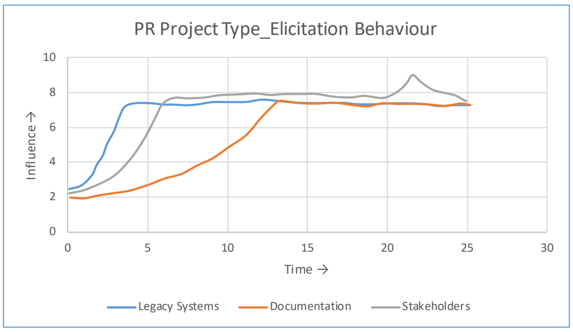


Figure C-16: SME-2: Project type – elicitation behaviour

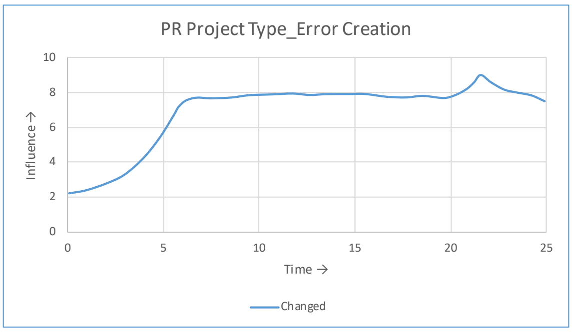


Figure C-17: SME-2: Project type – error creation

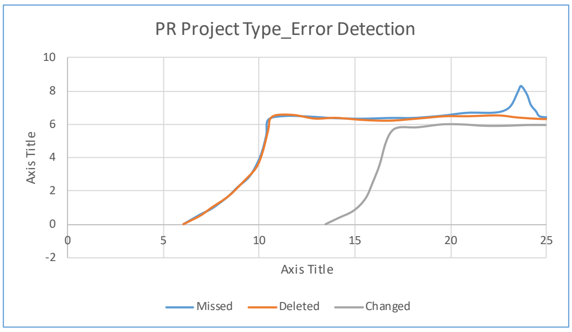


Figure C-18: SME-2: Project type – error detection

C.2.2 Project related – Project size

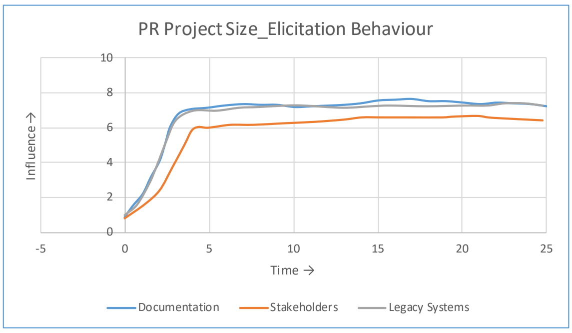


Figure C-19: SME-2: Project size – elicitation behaviour

Interview notes: Behaviour of error creation and error detection will be similar to that shown in Figure C-17: SME-2: Project type – error creation and Figure C-18: SME-2: Project type – error detection.

C.2.3 Project related – Project complexity

Interview notes: No response provided.

C.2.4 Project related – Number of stakeholders

Interview notes: No response provided. SME-2 works with dedicated focus groups.

C.2.5 Project related – Existing or legacy systems

Interview notes: Not applicable to this case study.

C.2.6 Project related – Project duration

Interview notes: Elicitation behaviour is the same is as shown in Figure C-16: SME-2: Project type – elicitation behaviour. If the project duration is very short, errors will only be detected later in the process.

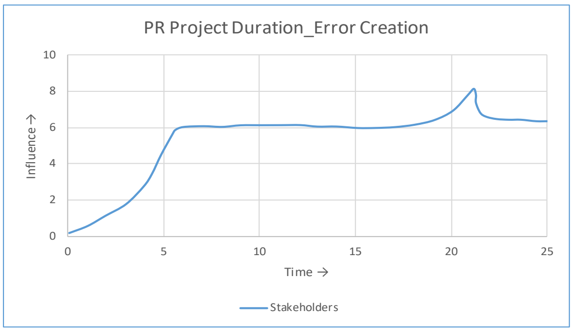


Figure C-20: SME-2: Project duration – error creation

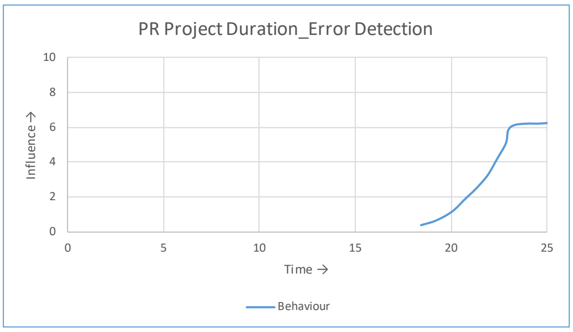


Figure C-21: SME-2: Project duration – error detection

C.2.7 Project related – Project budget

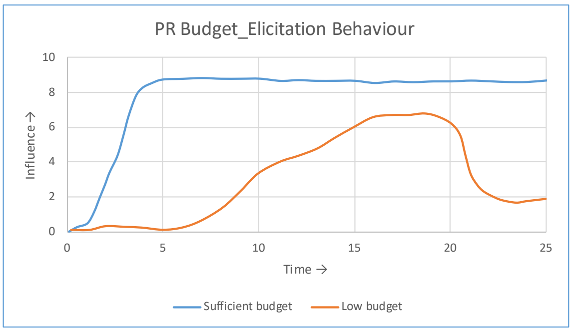


Figure C-22: SME-2: Project budget – elicitation behaviour

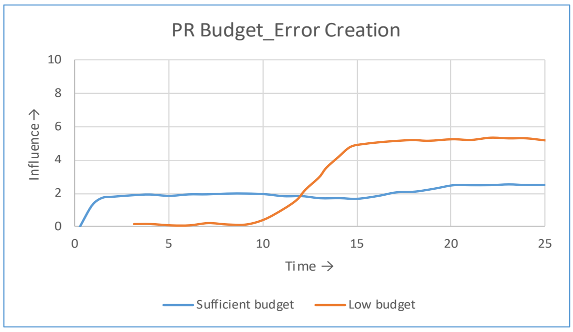


Figure C-23: SME-2: Project budget – error creation

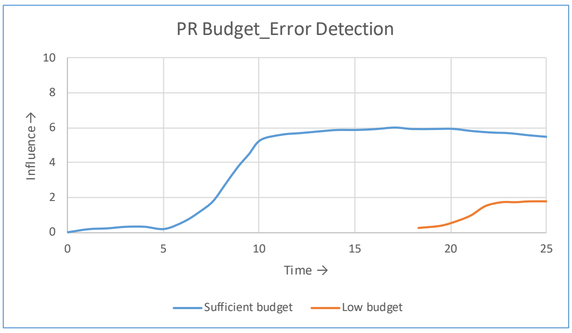


Figure C-24: SME-2: Project budget – error detection

C.2.8 Stakeholder characteristics – Domain knowledge

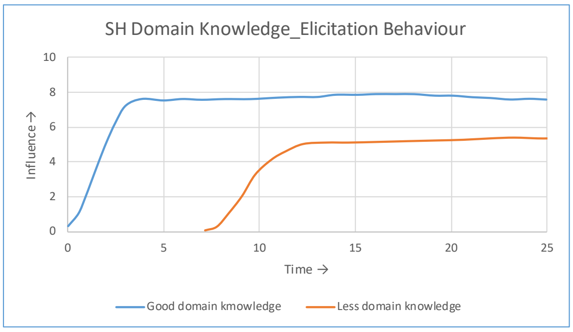


Figure C-25: SME-2: Domain knowledge – elicitation behaviour

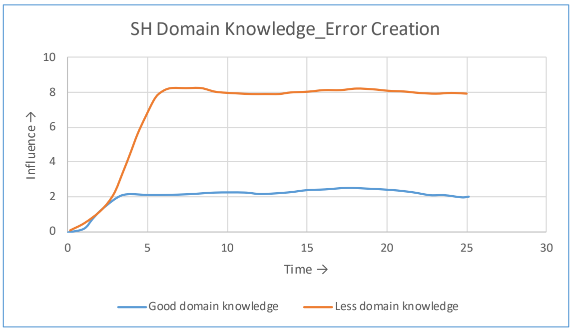


Figure C-26: SME-2: Domain knowledge – error creation

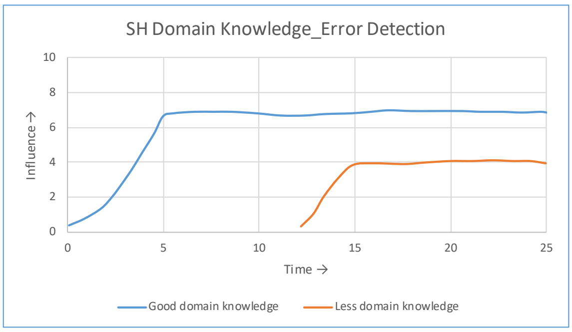


Figure C-27: SME-2: Domain knowledge – error detection

C.2.9 Stakeholder characteristics – Technical knowledge

Interview notes: No response provided

C.2.10 Stakeholder characteristics – Stakeholder misinformation

Interview notes: The requirements are elicited early in the process, but they are mostly incorrect or incomplete. The created errors are however only detected much later in the process.

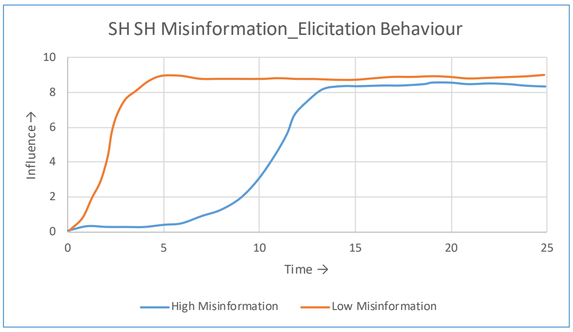


Figure C-28: SME-2: Stakeholder misinformation – elicitation characteristics

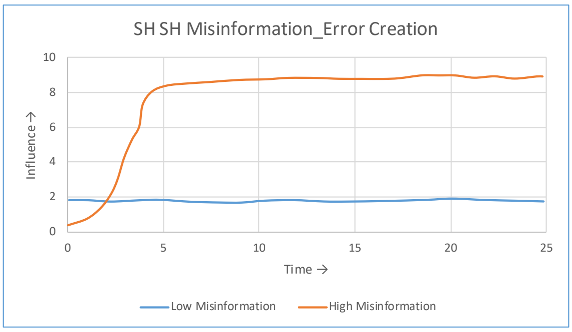


Figure C-29: SME-2: Stakeholder misinformation – error creation

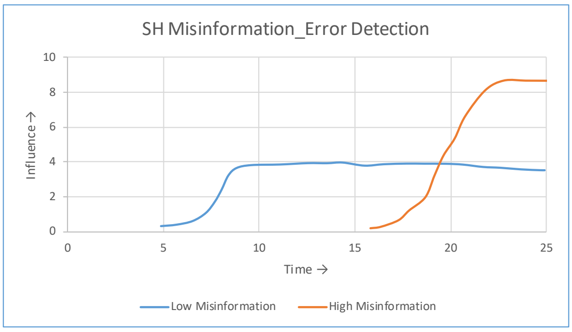


Figure C-30: SME-2: Stakeholder misinformation – error detection

C.2.11 Stakeholder characteristics – Stakeholder's nature

Interview notes: Similar behaviour as is shown in Figure C-28: SME-2: Stakeholder misinformation – elicitation characteristics, Figure C-29: SME-2: Stakeholder misinformation – error creation and Figure C-30: SME-2: Stakeholder misinformation – error detection.

C.2.12 Stakeholder characteristics – Resistance to change

Interview notes: (a) If stakeholders are only represented by company executives, the resistance to change at the other levels will be high. If all levels are considered, then the resistance to change will be less. (b) Resistance to change is directly related to the time that is available.

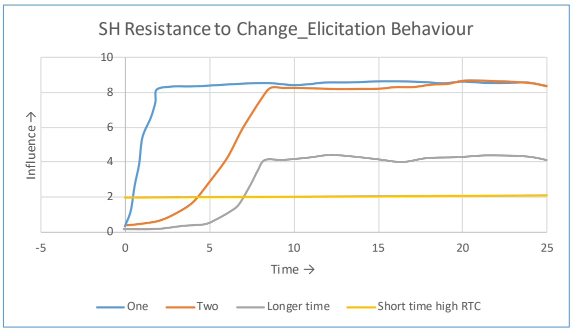


Figure C-31: SME-2: Resistance to change – elicitation behaviour

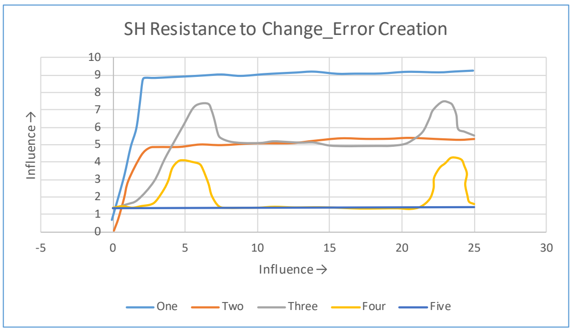


Figure C-32: SME-2: Resistance to change – error creation

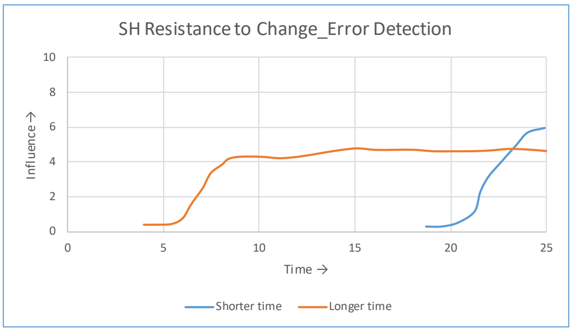


Figure C-33: SME-2: Resistance to change – error detection

C.2.13 Stakeholder characteristics – Changes in technology

Interview notes: No response provided.

C.2.14 Requirements engineer characteristics – Domain knowledge

Interview notes: No response provided.

C.2.15 Requirements engineer characteristics – Technical knowledge

Interview notes: No response provided.

C.2.16 Requirements engineer characteristics – Misinformation information

Interview notes: No response provided.

C.2.17 Requirements engineer characteristics – Experience

Interview notes: No response provided.

C.2.18 Requirements engineer characteristics – Personality

Interview notes: No response provided.

C.2.19 Stakeholder and requirements engineer interaction – Social nature

Interview notes: No response provided.

C.2.20 Stakeholder and requirements engineer interaction – Communication

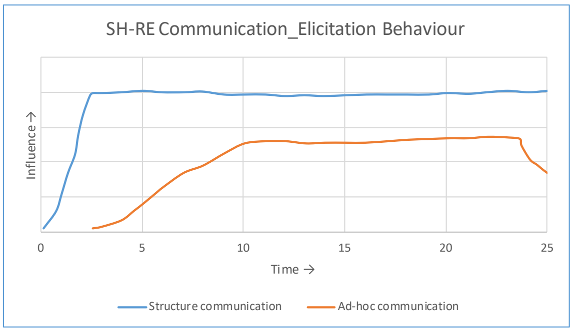


Figure C-34: SME-2: Stakeholder and requirements engineer communication – elicitation behaviour

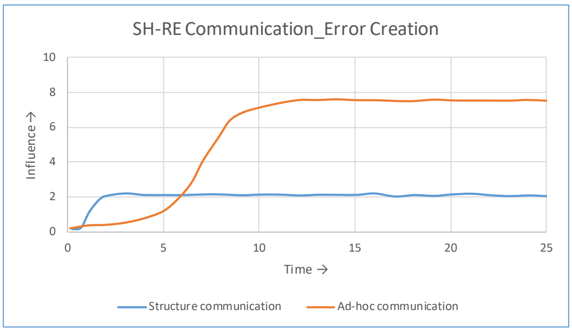


Figure C-35: SME-2: Stakeholder and requirements engineer communication – error creation

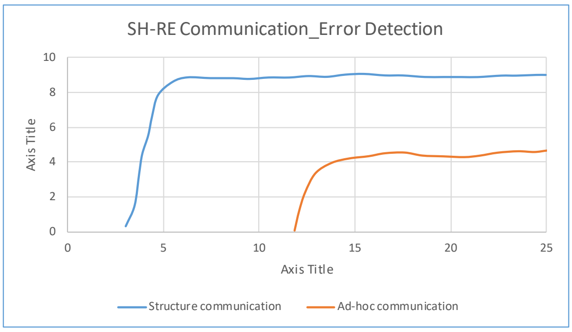


Figure C-36: SME-2: Stakeholder and requirements engineer communication – error detection

C.2.21 Stakeholder and requirements engineer interaction – Conflict

Interview notes: No response provided.

C.2.22 Stakeholder and requirements engineer interaction – Trust

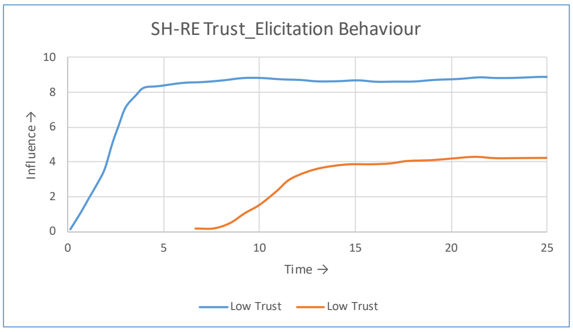


Figure C-37: SME-2: Stakeholder and requirements engineer trust – elicitation behaviour

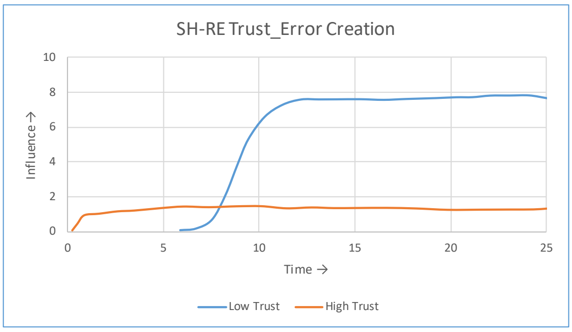


Figure C-38: SME-2: Stakeholder and requirements engineer trust – error creation

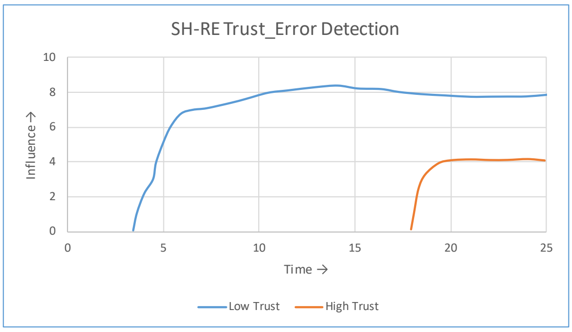


Figure C-39: SME-2: Stakeholder and requirements engineer trust – error detection

C.3 Subject Matter Expert 3

C.3.1 Project related – Project type

Interview notes: Stakeholders often depend on vendors to formalise requirements with COTS systems, only to realise later that there are requirements lacking or the implementation/integration complexity is more than expected.

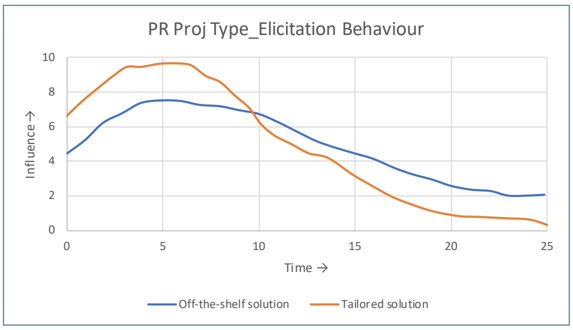


Figure C-40: SME-3: Project type – elicitation behaviour

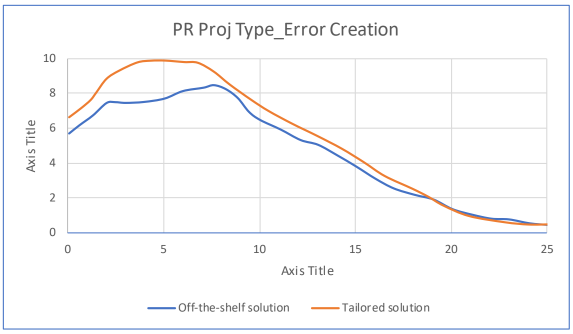


Figure C-41: SME-3: Project type – error creation

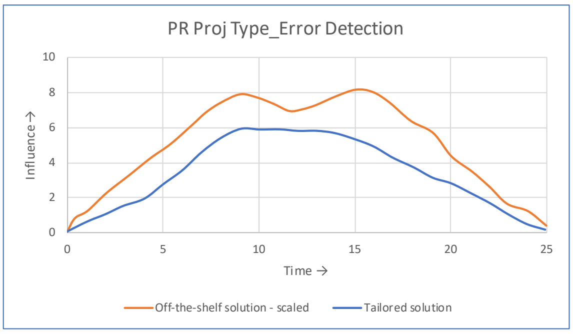


Figure C-42: SME-3: Project type – error detection

C.3.2 Project related – Project size

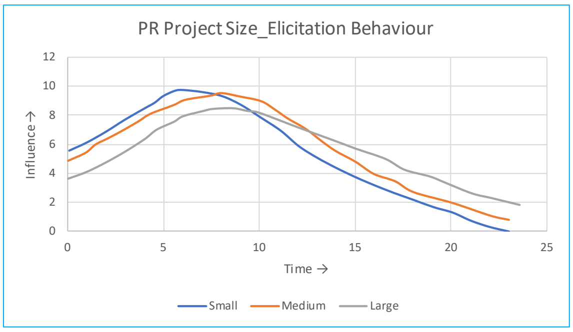


Figure C-43: SME-3: Project size – elicitation behaviour

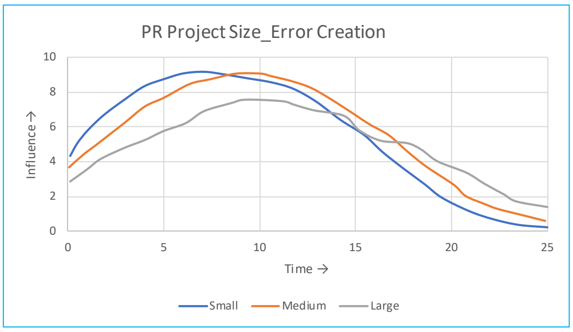


Figure C-44: SME-3: Project size – error creation

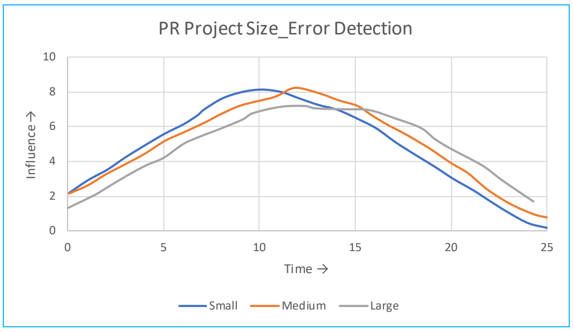


Figure C-45: SME-3: Project size – error detection

C.3.3 Project related – Project complexity

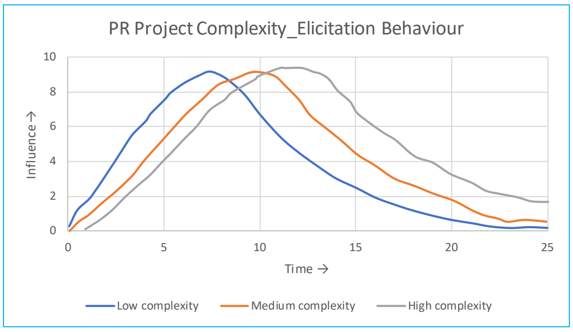


Figure C-46: SME-3: Project complexity – elicitation behaviour

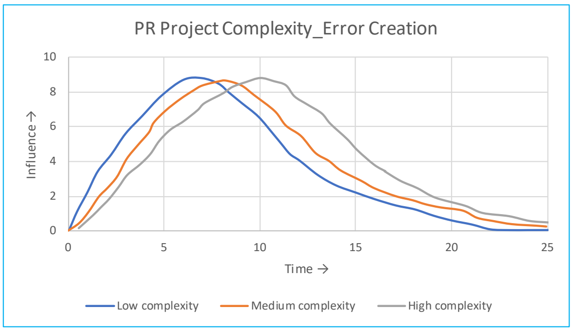


Figure C-47: SME-3: Project complexity – error creation

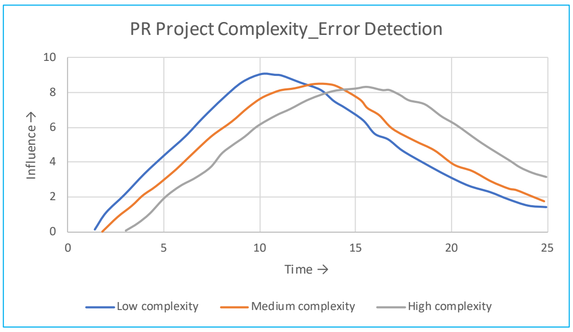


Figure C-48: SME-3: Project complexity – error detection

C.3.4 Project related – Number of stakeholders

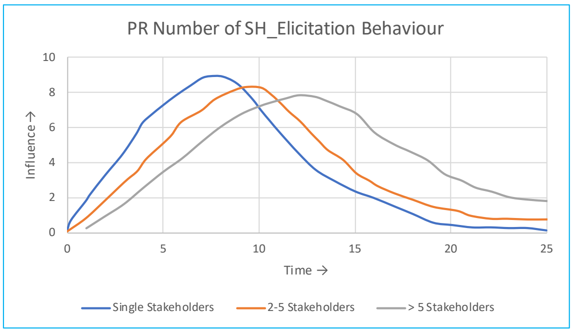


Figure C-49: SME-3: Number of stakeholders – elicitation behaviour

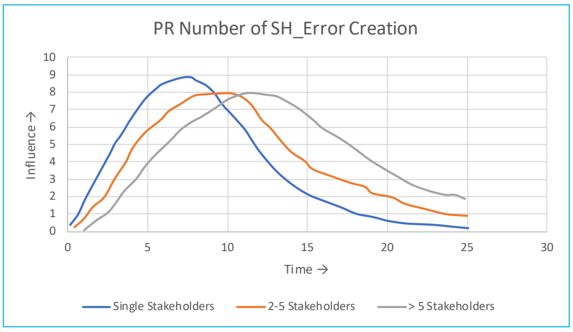


Figure C-50: SME-3: Number of stakeholders – error creation

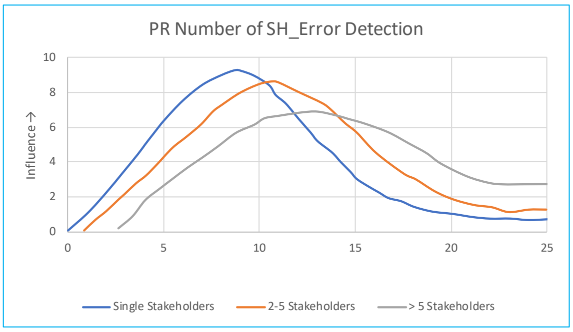


Figure C-51: SME-3: Number of stakeholders – error detection

C.3.5 Project related – Existing or legacy system requirements

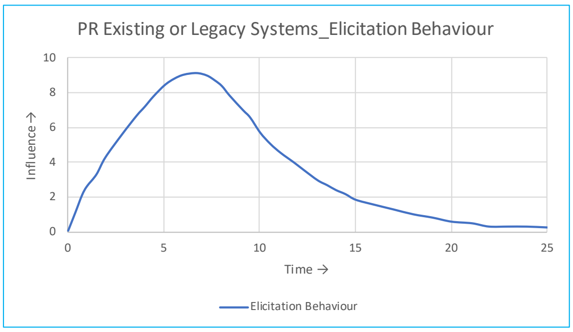


Figure C-52: SME-3: Existing or legacy system requirements – elicitation behaviour

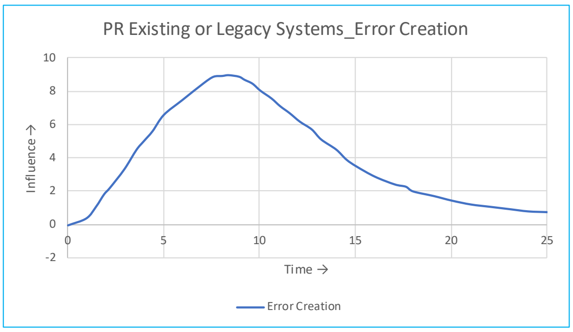


Figure C-53: SME-3: Existing or legacy system requirements – error creation

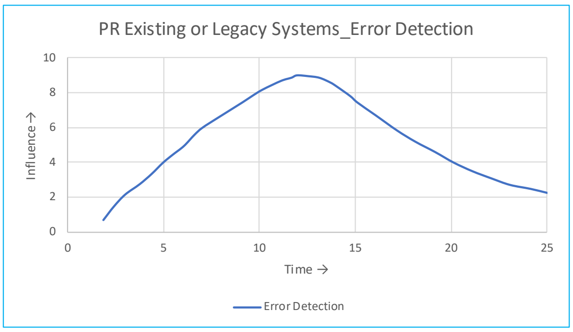


Figure C-54: SME-3: Existing or legacy system requirements – error detection

C.3.6 Project related – Project duration

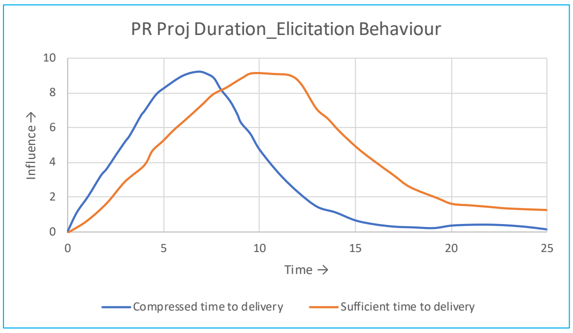


Figure C-55: SME-3: Project duration – elicitation behaviour

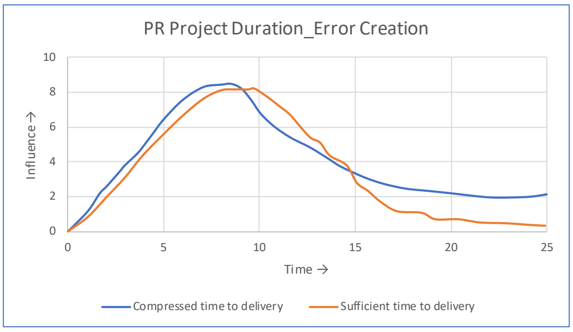


Figure C-56: SME-3: Project duration – error creation

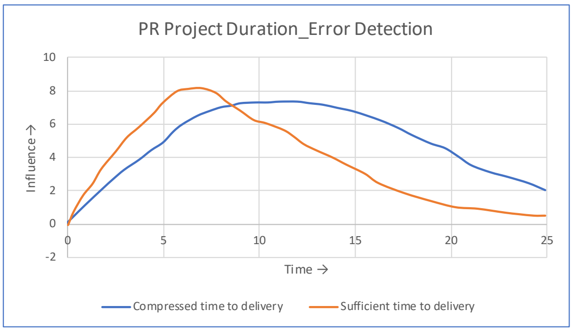


Figure C-57: SME-3: Project duration – error detection

C.3.7 Stakeholder characteristics – Domain knowledge

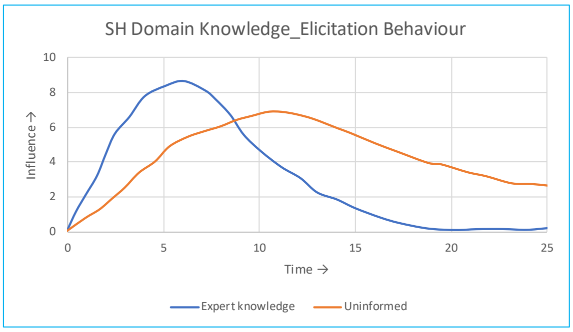


Figure C-58: SME-3: Domain knowledge – elicitation behaviour

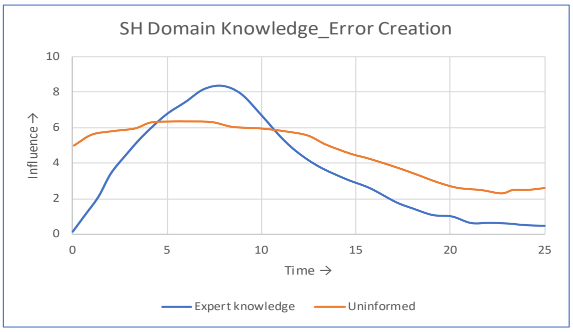


Figure C-59: SME-3: Domain knowledge – error creation

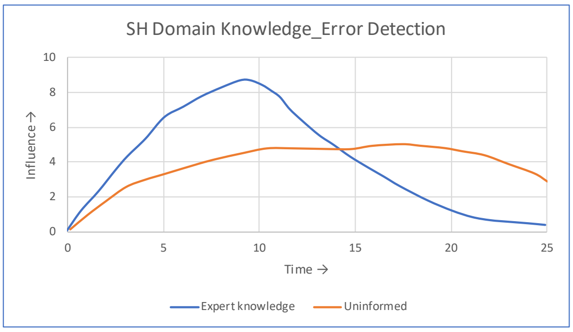


Figure C-60: SME-3: Domain knowledge – error detection

C.3.8 Stakeholder characteristic – Technical knowledge

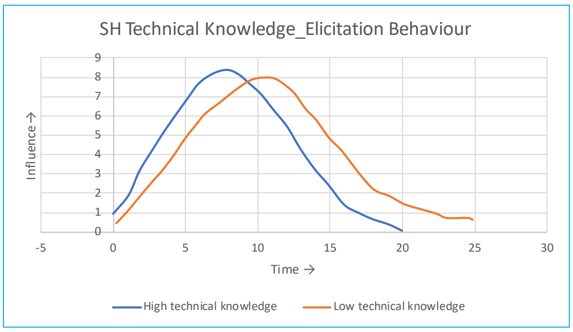


Figure C-61: SME-3: Technical knowledge – elicitation behaviour

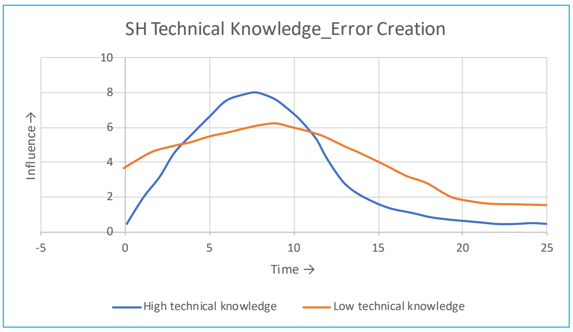


Figure C-62: SME-3: Technical knowledge – error creation

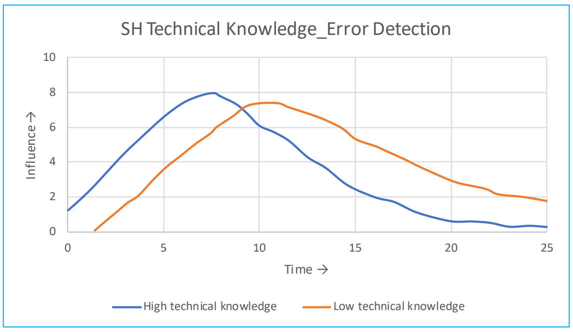


Figure C-63: SME-3: Technical knowledge – error detection

C.3.9 Stakeholder characteristic – Stakeholder misinformation

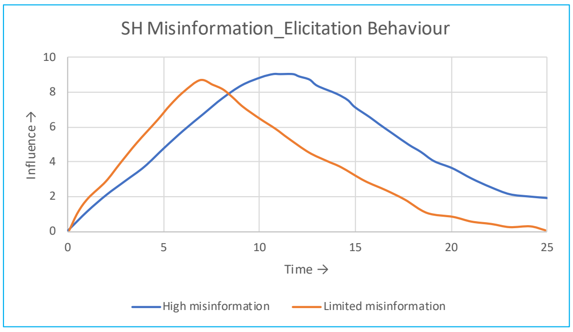


Figure C-64: SME-3: Stakeholder misinformation – elicitation behaviour

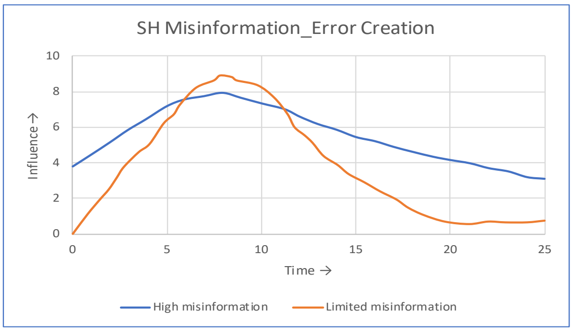


Figure C-65: SME-3: Stakeholder misinformation – error creation

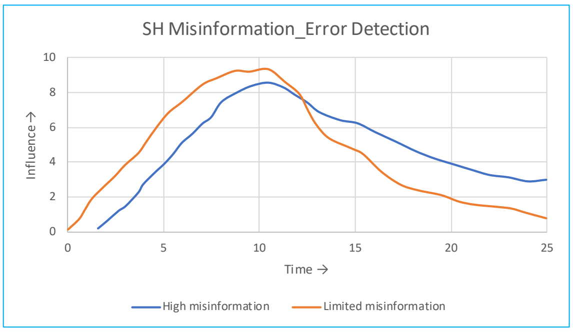


Figure C-66: SME-3: Stakeholder misinformation – error detection

C.3.10 Stakeholder characteristic – Stakeholder's nature

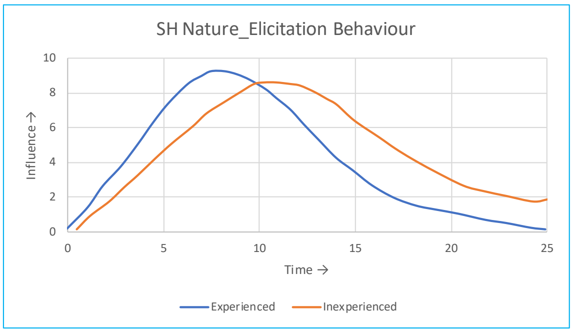


Figure C-67: SME-3: Stakeholder's nature – elicitation behaviour

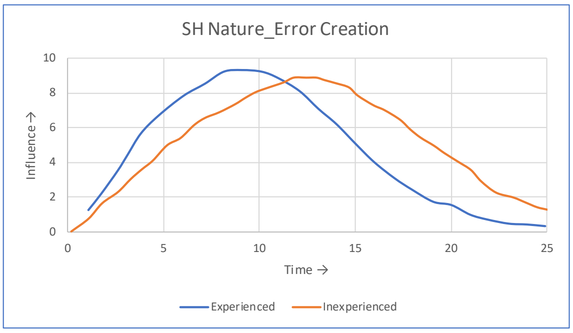


Figure C-68: SME-3: Stakeholder's nature – error creation

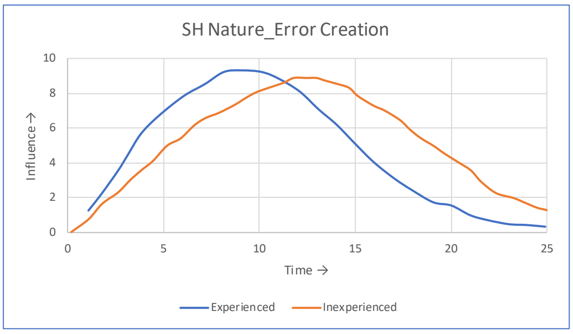


Figure C-69: SME-3: Stakeholder's nature – error detection

C.3.11 Stakeholder characteristic – Resistance to change

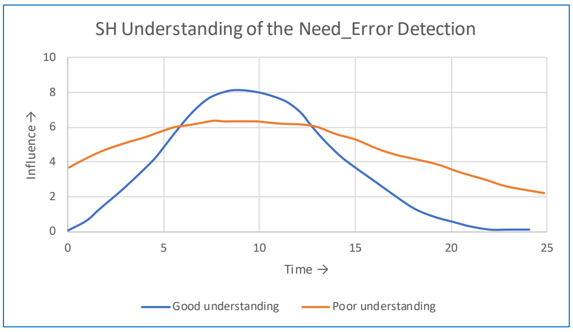


Figure C-70: SME-3: Resistance to change – elicitation behaviour

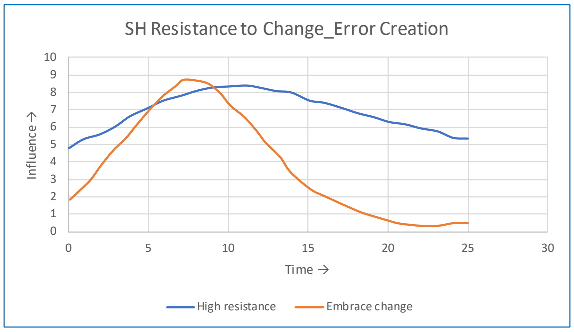


Figure C-71: SME-3: Resistance to change – error creation

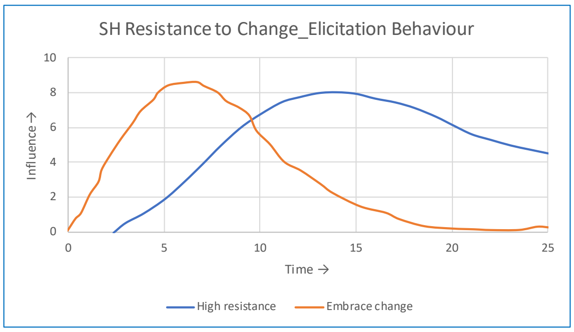


Figure C-72: SME-3: Resistance to change – Error detection

C.3.12 Stakeholder characteristic – Changes in technology

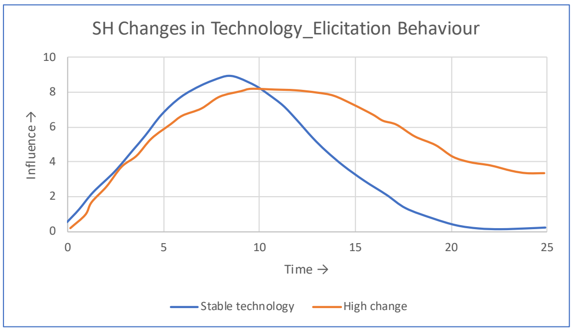


Figure C-73: SME-3: Changes in technology – elicitation behaviour

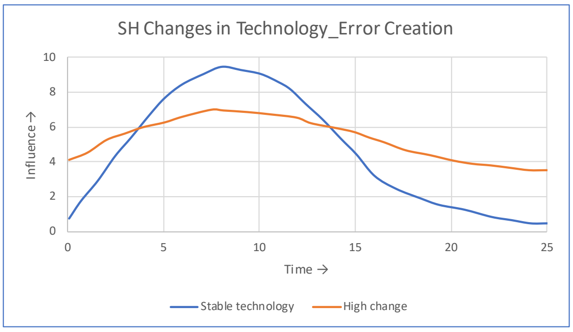


Figure C-74: SME-3: Changes in technology – error creation

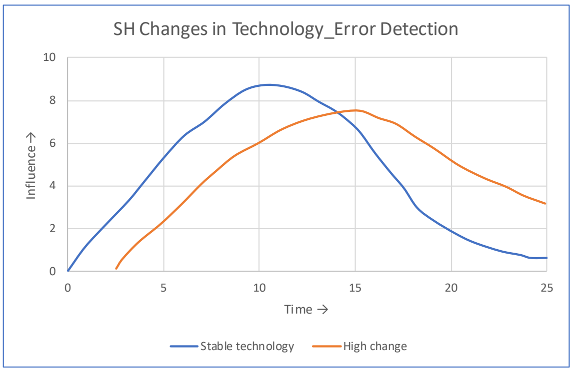


Figure C-75: SME-3: Changes in technology – error detection

C.3.13 Requirements engineer characteristics – Domain knowledge

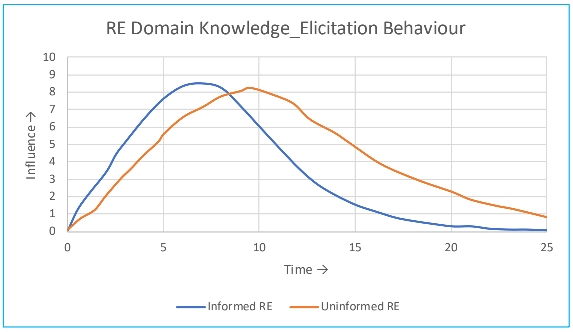


Figure C-76: SME-3: Domain knowledge – elicitation behaviour

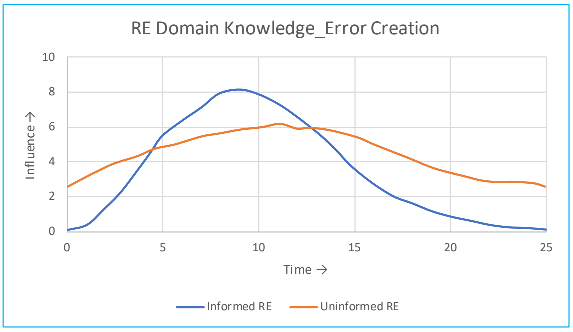


Figure C-77: SME-3: Domain knowledge – error creation

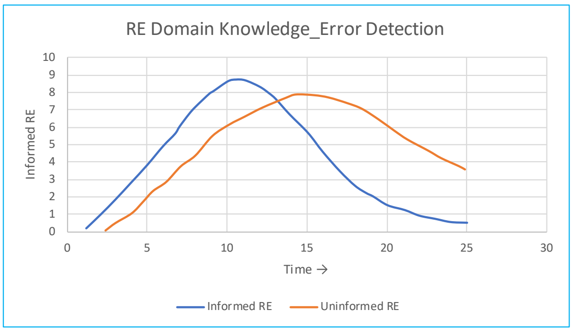


Figure C-78: SME-3: Domain knowledge – error detection

C.3.14 Requirements engineer characteristics – Technical knowledge

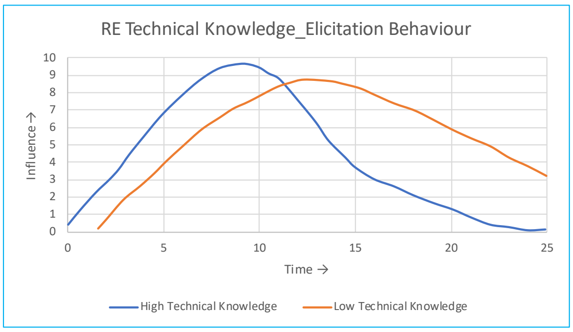


Figure C-79: SME-3: Technical knowledge – elicitation behaviour

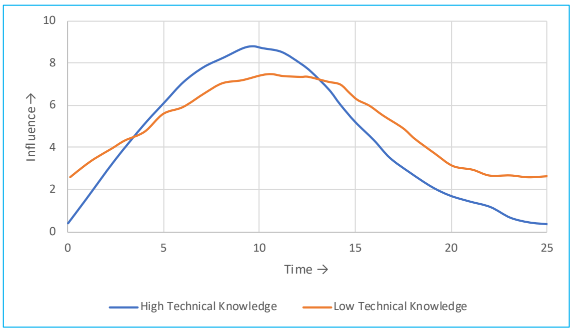


Figure C-80: SME-3: Technical knowledge – error creation

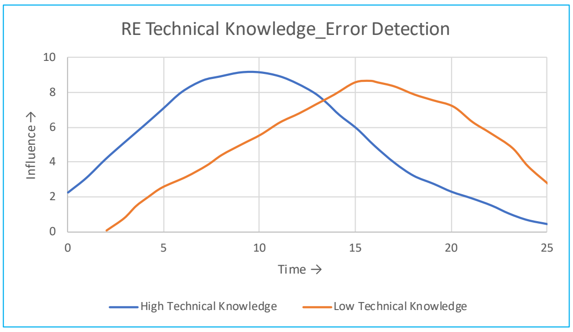


Figure C-81: SME-3: Technical knowledge – error detection

C.3.15 Requirements engineer characteristics – Requirements engineer misinformation

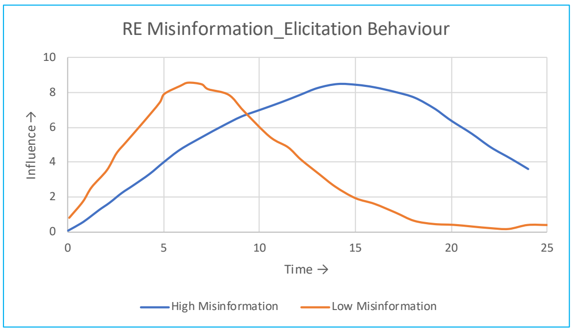


Figure C-82: SME-3: Requirements engineer misinformation – elicitation behaviour

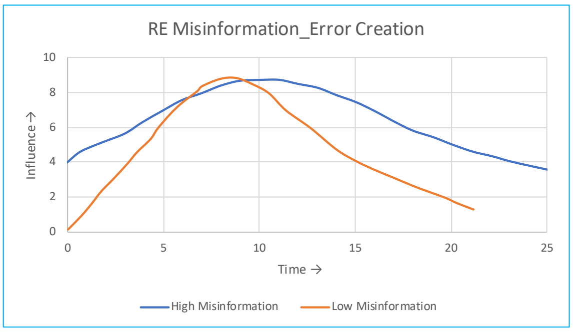


Figure C-83: SME-3: Requirements engineer misinformation – error creation

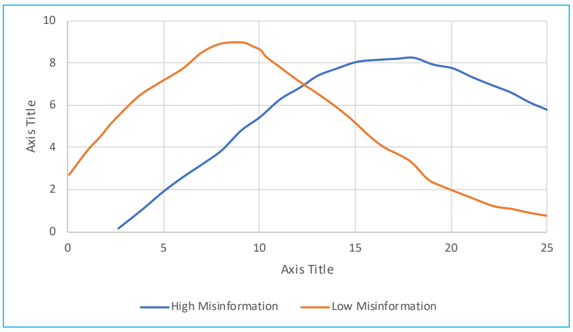


Figure C-84: SME-3: Requirements engineer misinformation – error detection

C.3.16 Requirements engineer characteristics – Experience

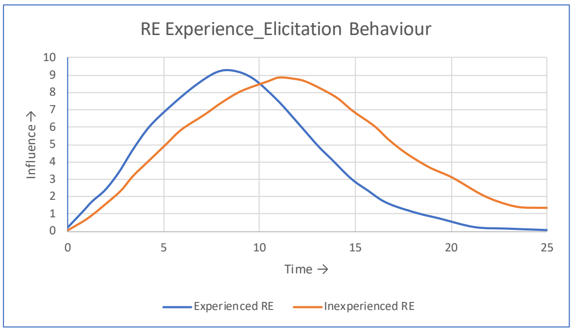


Figure C-85: SME-3: Experience – elicitation behaviour

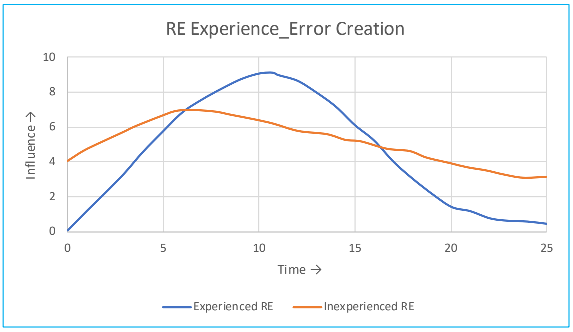


Figure C-86: SME-3: Experience – error creation

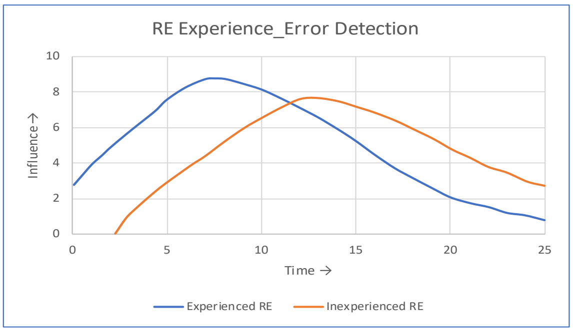


Figure C-87: SME-3: Experience – error detection

C.3.17 Requirements engineer characteristics – Personality

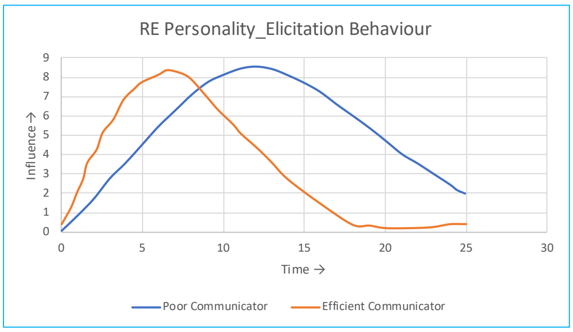


Figure C-88: SME-3: Personality – elicitation behaviour

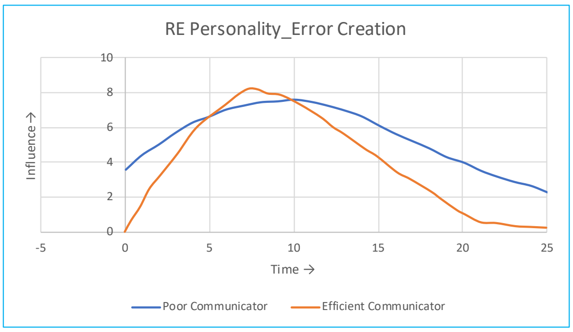


Figure C-89: SME-3: Personality – error creation

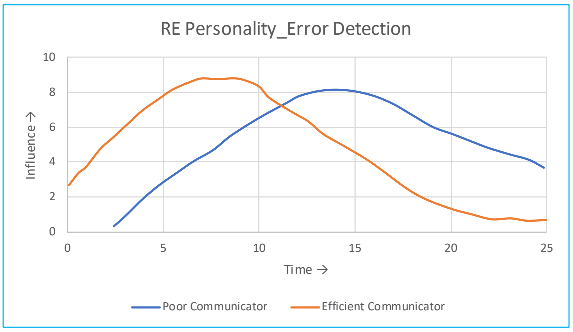


Figure C-90: SME-3: Personality – error detection

C.3.18 Stakeholder and requirements engineer interaction – Social nature

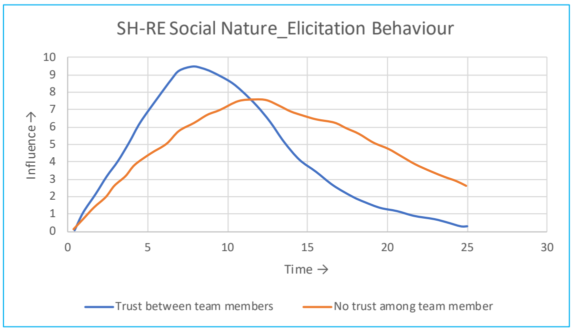


Figure C-91: SME-3: Stakeholder and requirements engineer social nature – elicitation behaviour

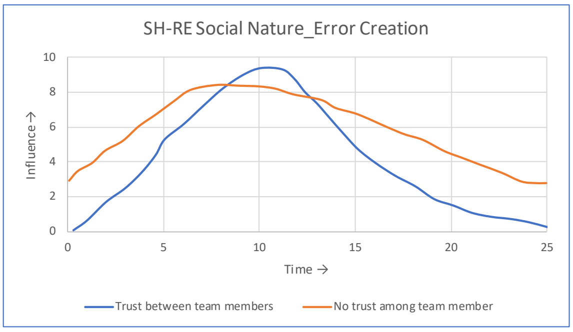


Figure C-92: SME-3: Stakeholder and requirements engineer social nature – error creation

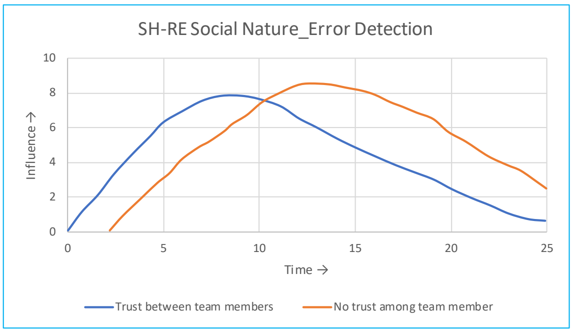


Figure C-93: SME-3: Stakeholder and requirements engineer social nature – error detection

C.3.19 Stakeholder and requirements engineer interaction – Communication

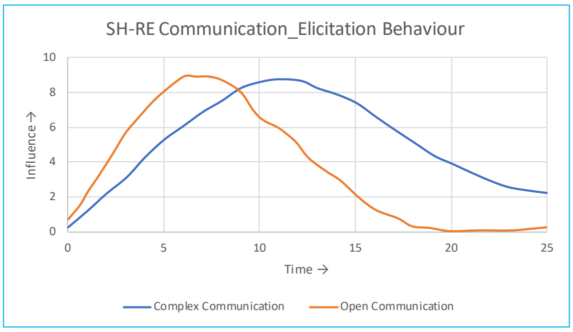


Figure C-94: SME-3: Stakeholder and requirements engineer communication – elicitation behaviour

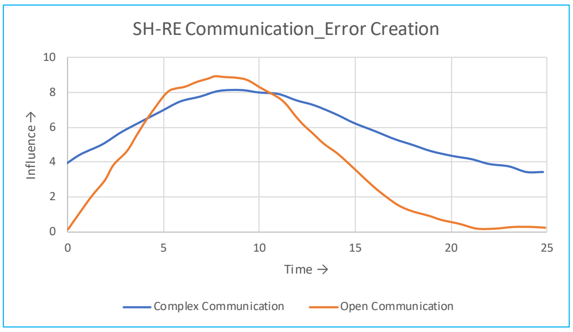


Figure C-95: SME-3: Stakeholder and requirements engineer communication – error creation

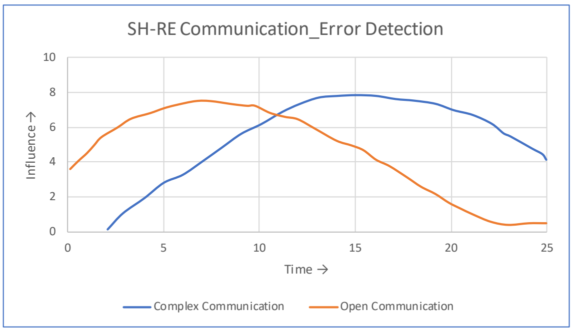


Figure C-96: SME-3: Stakeholder and requirements engineer communication – error detection

C.3.20 Stakeholder and requirements engineer interaction – Conflict

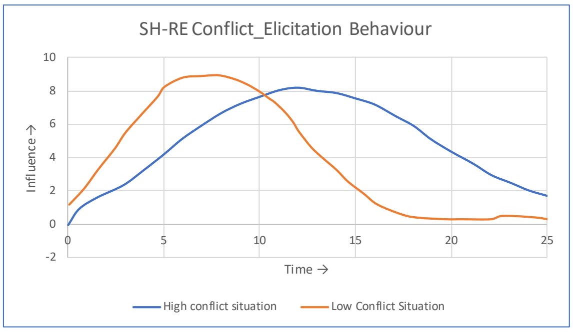


Figure C-97: SME-3: Stakeholder and requirements engineer conflict – elicitation behaviour

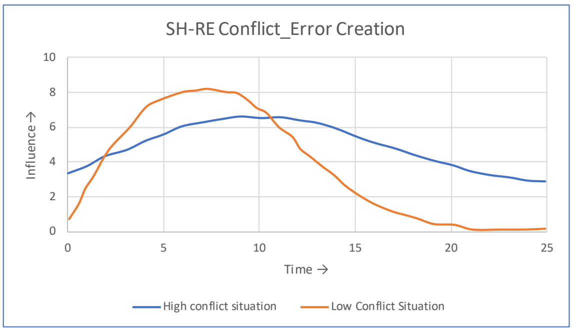


Figure C-98: SME-3: Stakeholder and requirements engineer conflict – error creation

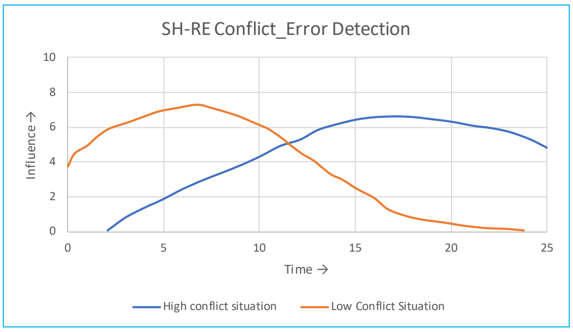


Figure C-99: SME-3: Stakeholder and requirements engineer conflict – error detection

C.3.21 Stakeholder and requirements engineer interaction – Trust

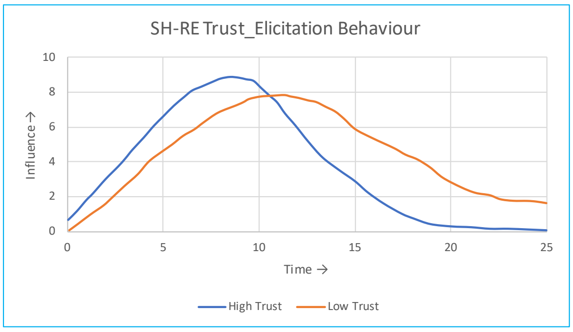


Figure C-100: SME-3: Stakeholder and requirements engineer trust – elicitation behaviour

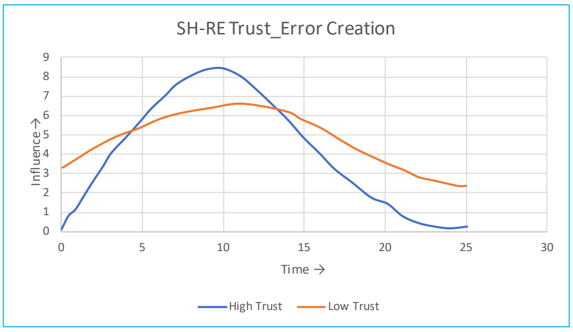


Figure C-101: SME-3 – Stakeholder and requirements engineer trust – error creation

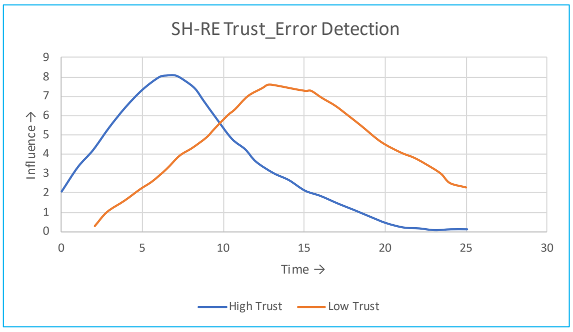


Figure C-102: SME-3: Stakeholder and requirements engineer trust – error detection

C.3.22 Agile/iterative development vs waterfall development

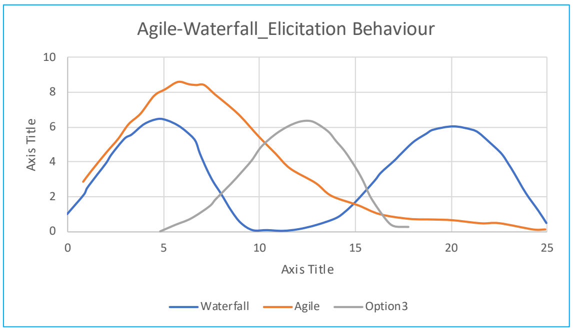


Figure C-103: SME-3: Agile – Waterfall elicitation behaviour

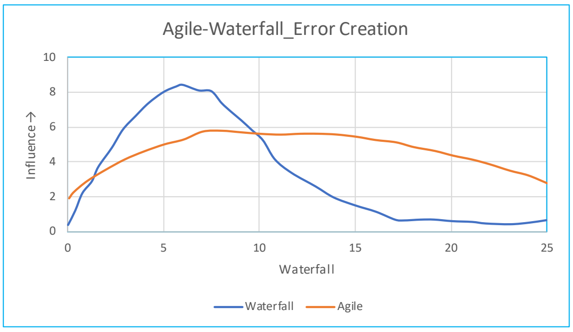


Figure C-104: SME-3: Agile – Waterfall error creation

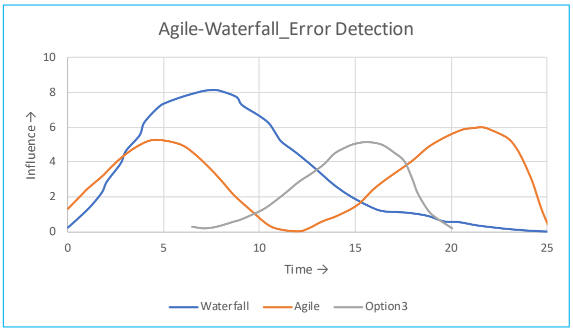


Figure C-105: SME-3: Agile – Waterfall error detection

C.4 Subject Matter Expert 4

C.4.1 Project related – Project type

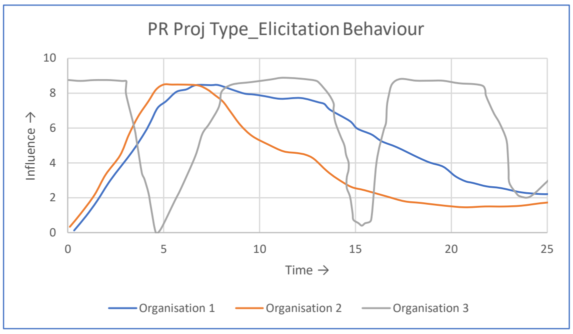


Figure C-106: SME-4: Project type – elicitation behaviour

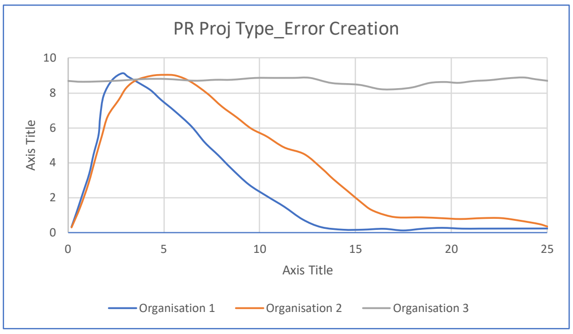


Figure C-107: SME-4: Project type – error detection

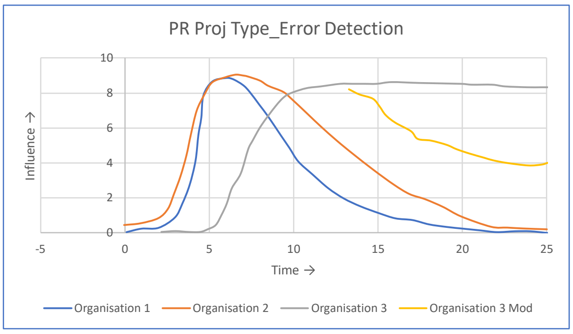


Figure C-108: SME-4: Project type – error creation

C.4.2 Project related – Project size

Interview notes: No response.

C.4.3 Project related – Project complexity

Interview notes: No response.

C.4.4 Project related – Number of stakeholders

Interview notes: No response.

C.4.5 Project related – Existing or legacy stakeholders

Interview notes: No response.

C.4.6 Project related – Project duration

Interview notes: Error were only detected during design reviews such as the preliminary design review and the critical design review.

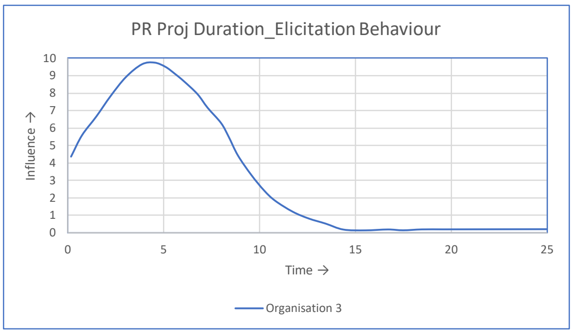


Figure C-109: SME-4: Project duration – elicitation behaviour

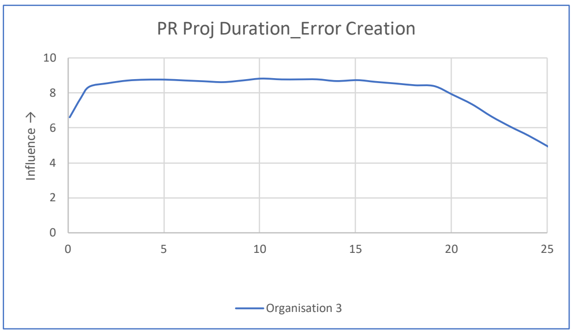


Figure C-110: SME-4: Project duration – error creation

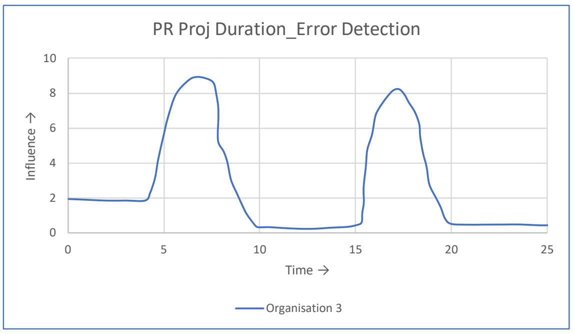


Figure C-111: SME-4: Project duration – error detection

C.4.7 Stakeholder characteristics – Domain knowledge

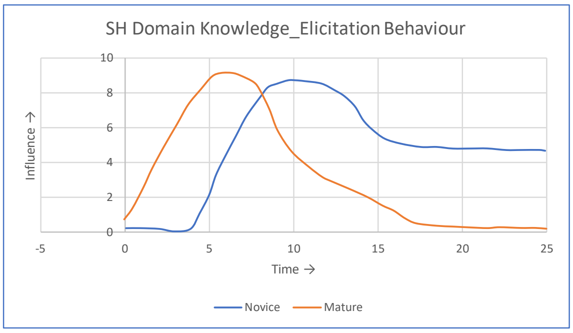


Figure C-112: SME-4: Stakeholder domain knowledge – elicitation behaviour

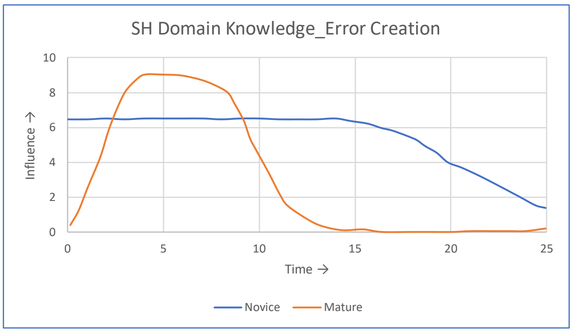


Figure C-113: SME-4: Stakeholder domain knowledge – error creation

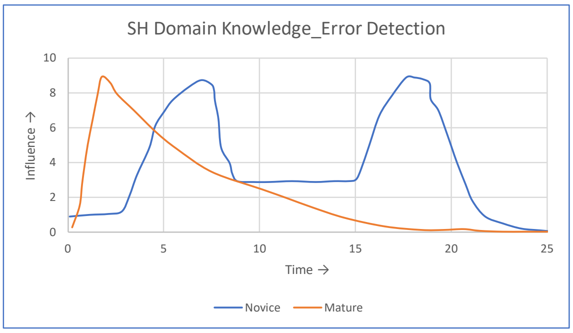


Figure C-114: SME-4: Stakeholder domain knowledge – error detection

C.4.8 Stakeholder characteristics – Technical knowledge

Interview notes: Behaviour will be the same as domain knowledge.

C.4.9 Stakeholder characteristics – Stakeholder misinformation

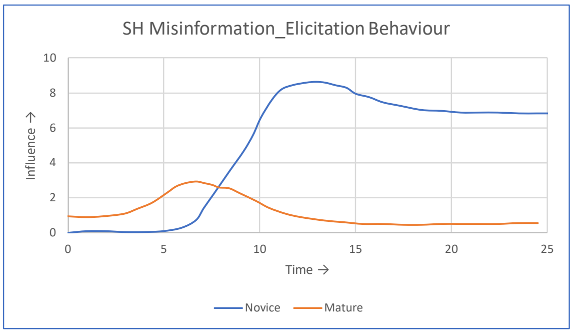


Figure C-115: SME-4: Stakeholder misinformation – elicitation behaviour

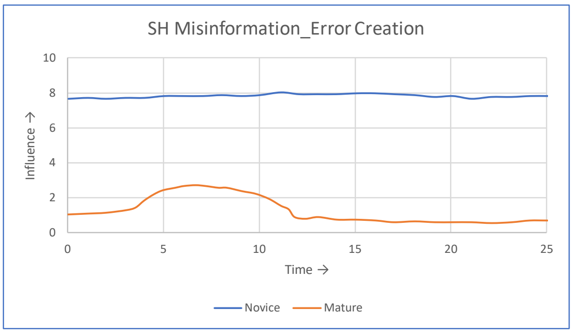


Figure C-116: SME-4: Stakeholder misinformation – error creation

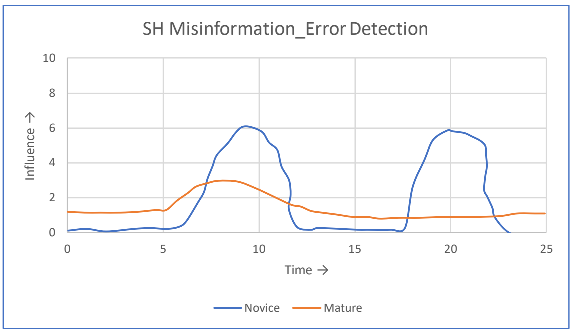


Figure C-117: SME-4: Stakeholder misinformation – error detection

C.4.10 Stakeholder characteristics – Stakeholder's nature

Interview notes: A mature stakeholder will not be influenced as much by individual personalities as a novice stakeholder.

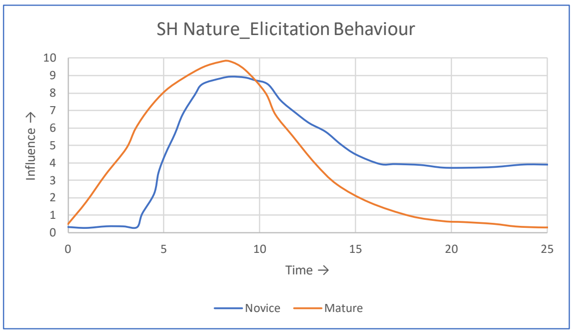


Figure C-118: SME-4: Stakeholder's nature – elicitation behaviour

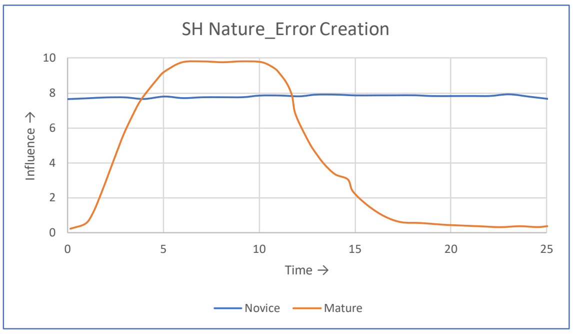


Figure C-119: SME-4: Stakeholder's nature – error creation

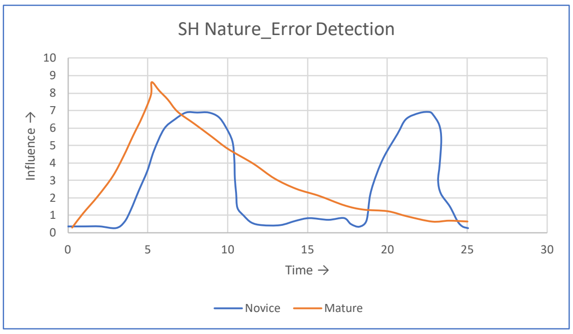


Figure C-120: SME-4: Stakeholder's nature – error detection

C.4.11 Stakeholder characteristics – Resistance to change

Interview notes: Similar behaviour as for the stakeholder misinformation case.

C.4.12 Stakeholder characteristics – Changes in technology

Interview notes: Only applicable during the early phases up to the PDR.

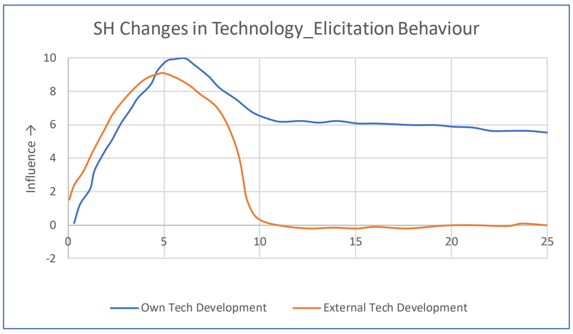


Figure C-121: SME-4: Stakeholder changes in technology – elicitation behaviour

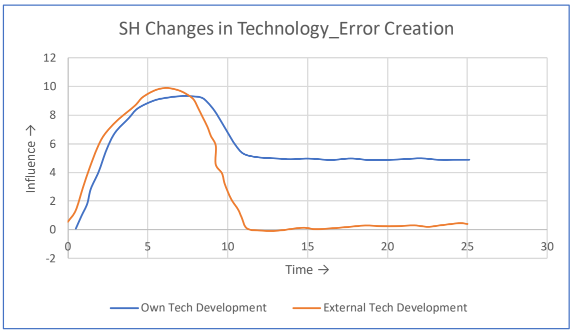


Figure C-122: SME-4: Stakeholder changes in technology – error creation

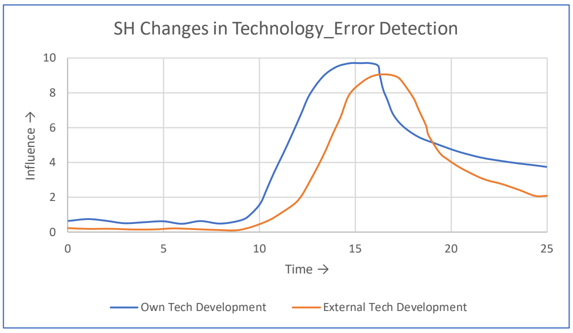


Figure C-123: SME-4: Stakeholder changes in technology – Error detection

C.4.13 Requirements engineer characteristics – Domain knowledge

Interview notes: Same behaviour as for the stakeholder's domain knowledge case.

C.4.14 Requirements engineer characteristics – Technical knowledge

Interview notes: No response provided.

C.4.15 Requirements engineer characteristics – Requirements engineer misinformation

Interview notes: No response provided.

C.4.16 Requirements engineer characteristics – Experience

Interview notes: Relates significantly to the maturity of the organisation.

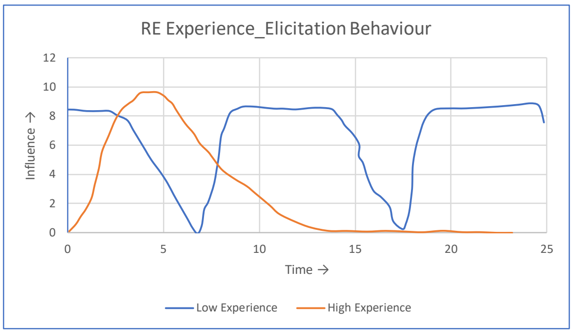


Figure C-124: SME-4: Requirements engineer's experience – elicitation behaviour

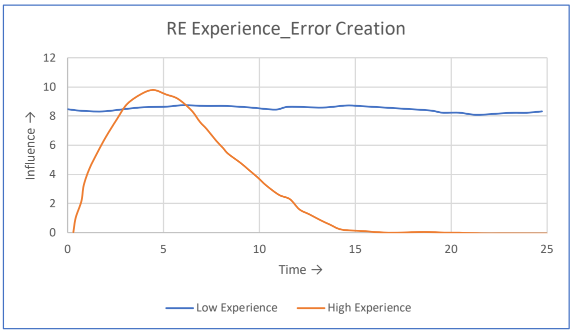


Figure C-125: SME-4: Requirements engineer' experience – error creation

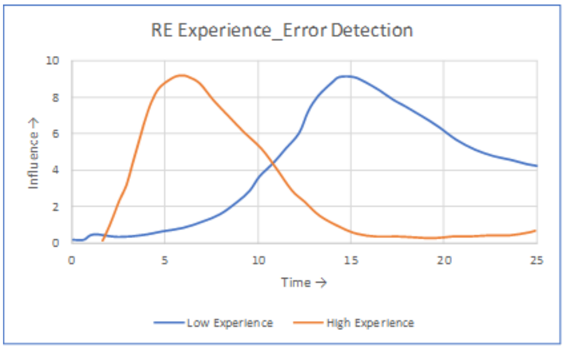


Figure C-126: SME-4: Requirements engineer's experience – Error detection

C.4.17 Requirements engineer characteristics – Personality

Interview notes: Like the behaviour of domain knowledge, technical knowledge and experience. Is also linked to the competence of the requirements engineer, the support provided by the higher organisation to this person, and the ability of the requirements engineer to take calculated risks.

C.4.18 Stakeholder and requirements engineer interaction – Social nature

Interview notes: Trust between the stakeholders and the requirements engineer remains a central theme.

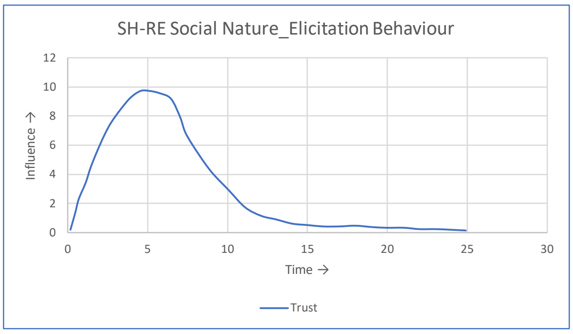


Figure C-127: SME-4: Stakeholder and requirements engineer interaction – social nature – elicitation behaviour

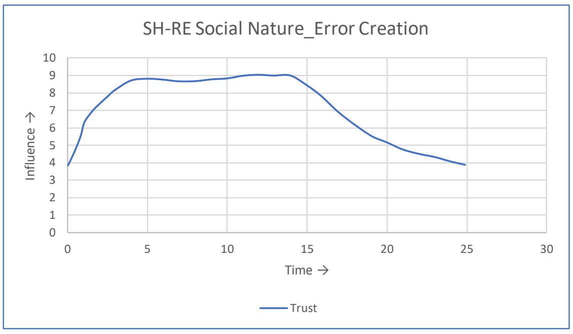


Figure C-128: SME-4: Stakeholder and requirements engineer interaction – social nature – error creation

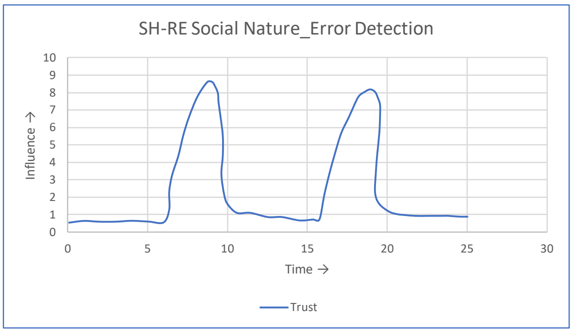


Figure C-129: SME-4: Stakeholder and requirements engineer interaction – social nature – error detection

C.4.19 Stakeholder and requirements engineer interaction – Communication

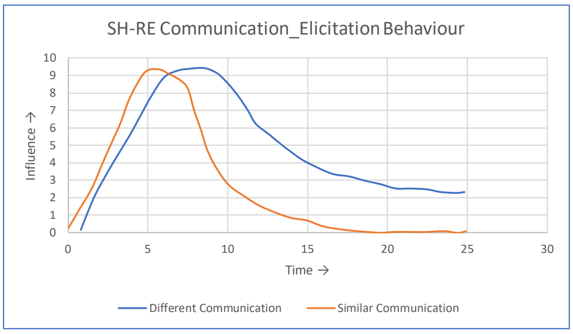


Figure C-130: SME-4: Stakeholder and requirements engineer interaction – communication – elicitation behaviour

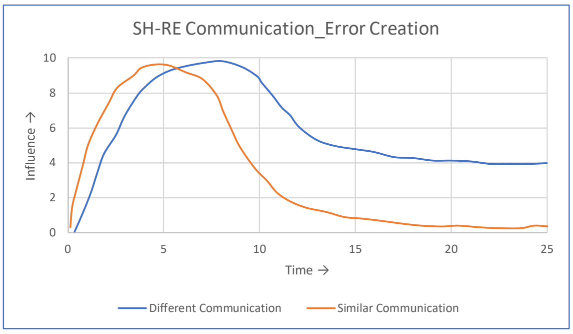


Figure C-131: SME-4: Stakeholder and requirements engineer interaction – communication – error creation

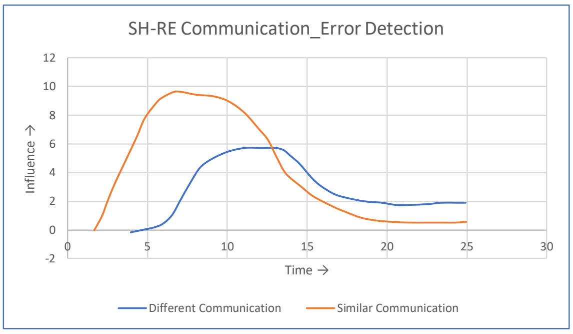


Figure C-132: SME-4: Stakeholder and requirements engineer interaction – communication – error detection

C.4.20 Stakeholder and requirements engineer interaction – Conflict

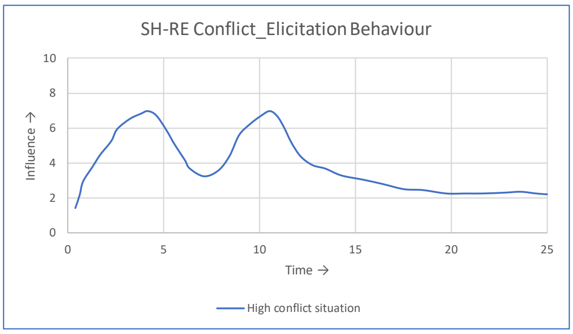


Figure C-133: SME-4: Stakeholder and requirements engineer interaction – conflict – elicitation behaviour

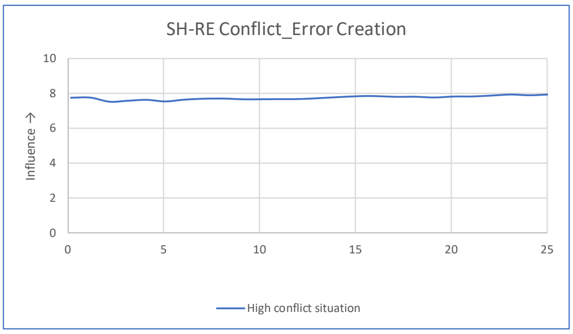


Figure C-134: SME-4: Stakeholder and requirements engineer interaction – conflict – error creation

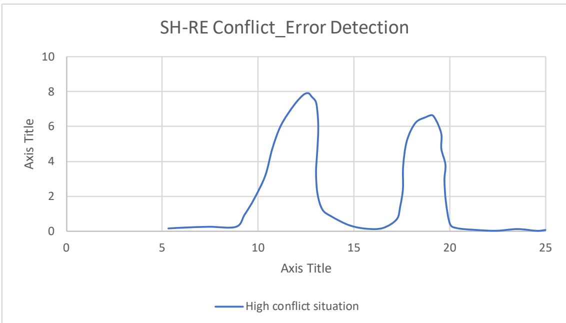


Figure C-135: SME-4: Stakeholder and requirements engineer interaction – conflict – error detection

C.4.21 Stakeholder and requirements engineer interaction – Trust

Interview notes: Similar behaviour as for the requirements engineer's personality.