Faculty of Engineering and the Built Environment

Department of Industrial Engineering

An investigation into the effectiveness of industrialising Software Quality Assurance (SQA) in small software businesses

Meena Patel

Student Number – 22176189

Submitted in fulfilment of the requirements of the

Masters of Engineering degree

August 2023

Supervisor: Dr O A Olanrewaju _ **Date:** 13/01/2024

DECLARATION

I hereby declare that this submission is my own and to the best of my knowledge, it neither contains material previously published nor written by another person, nor material that to a major extent has been accepted for the award of any other degree at Durban University of Technology or any other educational institution. I also declare that the intellectual content of this thesis is a product of my work.

_____

Signature

30 August 2023

_____

Date

ABSTRACT

Poor software quality has far-reaching consequences, including financial losses and potential risks to life. A significant proportion (92%) of software development is undertaken by very small to medium-sized software businesses. However, resource constraints often limit their ability to implement quality standards and methods that could enhance their product quality. Industrialization, a management concept for cost-effective production, offers potential solutions. While prior research has explored the industrialization of software development and quality assurance in large companies, a gap exists regarding its application in small software businesses. This study aims to investigate the effectiveness of industrializing the software quality assurance (SQA) process within small software businesses.

Adopting a qualitative approach with an interpretivist philosophy, inductive methodology, and exploratory strategy, this study employed semi-structured interviews to collect data. Participants included software developers and quality assurance representatives from five small software development businesses located in KwaZulu Natal and Western Cape provinces, South Africa. The theoretical foundation of this study draws upon Total Quality Management (TQM) principles and dimensions of industrialization (modularization, standardization, specialization, automation, and continuous improvement), as developed by experts during the early and mid-20th century.

The data collected was analysed using thematic analysis with support from Nvivo software. Results revealed that despite lacking established quality assurance strategies and process evaluation mechanisms, small software businesses effectively meet customer needs. Moreover, these businesses exhibit potential for successful industrialization, particularly focusing on the testing processes. Implications of these findings include the positive impact on the capacity of small businesses to sustain themselves within local economies.

ACKNOWLEDGEMENTS

I acknowledge and thank my supervisor Dr Oludolapo Olanrewaju, for his support and guidance throughout the course of this study.

GLOSSARY

Term, Acronym, Abbreviation	Description/Definition
CMMI	Capability Maturity Model Integration
CoSQ	Cost of Software Quality
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International standards organisation
SDLC	Software Development Lifecycle
SME	Small and Medium enterprise
SQA	Software Quality Assurance
TD	Technical Debt
TQM	Total Quality Management

TABLE OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 CHAPTER 1 – INTRODUCTION

"Software is a great combination between artistry and engineering." ~ Bill Gates

1.1 Introduction

Software applications are a product of the software development processes. These applications vary widely in size and complexity. They range from enterprise resource management software for global corporates to online shopping applications on your mobile phone. The scope of software applications extends from usage in the space industry, to general to-do list applications. It is deployed on supercomputers, to brain implants and to mobile phones. Software coding encompasses high level languages to machine level coding. The use of software has become an indispensable part of our lives with a wide range of uses.

This industry has been considered a relatively young industry and has always searched for new ways to make the software product less prone to error. Software failures can range from minor inconveniences to catastrophic loss of finances or life. Generally, organisations do not pay careful attention to the costs of poor quality and instead focus on time to market as their primary business performance measure. In a US report Krasner (2021) said that the total cost of poor quality software in the United States for 2020 was reported to be $2.08 Trillion. The report concludes that continuing to ignore good quality in software products will cost organisations much more than assuring good quality into their products.

As with all industries, there is a need to produce high quality products quickly, flexibly and cost efficiently. Industrial engineering has provided a means to achieve these goals. A guide presented by (BITKOM 2010) recommends an industrial approach to software development based on standardisation, automation, re-use, specialization and continuous improvement. Greenfield and Short (2003) took the industrialisation of software development a step further by presenting a conceptual framework for developing software using industrialised concepts. It is, however, difficult to measure the degree of implementation of industrialised software development worldwide. It is not the intention of this study to prove that software development is currently in an industrialised state but instead, to look at the Software Quality Assurance (SQA) process and its potential for

industrialisation. This will provide small businesses with a method to produce software of high quality using cost effective techniques.

Software Quality Assurance (SQA) is a process or set of activities, which provides a means to monitor, evaluate and improve the quality of the software process to reduce failures and inconveniences in the software product. Assuring good software quality is a challenge for software organisations globally. At the best of times, fully resourced organisations find managing quality difficult. For software businesses that are small, the challenge is far greater. Small software businesses are economically vulnerable as they are driven by cash flow and depend on project profits (O'Connor 2019).

Benefits of cost reduction and improved quality of the product will contribute towards sustainability of small software businesses. Small software businesses are crucial to the economy in any country and reducing their costs for producing good quality software product ultimately help sustain the software development industry.

1.1.1 Rationale of the study

According to (Laporte 2015) about 92% of the software development is carried out by very small to medium software development. Small organisations are limited in terms of human and financial resources and often cannot afford the cost of execution of international ISO and IEC standards for software quality.

One definition of Software Quality Assurance by (Agarwal *et al.* 2007), explains that SQA is the “planned and systematic approach to the evaluation of the quality, and adherence to software product standards, processes and procedures”. Currently, many software development organisations have defined processes to develop software. They use methodologies such as Waterfall, Scrum and Agile to develop their products. In their study, (Vijayasarathy and Butler, 2016), show that the most frequently used methodology is the Waterfall model, followed by the Agile Unified process and then Scrum and test-driven methodologies.

The impact of the costs related to developing poor quality software are often not measured (especially in small businesses) and not included in the budget to develop software products. Hence organisations are not able to determine and remove the root causes of these “poor quality” costs and they lead to recurring costs. Managing SQA will

help these organisations manage and prevent these expenses. Industrialisation provides appropriate techniques and tools to management, which allow for cost effective production.

Limited research has been done to determine a path for industrialising the software development process itself. In recent years research, as documented in the BITKOM report, BITKOM (2010) describes a process for industrialisation of software development. But very little has been done with the quality engineering aspect of the software product and processes other than the work done by (Wieczorek, Vos and Bons 2014). These authors have, however, focused on large and global software development enterprises. There is, therefore, a huge chasm between industrialising the software development process and the industrialising the SQA process in small to medium size enterprises, and therefore a need to address this topic. This study explores the effectiveness of industrialising the software quality assurance process in small software organisations.

1.2 Research Background

According to (Del Águila, Sagrado and Cañadas 2020), we entered the second age of our software engineering evolution, from 1993 to 2018, where software engineering started to be considered an industrial business (the first age being between 1950 to 1992 where hardware resources drove software). Development processes started to emerge. Frameworks such as CMMI (Capability Maturity Model Integration), adapted from the manufacturing field, were developed for the software development process (Chaudhary and Chopra 2017).

Up to the 1960s information system or software development was considered a craft and developers were craftsman in the software business (Booch 2018). Soon after that the software development lifecycle (SDLC) was introduced, to bring structure and rigour to the development process. What started off as an approach to computation has evolved into a discipline of engineering (Jacobson, Spence and Seidewitz 2016). This discipline however, is not as mature as other disciplines (Al-Sarayreh, Meridji and Abran 2021). To emphasise this Baptista and Salles (2012) indicate that the software engineering field is still experiencing challenges where other engineering disciplines may have overcome.

In the industrial sector, very early in the 20th century, quality assurance was implemented as an inspection activity, and it was solely the responsibility of the artisan who made the product. When mass production started, the artisan's sole inspection responsibility diminished, and standardisation and inspections became the rule on the factory floor (Weckenmann, Akkasoglu and Werner 2015). Statistical process control (SPC) was then introduced to economically control higher manufacturing outputs. Quality assurance techniques for software also needed to evolve but the evolution was not as straightforward as with hardware products. Standards were developed to provide a guide for improved software quality product and processes. To achieve and maintain a desired level of quality to any product or process a financial measure is necessary. Small software businesses found difficulty in measuring their costs of assuring quality. Hence the guide ISO 29110, (International Standards Organization (ISO) 2016) for small software businesses was developed.

According to (O'Connor and Laporte 2014), small software businesses make up almost 95% of businesses worldwide. Larrucea et al. (2016) asserts that small and very small organisations struggle with implementing standards such as those presented by ISO, CMMI and IEC and are reluctant to follow any quality improvement processes (Sultana, Syeed and Fatema 2020). Numerous tools and models to help small businesses have been proposed by (Sowunmi et al. 2016; Janes, Lenarduzzi and Stan 2017; Yahaya et al. 2017). However, these authors also recommend future study be undertaken to find more effective sustainable models.

The notion that the software quality assurance (SQA) process can be industrialised was raised by (Wieczorek, Vos and Bons 2014) to reduce the overall software production costs. Industrialisation is associated with the dimensions of modularisation, standardisation, specialisation, automation and continuous improvement (BITKOM 2010). These dimensions are used by the model presented by Wieczorek, Vos and Bons (2014). The limitation of this industrialised software quality model, however, is that large and global industries will benefit from the proposed industrialisation. It is the purpose of this research study, therefore, to extend the research into small software businesses to address the gap and hence answer the central research question.

1.3 Research Problem

Poor software quality results in a loss of profit and even loss of life at worst case. The general business problem is that insufficient focus is placed on the quality assurance of software products, in the ever-increasing small software business sector. SQA is a process that occurs in parallel with the software development process. The purpose of this research is to understand the costs relating to poor quality software and determine if industrialising the SQA process will reduce these costs for small software businesses and make its use more effective.

1.4 Research Aim and Objectives

The aim of the study is to determine whether industrialising the SQA process will help increase the effectiveness of SQA implementation in small software businesses and thus lead to effective implementation of software quality. The discussion in sections 1.1 to 1.3 above, leads to the formulation of the following central research question which will be developed further in this study:

Can Industrialisation of the SQA process help improve the effectiveness of SQA implementation in small software businesses?

To meet the main objective, the following subordinated objectives were found to be necessary:

- To get rich insights into the factors that affect the effectiveness of the current SQA processes in small businesses.
- To explore how industrialisation impacts the effectiveness of the SQA processes.

1.4.1 Research Questions

To meet the objective, it is necessary to answer the following sub-questions:

- What costs are incurred by the organisations to achieve their desired quality level?
- Do small software development businesses have a defined SQA process in place?
- How effective are the SQA activities in these organisations?
- How could the SQA process be industrialised in its application?
- Will this industrialisation improve the effectiveness of the SQA process?

1.4.2 Assumptions and Limitations

To improve the quality and integrity of the research findings and the evidence presented the authors should report their research assumptions, limitations, and delimitations. These assumptions, limitations and delimitations will set the context for how this research is focused (Nenty 2009).

The following assumptions have been made on this research study:

- Participants were honest with their answers during the interviews.
- The participants would willingly share documents of their processes as a secondary source of data.
- The number of participants selected were sufficient to conduct the research as data saturation would be reached.

The studies' limitations include:

- The data obtained during the interviews is limited to the participants experience in their existing organisation and not to their past experiences.
- The size of the sample may not represent the population.
- Information obtained will be limited to the participant organisations only.

The delimitations of this study are:

- This study does not aim to research the industrialisation of the software development process, only the SQA process.
- Only the software development process is under study and not the associated hardware integration.

1.5 Research Methodology

1.5.1 Total Quality Management (TQM) as a Framework

Whether software is developed by students in their classes, enthusiasts on open-source development teams or professional teams, quality of the software they develop must be addressed. Ensuring quality of software requires that the source code and its interaction with information technology infrastructure be considered. This will ensure that a quality system is produced. Software errors and defects can be introduced at each step of the software development cycle:

According to (Institute of Electrical and Electronic Engineers (IEEE) 2014) , software quality is “The degree to which a software product meets established requirements; however, quality depends upon the degree to which those established requirements accurately represent stakeholder needs, wants, and expectations”. Focusing on the quality of software products in a holistic, business driven quality approach will help organizations achieve their best structure for the entire organization.

For this reason, the conceptual framework in this study is based on the following concepts:

- total quality management (TQM), which encompasses:
    - industrialisation of software quality assurance
    - the right quality

TQM is an appropriate conceptual framework for this study since, by definition, TQM describes a management approach to long-term success through customer satisfaction. In TQM, all members of an organization participate in improving processes, products, services, and the culture in which they work. This is especially relevant for small businesses as employees serve multiple roles (Walkinshaw 2017).

TQM provides a systems approach to effective quality management which ultimately provides stability, growth and business success (Issac, Rajendran and Anantharaman 2004). TQM methods, as in other industries, must apply to the entire software development not just the development process.(Alamri and AbdulAziz 2016). TQM adopts an organisational approach to address the supporting issues with regards to

human resources and technical challenges that are likely to be encountered in software development (Alamri and AbdulAziz 2016).

There are two streams of software quality:

(1) Management of quality associated with the development process (Wieczorek, Vos and Bons 2014) and
(2) Testing, in terms of the software development model (such as Agile or Waterfall)

This is further accentuated by Sommerville (2016), who states that Quality management provides an independent check on the software development process. Quality assurance thereby, checks the project deliverables to ensure that they are consistent with the organisational standards and goals. Figure 1 below displays the parallel paths between quality management and software development processes.

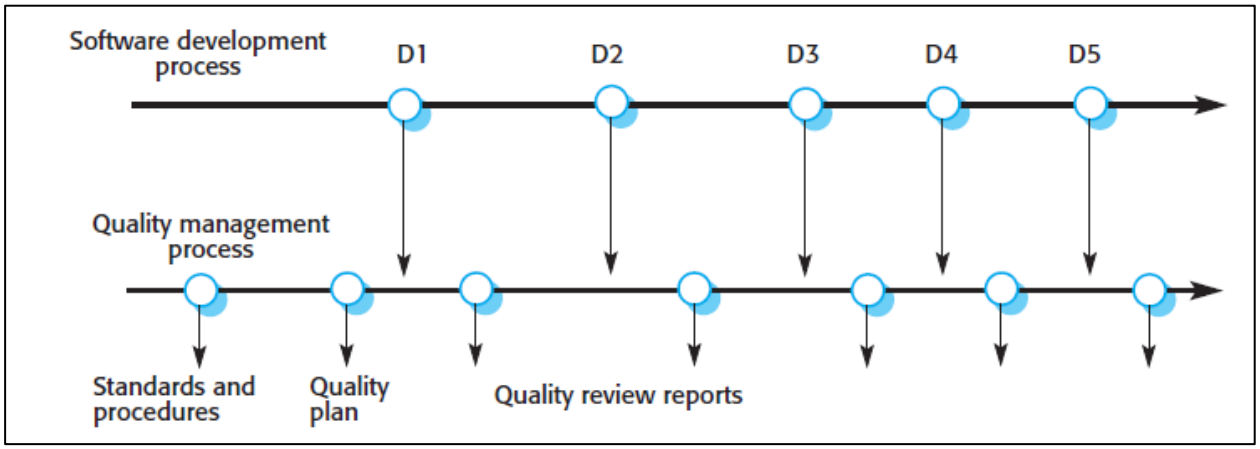


Figure 1 - Quality Management and Software Development Processes

Source: Sommerville 2016

For this study, quality criteria will be independent from the development models (such as Waterfall, Agile, V-model) since a holistic or total approach to quality will be investigated and the development model forms part of the operational layer of the business. Software quality assurance is a collective output developed by a team. This implies that all team members of an organization need to comply to the needs of the product to achieve the desired quality levels (Wong, Yu and Too 2018).

Wieczorek, Vos and Bons (2014) explain further that focusing on quality issues in a holistic, business-driven quality approach will help businesses reach an optimal

performance for the whole business. This emphasises the benefits of TQM in a business and is only possible if all three levels of a business, strategic, tactical, and operational are aligned in terms of next step for ensuring and improving quality of the business.

In concluding this conceptual framework, John, Kadadevaramath and Immanuel (2016) have asserted that there is a positive correlation between TQM implementation and software quality. This re-affirms that TQM provides a sound and meaningful framework for this research study.

1.5.2 Overview of research design and methodology

To answer the central research question, it is not necessary to prove that the SQA industry is currently industrialised but to demonstrate that it can, in the future, be industrialised. Interpretive philosophy will allow this research study to gather the information from 'actors' in the SQA space and the software development industry. In this way the researcher can gather information in a natural setting and interpret its results (Saunders, Lewis and Thornhill 2009).

The research strategy will encompass investigating companies that perform the SQA function to better understand the nature of the SQA challenges. The task would then involve analysing this information and formulating a theory (Creswell 2014). This research will therefore use the inductive approach, where a more informed decision about the research design can be determined (Saunders, Lewis and Thornhill 2009).

The absence of relevant information on the industrialisation of SQA process from the past is a justification for using the exploratory method in this study. An exploratory study will help understand the nature of the problem and explore the topic at different levels (Casula, Rangarajan and Shields 2021).

The timeframe of the research eliminates the use of the quantitative method as a research design choice since software project durations vary and relationships between variables through hypothesis testing is not practical (Antwi and Hamza 2015). The choice of this project is therefore a qualitative choice. Qualitative research explores the phenomenon under study through the participation of the research participants (Kapoulas and Mitic 2012). The chosen research method will use interviews as a data collection method.

Semi structured interviews will allow the researcher flexibility to explore the complexity of the research topic.

Since the problem being studied in this research has not been deeply investigated before, both primary and secondary research methods will be pursued. Primary research will involve interviewing the participants of the research whereas secondary research will involve reviewing the literature provided by the participants to arrive at an analytical conclusion. The study of the state of software quality assurance in a small software business will be done at a particular point in time. Therefore, the use of a cross-sectional study is chosen.

This research study will use semi-structured interviews as these are an appropriate instrument for qualitative research (Saunders, Lewis and Thornhill 2009). The interview questions will be based on themes to answer the objective of the research questions. Questions are categorized into areas of Strategy, Planning and Operations but differentiated by the stages of industrialisation (modularisation, standardisation, automation, and continuous improvement). An interview guide was developed in preparation for the interviews since a questionnaire was not to be used due to it being a fixed instrument to collect data (Adams 2015). The population of the participating organizations for this study are small software development companies within the country. To achieve the appropriate sampling, interview participants will be purposefully selected. This will ensure that the rich source of information of the participants align with purpose of this study (Palinkas et al. 2015).

The data collected during semi-structured interviews are at risk of reliability, bias, and validity issues. These challenges will be taken into consideration when reporting the results and drawing conclusions. Inductive data analysis will allow for the development of an answer to the research question in this study. This will involve using the raw data collected, organising the data, and coding the data into themes. Interpreting and verifying the meaning of the data and applying critical thinking to draw conclusions (O'Leary 2021) will then follow. The computer aided qualitative data analysis tool, Nvivo, was used to analyse the research data.

1.6 Significance of the Study

This study will provide insight into the issues experienced by small software development businesses and how to help them become successful and competitive by analysing their concerns and adaptation of processes that are available to larger software organizations. Industry standards and best practices, especially those that are available to the industrialisation of small businesses may be highlighted and made available to software development.

The body of knowledge available to larger organisations such as TQM, CMMI, ISO 9001, Six Sigma and Agile may be leveraged to small businesses through the analysis of this study. Industrialisation may be applied to parts of the software development lifecycle (SDLC) for example at validation and verification phase without being implemented at the acceptance testing stage (Wieczorek, Vos and Bons 2014).

Singh and Kannojia (2013) concluded that understanding the software quality models already used and its applicability is important to the participating organisations to gain optimum quality of their products.

This study may also contribute to positive social change by strengthening the SQA knowledge of QA leaders. The output of this study may impact small organisations by strengthening their efficiency and thereby their sustainability and longevity since small organisations contribute to economic prosperity of a community and provide employment opportunities.

1.7 Thesis Structure

This study is structured into different chapters addressing various areas of the research. Firstly, chapter 2, the literature review will provide a contextual background about software development, Software quality and industrialisation. Total Quality Management, TQM, will also be discussed in this chapter. Chapter 3 introduces the approach to apply industrialisation concepts to software quality assurance process. Chapter 4, Research Methodology, will detail the research design and process followed to address the research questions. The results and analysis will be discussed in Chapter 5 and finally, Chapter 6 (Conclusion and Recommendations) will present the authors recommendations and the conclusions of the study.

1.8 Conclusion

This chapter outlined the research study conducted and the research context. This chapter introduces the importance of quality software product delivery and its benefits and the struggles of small software businesses to achieve the levels of quality that are expected by consumers. This chapter briefly introduces the software development process and how it relates to the quality assurance process.

The aim of the study is to investigate if industrialisation, as proven in other industries, will provide an effective solution to software quality assurance challenges in small businesses. The sustained longevity of small organisation is important because of the contribution that small businesses make to the economic development of the communities in which they exist.

The next chapter, Chapter 2, Literature Review, will provide a review of academic literature used in this study with regards to the themes that emerge, namely software development, software quality assurance, small businesses, industrialisation, and Total quality management (TQM).

2 CHAPTER 2 - LITERATURE REVIEW

"High-quality software is not expensive. High-quality software is faster and cheaper to build and maintain than low-quality software, from initial development all the way through total cost of ownership." ~ Capers Jones

2.1 Introduction

Much of our evolution in industry and society to a large extent, has been through the introduction of software. Software has enabled our businesses and processes to evolve in a way that was not predicted. The software development process falls within the software engineering discipline. All aspects of software production, from software specification to the production and maintenance of the software product are overseen in the software engineering discipline (Sommerville, 2016). Engineering is about finding solutions to a need within the required quality with the constraints of time and budget and this applies to software engineering as well.

From the 1970's software has been a part of our lives and currently reinforces and supports our everyday activities. In many cases, we are not aware of its presence until it fails or crashes. The software industry generates a huge amount of revenue and continues to grow in scope and size.

Many software projects, today, do not find the correct balance between quality and time-to-market and therefore do not remain within budget (Wieczorek, Vos, Bons, 2014). This is especially true for small businesses who struggle with the balance (Nirnaya et al, 2016 and Janes, 2017). Several different SQA approaches (tools and methods) have been developed and introduced to small businesses but with confusing or ambiguous outcomes (Yahaya, 2017).

Small entities cannot afford these tools for SQA, and automation and therefore many software companies have a limited lifespan, having to shut down their companies eventually. With small number of resources, small businesses consider SQA activities difficult to execute, resulting in products that may risk their quality requirements. This is evident in a study of the implementation of CMMI (Capability Maturity Model Integration)

done by (Lester, 2010). However (Lester, 2010) highlights that communication within smaller companies is more effective than that in medium sized companies.

Janes (2017) has proposed an automated set of tools to help small businesses improve their SQA. These tools include, Gitlab, Jenkins, Jira, SonaQube and SonarLint. These are project management, version control and test automation tools. Sowunmi (2016), recommended that tools such as ConQAT for code reviews be used to help small businesses after researching the SQA challenges experienced by companies in developing countries.

In a study done by Yahaya *et al.* (2017) a software quality assurance model was proposed, which is applicable to small businesses. However, Yahaya *et al.* (2017) also recommends that it be used for future study to find a more effective model. They believe that consideration be given to strategy of the organisation for a more sustainable SQA model.

It can be seen that numerous studies have generated quality tools and models to help small businesses improve their software quality assurance aspect of the business but as also highlighted by (Mishra, 2009) future studies are needed to focus on specific needs of small software organisations.

A survey done by (Capgemini, 2012) showed that 42.6% of a company's IT budget is spent on keeping existing software running and maintaining software to address defects and change requests. This implies that customers and users of software want their products to be innovative, flexible, and inexpensive yet maintaining their desired level of quality. It makes sense then to standardize, automate and modularise the development and business processes that lead to the final product, particularly in small software businesses.

According to (Águila, 2020), we entered the second era of our software engineering evolution, from 1993 to 2018, where software engineering started to be considered as an industrial business. Software companies became software factories and they needed to assure quality of their products. Development processes started to emerge, and software development process and software product were distinguishable. Frameworks such as CMMI (Capability Maturity Model Integration), adapted from the manufacturing field, were developed for the software development process (Chaudhary, 2017). Software factories had the goal of producing software with reduced defects, faster, with better quality and

thereby increasing the profit earned by the businesses. Quality management methodologies that were applicable to traditional organisations then became applicable to the software development industry as well.

According to Okumoto, Mijumbi and Asthana (2018) defects are introduced into a software product at various phases of the process. New defects and existing defects can be transferred from phase to phase. Section 2.5.1 gives a detailed review of defect introduction into a software product. Some activities in the software process involve Identification of needs, scope definition, project planning, requirements analysis, design, development, integration, test, deploy, maintenance, and disposal as shown in Figure 2 as adapted from models presented by (Ruparelia 2010).

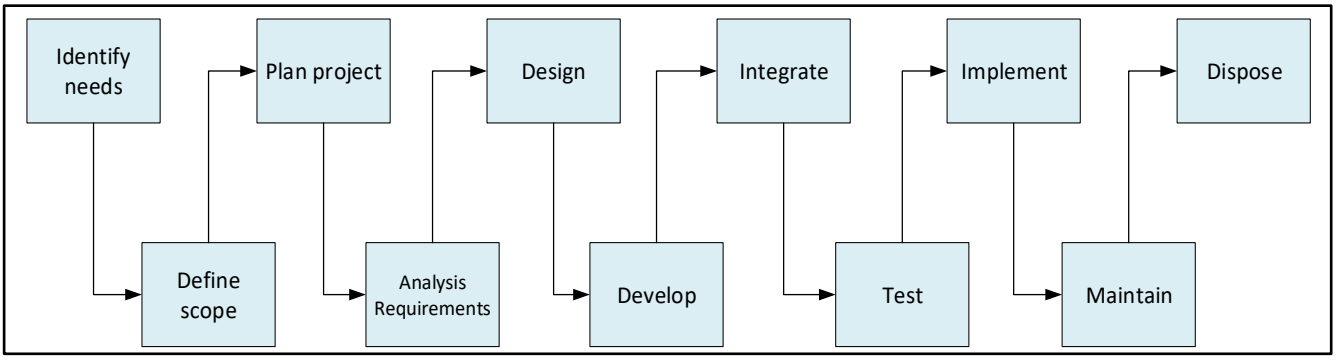


Figure 2 - A typical Software development process

Source: adapted from (Ruparelia 2010)

Wieczorek, Vos, Bons, (2014) believe that industrialisation may apply to the whole cycle, individual boxes, or two independent parallel streams of the whole process – namely, the development process and the quality assurance process.

Wieczorek, Vos, Bons, (2014) have chosen to develop a model (RISSQ model – Right Software and Systems Quality) for the industrialisation of the quality process based on the modularisation, standardisation, automation, and continuous improvement of processes. Their model, however, does not take into consideration the constraints and limitations faced by small businesses and instead they look only at global and large software development companies.

The objective of this research study is to answer the question of how industrialisation of the SQA process can help increase the effectiveness of SQA in small software businesses as stated in section 1.4.

The sections that follow provide an overview of the literature on which the research will focus. Specific aspects of the research that were introduced in sections 1.1 to 1.4 will be categorised and drawn on in this review. This literature overview serves to introduce the sources used in this study and places these sources into context for this research study. The framework for the argument hence becomes apparent.

The sections that follow are organized in the following way. Section 2.2 gives a high-level look at small businesses and Section 2.3 reviews Software development, its evolution and why software is now seen as an engineering discipline. An overview of some process and methods for developing software are given. The next section (section 2.4) presents the role of quality management and total quality management (TQM) in general and then in particular to small businesses (section 2.4.1) and to software quality assurance (section 2.4.2).

Section 2.5 then shows Software Quality Assurance (SQA) process steps and how its costs have an impact on an organization's performance. Section 2.6 looks at international standards and support for software quality and finally, in section 2.7, the possibility of industrialising the quality assurance process is considered and status of the industry is highlighted.

These main themes in the literature review are described in Figure 3. These themes are:

- Software Development and Engineering
- SQA (Software Quality Assurance)
- Industrialisation
- Small Software Businesses

For more clarity on the content of the literature review, sub-themes are also shown in Figure 3.

To further place into context the hierarchy of themes discussed in this chapter - Literature review, a model of the themes has been depicted as shown in Figure 4. The final concentric circle represents the central research question of this study. Each theme, in each section in the literature review consistently refers to small businesses. This is depicted by the downward arrow through the concentric circles.

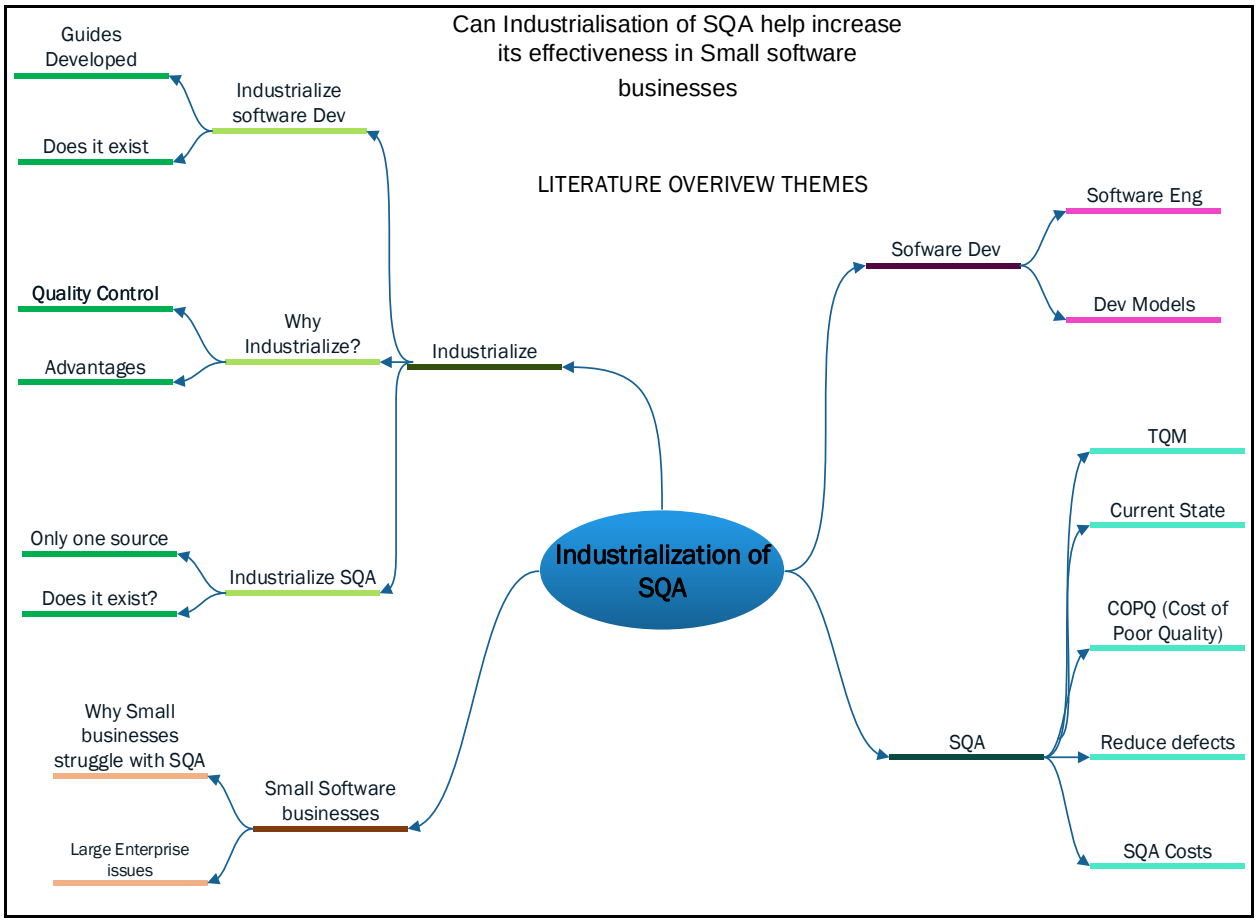


Figure 3 - Literature Overview Themes

(Source: Developed by Author)

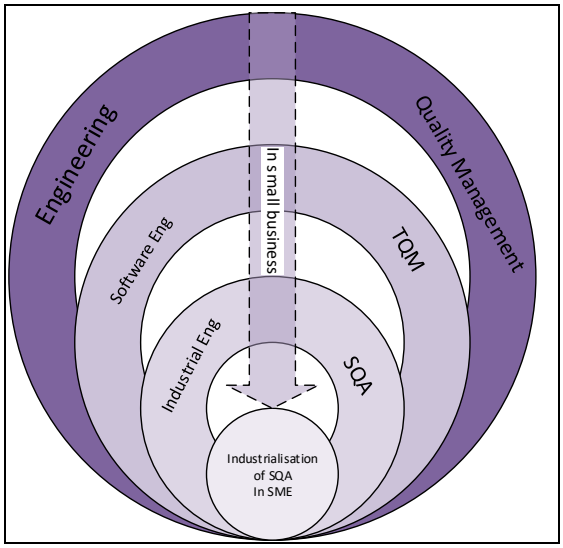


Figure 4 - Literature theme hierarchy

Source: Developed by Author

2.2 Small software businesses

The definition of small enterprise is stated as ‘a separate and distinct business entity, together with its branches or subsidiaries, managed by one or more owner in any sector or subsector of the economy’ (South Africa 2019). Small businesses continue to struggle in South Africa due to a wide range of challenges and their survival beyond 5 years is rare (In on Africa 2018). Small enterprises are categorized, depending on their industrial sector, as micro (0-10 employees), small (11-50 employees) or medium (51-250 employees) (South Africa 2019).

According to (O’Connor and Laporte 2014), small software businesses make up almost 95% of businesses worldwide. Software products are supplied from large software development companies, but many small organisations have recently started up to produce software. So small and very small software business are satisfying the market demand for software development. Larrucea, (2016) asserts that small and very small organisations struggle with implementing standards such as those presented by ISO, CMMI and IEC. Small companies believe that the overhead required to implement the software development standards such as ISO/IEC 25010, a standard for software quality, have been developed for large companies in mind. In their study, (Sultana, Syeed, Fatema, 2020) found that many small companies are reluctant to follow any quality improvement process.

Some differences between small software businesses and large enterprises give some insight into why small software businesses struggle with implementing a structured SQA process (O’Connor and Laporte 2014). For example, small software businesses tend to operate in an unstructured and operational orientation whereas large companies have a more structured and strategic direction. On the other hand, small software businesses are highly flexible unlike large enterprises and can easily adapt to other ways of operating.

Small business in every sector of the economy needs help. The focus of this study, however, is the software quality component in small software development businesses.

2.3 Software Development and Engineering

2.3.1 Software as an Engineering Discipline

In this section an overview of the software development process over time and highlights its evolution is given. Aspects of the software development process and its focus on producing product of acceptable quality is presented.

Software and information system development has its roots in as early as 1940 (Matkovic and Tumbas 2010). At that stage and up to the 1960s information system development was considered a craft and developers were craftsman in the software business. Soon after that the Software Development lifecycle (SDLC) was introduced to bring structure and rigour to the development process. The SDLC gave the industry a basis for methodological approach to software development. Methodologies added a systematic and scientific way to develop software that allowed for time, budget, and quality to be managed. In this way input artefacts could be converted to output artefacts under a managed and controlled way as in any other engineering discipline.

These methods had the overarching responsibility of reducing the risk of error of the output. According to Matkovic and Tumbas (2010), these models will continue to evolve with the goal to minimize risk of failure and lowering the cost of project maintenance. The common software development models are the Waterfall Model (with elements of Agile) and the Agile model itself (Saeed, 2019).

What started off as an approach to computation has evolved into a discipline of engineering. This discipline however, is not as mature as other disciplines according to a study done of software engineering principles (Al-Sarayreh, Meridji and Abran 2021). To emphasise this Baptista and Salles (2012) indicate that the software engineering field is still experiencing challenges where other engineering disciplines may have overcome.

As the software industry developed a foothold and size and complexity of products increased, standards such as ISO 12207 and ISO 25010 were developed to provide a guide for improved software quality product and processes. It was becoming clear that software development processes had many similarities to engineering processes.

Making the association of software development as an engineering discipline makes its relationship with other engineering streams, such as quality engineering and quality management easier to conceptualise and apply as will be seen in the following sections in this chapter.

2.3.2 Software development processes and their challenges

As in all industries. processes describe the activities required to achieve an end goal. A software development process encourages development of software in a systematic and disciplined way. When software is developed by a team the use of a process, or a lifecycle model becomes indispensable for successful completion of a project or product. Software development organisations understand that the adherence to a suitable lifecycle model helps to produce good quality software and helps to minimize the chances of time and cost overruns.

Software development lifecycles (SDLC) have been defined to show the different stages that software must go through to change from a customer requirement to a fully developed product and finally to a stage where it is no longer useful. Sommerville (2016) explains that there is no universal process model that is right for all kinds of software development. He further emphasised that organisations choose the process model that best suits their customer's and regulatory requirements.

SDLCs are typically plan-based or agile in nature. There are several plan-based software development processes that have been widely used in the software industry. These are some examples:

- Waterfall Model – This is the most popular and widely used plan-based software development process according to (Prenner, Unger-Windeler and Schneider 2021). It involves a sequential approach to software development, where each phase is completed before moving on to the next phase (Sommerville 2016).
- V-Model – This is an extension of the Waterfall model but places greater emphasis on testing. Each stage of this development process is associated with an associated testing phase.

- Spiral Model – This is an iterative model and involves a series of prototyping and testing cycles. Each cycle involves planning, risk analysis, development, and testing.
- Rational unified process (RUP) – Dragos (2021) states that RUP is a process framework provides a customizable, iterative approach to software development . It is based on a set of best practices, which are organized into four phases: inception, elaboration, construction, and transition.
- Agile Model - Although Agile is not a plan-based methodology, it does involve planning, and the planning process is typically iterative. Agile development involves a flexible, collaborative approach to software development, with a focus on delivering working software early and frequently.

Prenner, Unger-Windeler and Schneider (2021) state that while these plan-based methods have been widely used in the past, they are now often seen as less flexible and adaptable to newer more iterative approaches such as Agile. Although these models have their advantages, they also have some challenges that can make them less suitable for certain teams and projects (Sommerville 2016). They are criticised for their rigidity which makes changes to a project difficult and limited. They also lack customer involvement which can result in not meeting the customer's requirements. These models can be more risky than other models as they require a large investment of time and resources upfront before any working code is produced. If the project fails, this investment can be lost. With plan-based models progress is often measured based on how well the project is following the plan rather than how well it is meeting customer needs or delivering value. This can lead to a false sense of progress and can make it difficult to identify and address problems.

On the other hand, while agile development has become popular due to its flexibility and adaptability it also has its own challenges. Due to its flexibility, a project's completion is difficult to predict. This implies that budgeting and planning are a challenge. Agile depends heavily on the collaboration and communication among team members. Maintainability of software developed using the agile model is challenging since documenting the project is not emphasized. Agile development does not have the same milestones as plan-based models, which can make it difficult to measure progress. This can make it challenging to communicate progress to stakeholders or to determine whether the project is on track.

Prenner, Unger-Windeler and Schneider (2021) assert that companies get the best application of the models when a hybrid approach is used to suit and organisation's needs, particularly regarding the following factors:

- Requirements change rate
- Chaos vs order
- Number of personnel
- Loss due to the impacts of defects
- Skill level

These factors play an important role in how a small software business conduct its business.

2.4 (Total) Quality Management (TQM)

To answer the research questions in this study it is important to understand the foundations of quality management. This section will delve into quality management and its impact on the software industry, and in particular small software businesses. Total Quality management (TQM) as a philosophy for managing quality forms the framework of this study.

The principles of TQM as stated by (Mohammed, Tibek and Endot 2013)

- Leadership
- Customer focused organisation
- Involvement of people
- Process approach
- System approach to management
- Continual improvement
- Factual approach to decision making

The impact of TQM in small industries is covered in section 2.4.1 and then the relationship of TQM and SQA is explored in section 2.4.2.

To provide a contextual framework for this research study and the several themes used in this literature review, a representation of the themes has been presented in a house format in Figure 5. The format of a house, adapted from the 'house of quality' in (Wieczorek, Vos and Bons 2014) appropriately depicts the components of this literature review. This 'house' will be built up further in Chapters 3 and 4 to further illustrate the roles of the several components in this study to answer the central research:

Can Industrialisation of the SQA process help increase the effectiveness of SQA implementation in small software businesses?

Total Quality Management (TQM) is a comprehensive approach to quality management that involves all employees in an organization in a continuous effort to improve quality and customer satisfaction, hence TQM stretches across the entirety of the small business in Figure 5. TQM has progressively gained more prominence and importance to both large and small firms widely and turned out as a benchmark for quality, organizational sustainability, and improved business performance.

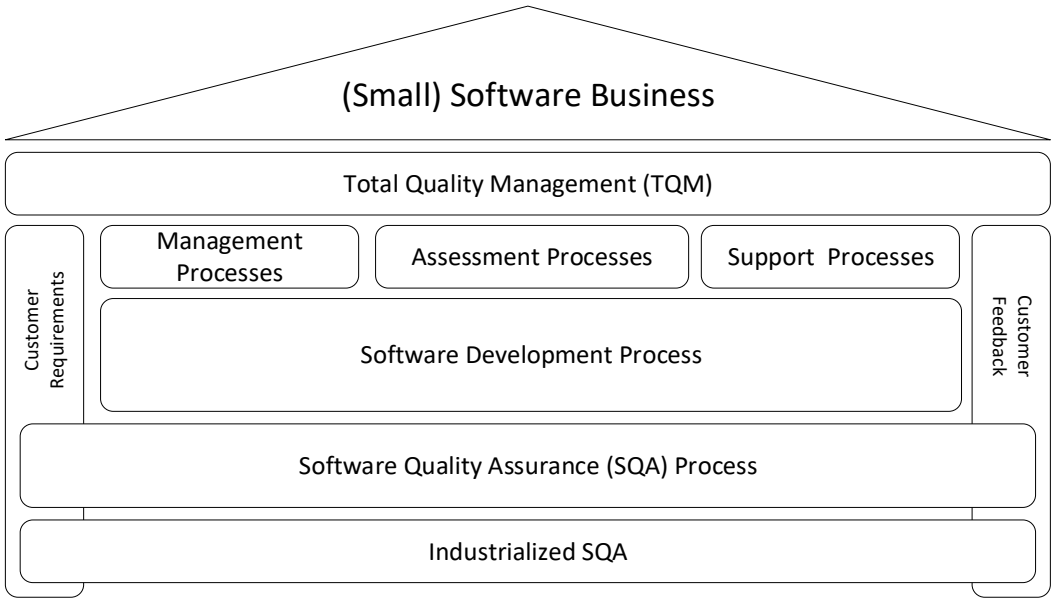


Figure 5 - The small business quality house

Source: Adapted from Wieczorek, Vos and Bons, 2014

The software development process commences once customer requirements become available. Management, assessment, and support processes form the support system

that every organisation requires for sustainability. The span of the SQA Process component and the Industrialized SQA component emphasises the scope of this research study and the parallel nature of these two components. The industrial component will be expanded on in section 2.7.

2.4.1 TQM in Small businesses

Saha *et al.* (2022) have found that TQM principles help an organisation to increase customer satisfaction, reduce errors, and improve operational performance which directly contributes to the organisation's sustainability. There is no doubt that many large businesses and corporates such as Toyota, Ford and Xerox have had success with their implementation of TQM. Zhou (2016) points out that many small businesses have become important players in large supply chain networks and they either voluntarily or have been pressured into application of TQM to gain and sustain a competitive advantage.

Implementing TQM in a small organization can, however, be challenging, as it requires a significant investment of time, resources, and effort. Toke and Kalpande (2020), state that models for TQM implementation are available but there are hardly any that describe a model for small businesses, yet small businesses contribute significantly to the economy in any country. Some studies, though, indicate that there is a correlation between TQM implementation and sustainability in small businesses (Magd and Karyamsetty 2021).

The relationship between organizational size and TQM implementation can vary depending on a variety of factors. While TQM can be implemented in organizations of any size, there are some general trends in terms of how size can affect TQM implementation. For example, smaller organizations often have a more centralized structure and are more flexible, which can make it easier to implement TQM initiatives. Small organizations may also have fewer layers of management, which can make it easier to involve employees in quality improvement efforts and create a culture of continuous improvement (Georgiev and Ohtaki 2020). This is further emphasized in a study that identifies that the success of TQM initiatives depends largely on the leadership style of businesses, who ought to focus on creating an organisational culture that is conducive to supporting TQM

implementation in small businesses (Sahoo and Yadav 2018). Quality management researchers have agreed that the critical success factors of TQM can be grouped into two main categories – hard factors (such as statistical process control (SPC) and production techniques) and soft factors (such as top management involvement and leadership).

2.4.2 TQM and SQA

Given the success of TQM in improving the quality of manufactured products and that software development is considered an engineering discipline (section 2.2.1), several SQA practitioners have suggested that applying TQM practices to software development is an effective way to improving software quality. The Software Engineering Institute (SEI) at Carnegie Mellon University formalised the TQM based principles as the Capability Maturity Model for Software Development called the CMMI. The premise underlying the SEI's work on software process maturity embodies TQM principles (Mistrík *et al.* 2015), namely that quality of the software product is largely determined by the quality of the process used to build and maintain it.

Within CMMI model for software development the core engineering process capabilities, project management capabilities and process management capabilities are presented alongside SQA, software quality control and software process improvement, to form a comprehensive TQM based approach to the development of software. CMMI is further reviewed in section 2.6.1.

2.5 The Components of Software Quality Assurance

As discussed in section 2.2.2, it has been observed over the past few decades that traditional plan-driven software development methods such as the Waterfall methods have been replaced by change-driven development methods like agile development. In some cases, a hybrid of these methods are used. Lagstedt, Dirin and Williams (2021) explain that as software development methods change constantly quality assurance practices must also change. It is important to note that SQA is not a final development step but an ongoing assessment mechanism that continues during the ongoing maintenance phase of a software product.

There have been many definitions of SQA used by researchers and authors of SQA literature, but the most commonly used definition is offered by Institute of Electrical and Electronic Engineers (IEEE) (2014):

"Software Quality Assurance is a set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes.

The definition further states that:

"A key attribute of SQA is the objectivity of the SQA function with respect to the project. The SQA function may also be organizationally independent of the project; that is, free from technical, managerial, and financial pressures from the project.)"

It is not the aim of this study to align with any software development methodology (such as Waterfall, scrum, agile etc), nor is it the aim to develop a standard SQA process to align to any of the development methods. The aim is to answer the research question,

Can Industrialisation of the SQA process help increase the effectiveness of SQA implementation in small software businesses?

Hence generic views of the components of the SQA process will be reviewed for its ability to be industrialised.

Figure 2, in section 2.1 describes components of a typical software development process. In this section, however, the components of a typical SQA process will be discussed and its alignment to a typical software development process will be reviewed.

A six-component company-wide structure of SQA identified by (Galin 2004), which is also backed up by similar holistic structures described by other authors such as Institute of Electrical and Electronic Engineers (IEEE) (2014) and Wieczorek, Vos and Bons (2014), is clearly an adaptable guide for any software quality assurance process.

The six components are:

- Pre-project quality
- Project life cycle quality
- Infrastructure and error prevention and improvement

- Software quality management
- Standardisation, certification and SQA system assessment
- Organising for SQA

While most research has consistently highlighted the advantages of these software quality assurance components, some challenges exist. The main challenge being that quality assurance activities are seen as unnecessary and unproductive to some because they do not directly add value in terms of code and features (Itkonen, Rautiainen and Lassenius 2005). As a result of this perceptions some activities are skipped. This is often the case in small software businesses where resources are limited.

Based on the chosen TQM framework for this study (discussed in section 2.4), SQA is not limited to the development process, nor are its activities limited to the technical aspects or the functional requirements, but it is also involved with the activities that deal with scheduling and budget of the project. The extension of the SQA function across the entire scope of the 'house' provides an overview of the scope of this research study.

To demonstrate the scope of the SQA process and its alignment with the software development, Figures 2 and 5 have been superimposed in Figure 6. The figure serves to emphasise that this research study is focussed on the industrialisation of the independent SQA process, indicated by the last stream on the quality house.

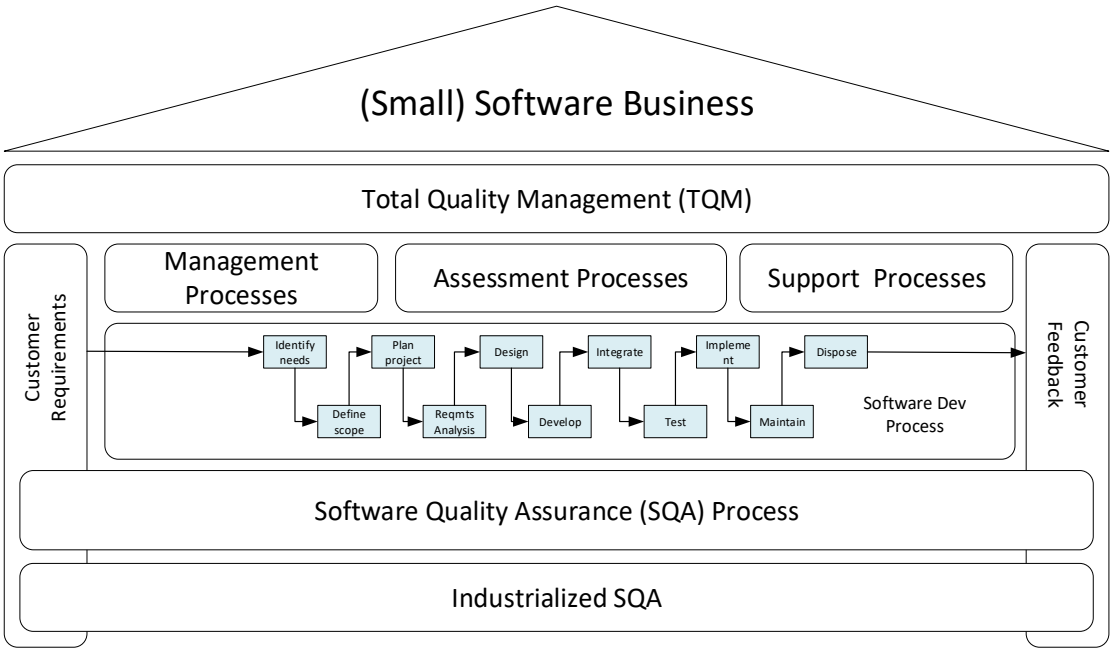


Figure 6 - The small business quality house (with software development process steps)

Source: Adapted from Wieczorek, Vos and Bons, 2014

2.5.1 Software defect flow

In line with the definition of SQA given in section 2.5 above it is clear that software quality assurance activities will limit downtime and vulnerabilities that may have unprecedented consequences for the user of the software product and the producer of the software product (Okumoto, Mijumbi and Asthana 2018). These vulnerabilities and downtime may be the result of errors, defects, and failures. These terms, which describe software problems, are distinctly different. According to (Institute of Electrial and Electronic Engineers (IEEE) 2017) an error is a human action that produces an incorrect result, whereas a defect is a problem which, if not corrected, causes a failure or an incorrect result. A failure, on the other hand, is the termination or inability of a product to perform its required function.

Researchers in the field of software quality continually investigate these issues to discover means to eliminate these problems (Laporte and April 2018). Errors can occur at any stage of the software development phases and throughout its lifecycle.

In this section the flow of defects through various development phases are examined briefly. This enables the study to further emphasize the need for process steps within the entire software development process and beyond to enable software quality assurance activities.

Okumoto, Mijumbi and Asthana (2018) show, in Figure 7, how new defects are introduced in the requirements, design and coding phases and how old defects are carried over from previous releases into the current release. Defects are typically removed from every phase as well as during customer usage of the software. These activities to remove and prevent defects form part of the software quality assurance process. Each of these activities are executed at the expense of project deadlines and project costs and ultimately customer satisfaction and the businesses bottom line.

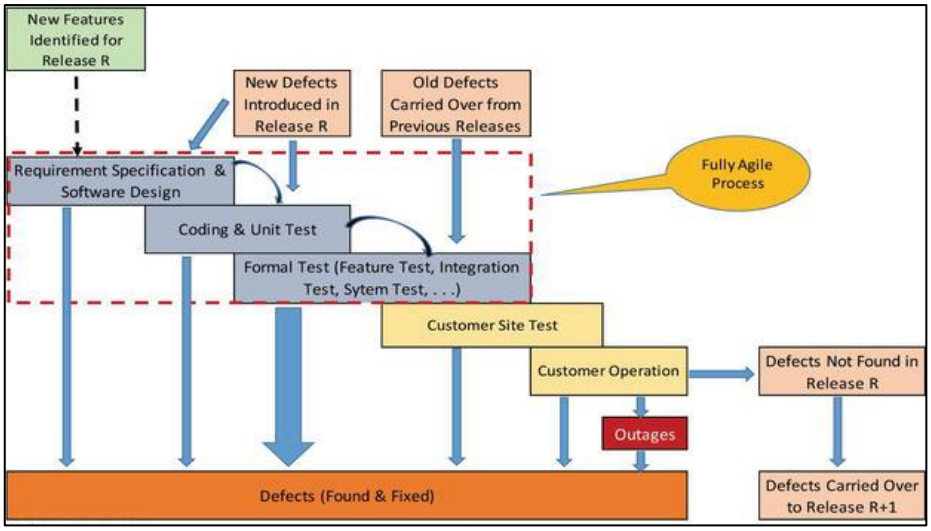


Figure 7 - Defect Flow Development, test, field

Source: Okumoto, Mijumbi and Asthana 2018

Early defect prevention and prediction is needed to minimize the early identification of software quality and cost overruns. SQA activities, thus, involve preventive actions (such as reviews, auditing and testing) and corrective actions (such as debugging and re-working). The next section will delve deeper into the activities and costs incurred by software projects when software quality activities are implemented and/or ignored by an organisation and its impact on the business.

2.5.2 The associated costs of SQA

To answer research question 1,

"What costs are incurred by the organisations to achieve their desired level of quality?"

further depth and understanding of Cost of software quality (CoSQ) is required. Laporte et al, (2012), explain that small software businesses find difficulty in measuring their quality costs due to the lack of resources. The general business problem is that inadequate SQA processes negatively affect some software development organisation's profits.

Several quality cost models haven been developed by renowned quality experts, such as Juran in the 1950s and Feigenbaum in the 1990's. The definition of quality most

recognised and used by many researchers and authors is undoubtedly the one by (Juran and De Feo 2010) which states that quality costs are the costs that would disappear in the organization if all failures were removed from a product, service, or process. It is therefore understood that achieving and maintaining a desired level of quality to any product or process is realised after a financial investment has been made.

For this study, the Cost of Software quality will be discussed further to achieve an answer to the central research question. The American Society of Quality (Daughtery 2013) further describes the cost of Quality to be:

$$(CoSQ) = (CoPQ) + (CoGQ) \quad [1]$$

where:

CoSQ = cost of software quality

CoPQ = cost of poor quality = internal failure costs + external failure costs [2]

CoGQ = cost of good quality = prevention costs + appraisal costs [3]

With reference to equation [2], Internal failure costs include costs of correcting errors before the software is installed for use by the customer and external failures costs arise when failures detected by customers are corrected. In equation [3], prevention costs are incurred to prevent or avoid quality problems, whereas appraisals costs are those associated with actions designed to find quality problems (Krasner 2018).

In recent years, technical debt (TD), a metaphor from economics, that refers to inefficiencies during software development that lead to additional maintenance effort has attracted significant attention from academia and industry (Arvanitou *et al.* 2019). As stated in the report by Krasner (2022), the main problem with TD is the lack of ways to measure that debt, however the estimated values for TD in the USA is $1.52 trillion. For the purposes of this study, TD will not be considered.

The steps and activities in a software quality assurance process will incur these costs when implemented within the SQA process. Some of these activities for internal failure costs are generated by reworking the product due to non-conformances that may be detected during testing. External failure costs comprise of warranty charges, rework costs, recalls, returns, patches and repairs. Krasner (2018) also explain that while these costs are the most apparent, they are difficult to quantify, especially, costs such as loss of sales and tarnished reputations that result from these external failures.

Lean management, a set of principles, a holistic approach aiming at providing the right product in the correct amount of time at the right place with the right quality. In lean management, reworking or fixing defects is considered a non-value-added activity (NVA). A non-value-added activity is an activity that does not transform a product, nor is a customer prepared to pay for such an activity. The target of any software development organisation should be to deliver a product with the lowest possible defect rate (Lárusdóttir, Cajander and Simader 2014).

Measuring, evaluating, inspecting, testing, reviews and auditing are some activities that are used to find problems in quality. These are appraisal costs. The biggest cost element in appraisal costs is usually testing, validation and verification activities. Prevention costs, such as those due to planning, review and analysis, training and quality audits prevent further failure, thereby reducing the costs of poor quality (Laporte and April 2018).

Nonetheless, there are several challenges to implementing a measuring system for the cost of software quality (as in other industries). The difficulties that affect the accuracy of quality cost data are inaccurate and incomplete identification of classification of costs, negligent reporting and biased reporting for internal and external failure costs (Daniel 2018). These problems impinge on the measurement of the costs of software quality.

It is logical that software organisations need to make a significant contribution towards achieving and maintaining good quality software products. It is also understandable that small software businesses are not always able to commit to the resources required to do this. An effective and efficient software quality assurance process will aid small businesses achieving this goal and thereby achieving business success.

2.6 Software Quality Assurance Standards and Models

Standards and management systems help organisations to meet their goals and to continuously improve their products and processes. Several software quality assurance standards have been compiled to help organizations formalise their SQA efforts. This section will introduce some common quality management standards and some project management standards.

2.6.1 Quality Management standards

Quality management standards set out ‘what’ to achieve for an assurance of the quality of its products and processes. ISO 9001 is an international standard that specifies requirements for a quality management system. Organisations that comply with these requirements and certify to these standards show that they can consistently be able to provide products and services that meet customer and regulatory requirements. ISO 9001 is applicable to small and large entities, both in the manufacturing and the service provision sectors. While the ISO 9001 system, when well-managed, is successful in most organisations, Sfakianaki and Kakouris (2020) have found that smaller businesses struggle more than larger businesses from difficulties in adopting the system. They further emphasize that obstacles to ISO 9001 that small businesses experience are associated mainly with bureaucracy, demand of money, time and resources, lack of top management commitment and the reaction of employees.

In 1991 the Software Engineering institute of Carnegie Mellon University defined a capability maturity model (CMMI) for the software community (Daniel 2018). The model enables organizations to measure, build, and improve capabilities to improve their overall performance. CMMI divides an organization’s capabilities into five levels: the initial level, managed level, defined level, quantitative management level and optimization level. Once again, CMMI is a resource intensive system to implement and maintain and small software businesses struggle to succeed with its adoption.

2.6.2 Software process management standards

Process standards, as opposed to management standards, focus on methodologies for carrying out software development and maintenance, that is, ‘how’ a project is to be implemented (Galin 2004). Some examples of these standard are ISO/IEC/IEEE 12207, ISO/IEC 29110, ISO/IEC/IEEE 15289, IEEE 730.

ISO/IEC/IEEE 12207 establishes a common framework for software lifecycle processes. It applies to the acquisition of systems and software products and services, supply, development, operation, maintenance, and disposal of software products and the development of the software part of a system, whether performed internally or externally to an organization. ISO 12207 does not prescribe a specific software life cycle model, nor does it require the use of a management system. It is intended to be compatible with

ISO 9001. This is a limitations of ISO 12207 as expressed by Laporte and April (2018). Laporte and O'Connor (2017) have explained that small organisations do not have the expertise and resources to adapt these standards for their applications and therefore do not have success. For this reason, the ISO/IEC 29110 for very small businesses was developed. This standard is designed to help small businesses improve their software development processes by providing a framework for developing, implementing, and assessing software engineering practices that are appropriate for their size and scope. The standard covers a range of topics, including project management, requirements management, software design, testing, and configuration management. This standard however, does not cover all areas of software development, such as software security, data management, or user experience design.

ISO 15289 focuses specifically on the management of systems and software engineering documentation. It provides guidelines for creating, managing, and maintaining documentation throughout the software development life cycle. The standard is intended to help organizations ensure that their documentation is of high quality, consistent, and effectively supports the development, use, and maintenance of software systems.

ISO 730 is an international standard that provides definitions and guidelines for software quality assurance and testing. sQA according to IEEE is a set of proactive measures to ensure the quality of the software product.

While standards can provide significant benefits for software development companies, they also come with potential disadvantages. Implementing and complying with standards can be expensive, particularly for small or mid-sized companies. The cost of training, documentation, and certification can be prohibitive, especially if the company does not have the necessary resources. Standards can be perceived as bureaucratic, with a focus on compliance rather than innovation or customer needs. This can lead to a negative perception among employees and customers, which can affect morale and sales of an organisation.

2.7 Industrialisation

The manufacturing industry serves as a model for software in since it has taken products from a ‘skill’ to mass production. The objective of an industrialised approach is the

reduction of costs per ‘piece’ or reduction overall production costs Wieczorek, Vos and Bons (2014). Industrialisation is defined as the implementation of standardized and highly productive methods in order to increase efficiency and reduce cost.

2.7.1 Why industrialisation matters to Software Quality

Software development is faced with contradictory demands to deliver product in shortest time at the best prices vs an efficient and economical delivery of products. This is true for software development teams and software service providers.

In industries that are industrialised, the quality assurance function usually starts with one or a few people who do their jobs individually when producing a certain product or output. For a simple product this method will work but as the process/product becomes more complex more people with specific skills are needed. As soon as the workflow is understood and suitable tools and technology are available, flows or a part of a flow can be automated and hence industrialised. This research study assumes that quality assurance is an engineering discipline.

Industrialisation of software can be considered to lag industrial production by almost two decades Hochstein *et al.* (2007). But the principles of industrialisation may be transferred to the software development area too. Standardization and specialization advance the level of reuse and enable automation of repetitive and menial tasks, whereas creative tasks, such as product design, are still performed by highly skilled workers. Further, efficiency gains may be achieved if simple but labour-intensive tasks are automated.

2.7.2 Industrialisation of Software Quality Assurance

The five basic principles of industrialization of processes are:

- Modularisation – this involves decomposing a process into smaller subsystems that can be worked on independently yet can function together as a whole (Miguel 2005). This is done so that intermediate artefacts may be worked on in different steps. In the case of SQA, examples of these artefacts are test cases, test scripts, test results and test reports which are typically used in the testing phase of a software development project.
- Standardisation - is the process of making something conform to a standard. In technical terms, it means implementing and developing standards based on the

consensus of different parties (such as users, interest groups, standards organizations, and governments) (Xie *et al.* 2016). This allows modularised processes to be conducted throughout the organisation in a uniformed way to avoid generation of new workflows every time a new task is started. Standardisation reduces the risk of defects being produced from processes. Standards may be unique to a company, or these companies may choose international quality standards such as ISO 9001:2018, ISO 25010 (IEC, 2011), ISO 15504 and CMMI. These standards are software development standards and maturity models.

- Specialisation – is concerned with ensuring not only that standardised processes are conducted throughout the enterprise or an autonomous organisational unit, but that people have the right qualification and skill for the tasks they perform. In small software businesses, software developers, who are highly skilled, also wear the hat of the software quality assurance personnel.
- Automation - involves replacing repetitive manual activities by automated activities. In the case of SQA, the generation and use of automatic test tools are a typical example of automation. Defect logging and tracking is also possible to automate during the testing stage.
- Continuous Improvement - is used to ensure that the desired level of quality is achieved and improved. Continuous improvement methods, such as Lean and Six Sigma may be applied to the SQA process as with processes in other industries. For example, the organisation may have key performance indices (KPI) to measure delivery dates or number of changes to a software product. Chosen continuous improvement methods will allow for the deviations from KPI to be analysed and improved.

The goal of an industrialised approach in any industry is to reduce production costs. Wieczorek, (2014) believes that when a process or workflow is fully understood, and the tools and technology are available, a workflow may be automated. The factory approach may then be applied to this process. Their guide to industrialising software development presented by (BITKOM, 2010) have provided several researchers and authors a basis for their positions on the possibility of industrialising software development but not its parallel SQA stream.

The notion that the software quality assurance (SQA) process can be industrialised was raised by (Wieczorek, 2014). The limitation in Wieczorek's literature, however, is that

large and global industries will benefit from their proposed industrialisation. It is the purpose of this research study, therefore, to extend the research into small software businesses to address the gap.

The specific steps involved in industrializing the SQA process may vary depending on the business's needs and goals. Wieczorek, Vos and Bons (2014) have provided a general framework to follow in their industrialised house of quality.

2.8 Conclusion

This chapter provided a review of the literature on the state software quality assurance in small businesses. The secondary intention of the chapter was to sketch an understanding of the software quality assurance domain within small businesses with TQM as an underpinning. The literature used in this chapter, addresses the four main areas of study from which the thesis draws. These are small businesses, software development, software quality assurance and industrialisation.

The development of the small business quality house was intended to simplify and show how these four areas of study interact and form a basis for this study.

Several authors and practitioners have put forward ideas of industrialised software houses in recent years. From the reviews, it has emerged that there is debate about how small businesses should apply software quality assurance effectively. Scholars argue that small businesses are essential to the economy in any country and that quality of software is a necessity for the sustainability of these businesses but not many have determined the most effective method to perform software quality assurance. The purpose of this study is to highlight what software businesses are currently doing in their SQA domain and investigate if an effective industrialised process for SQA in these small organisations can exist.

The methodology used to conduct this study is the subject of the next chapter.

3 CHAPTER 3 - THE UNDERPINNINGS OF INDUSTRIALISATION (OF SQA)

"Quality means doing it right when no one is looking." –Henry Ford

3.1 Introduction

The previous chapters set the basis for software quality assurance in small businesses and potential for applying industrialisation concepts into software quality assurance using the total quality management framework. Chapter 2 pointed out that, despite the concepts of software quality and industrialisation being well researched individually, industrialisation in the software quality assurance field is still immature and more so in small software development organizations. This chapter, however, introduces an approach to apply the industrial concepts to software quality assurance - to create a framework for the right quality that enables full transparency about the quality of products and alignment of risks, with the business needs to implement a holistic and integrated approach to quality.

3.2 Industrialisation of software quality

Drysdale (2022) eloquently states that enterprises in recent years have entered a new "industrial revolution" – the industrialisation of software creation, especially for code that resides in the cloud. Firstly, Drysdale (2022) states that running digitally at scale requires automation and that secondly the talent pool of software engineering practices in Australia is becoming harder to ignore. Software engineering teams are therefore pioneering standard processes and introducing discipline to the way software is created. He also states that underpinning all of this is the pursuit to achieve faster time-to-market, reduction of total cost of ownership and creation of an overall better experience for their customers. This describes the role of the software quality assurance process in an organisation hence an SQA process that needs to follow the same pursuit.

The most suitable way of evaluating and improving organisations through industrialisation is to successively implement the five dimensions of industrialisation: modularisation,

standardisation, specialisation, automation and continuous improvement (Wieczorek, Vos and Bons 2014), as also stated in section 2.7.2 in this document.

The highest leverage effect can be achieved by following the sequence of dimensions in the order mentioned, even if it is possible to work in parallel on selected issues of different dimensions for local improvements or to mitigate high priority risks. For example, automation of specific quality aspects is most efficient when modularisation and standardisation has already been implemented to a certain extent to avoid redundant work and maintenance. Additionally, this order gives a high degree of re-use of pre-existing components.

3.3 Industrialisation dimensions related to business layers

Clear goals from top management or owners of small businesses need to set the scope of actions required for the execution on quality at all layers of the business. The holistic approach of quality considers the SQA strategy at the strategic layer, the quality models and QA management at the planning layer and the quality requirements at the operational layer.

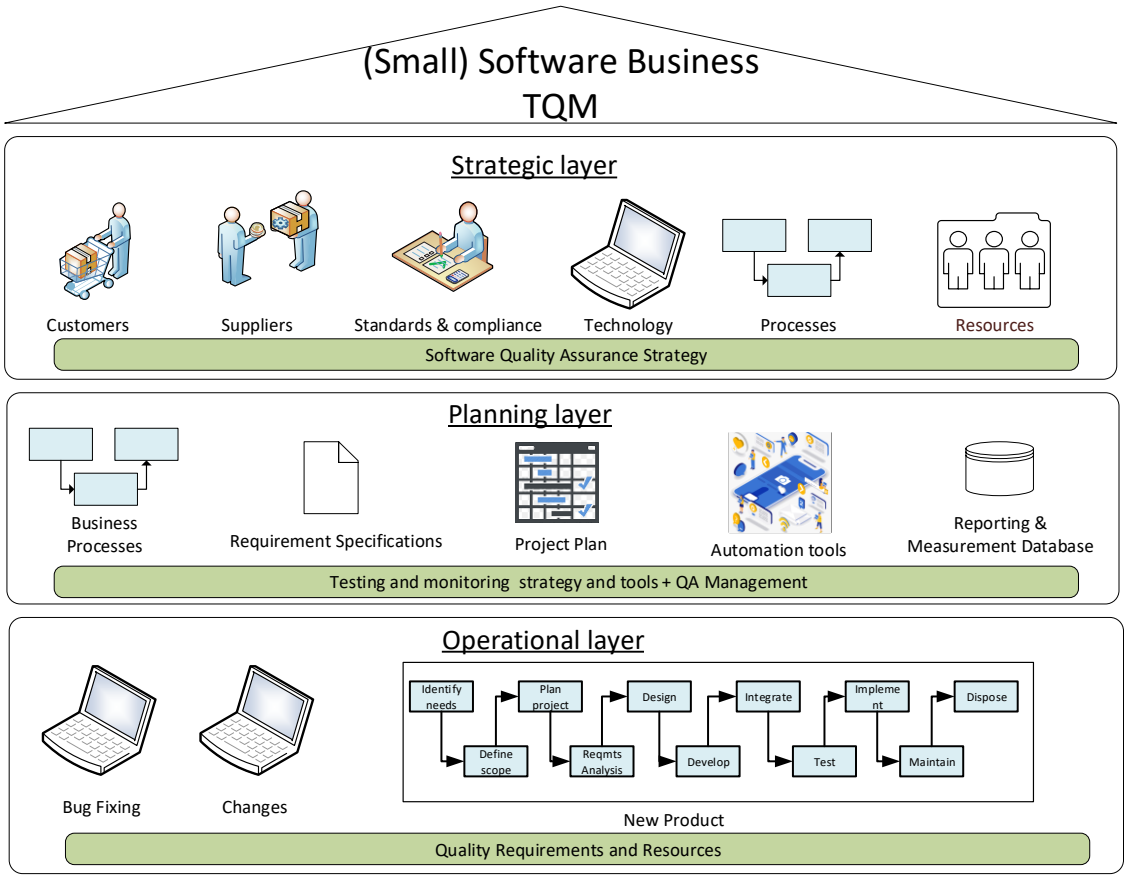


Figure 8 - Holistic view of Quality within business layers

Source: Adapted from (Wieczorek, Vos and Bons 2014))

For a holistic view of quality, total quality management (TQM), as stated in section 2.4 has entrenched a view that quality assurance activities be rooted in all layers of the business as shown in Figure 8. Ali and Johl (2022) further affirm that TQM practices from an Industry 4.0 perspective have positive and significant impact on financial, social and environment for small businesses. All actions taken for industrialisation, therefore, must be considered with respect to the impact on the business layers as well. These are the strategic, planning, and operational layers. These layers may exist successfully for small businesses too.

The ISO 9001 standard for quality management systems (9001: 2015), is an example of how a business, big or small may be managed successfully by using a structured system of business layers for managing their sustainability (as in Figure 8). According to Magodi, Daniyan and Mpofu (2022), the adoption of quality management systems such as ISO 9001 is recommended for operational efficiency, increased customer satisfaction and small business sustainability. The business layers (strategic, planning and operational) outlined in the ISO 9001 standard are very much in alignment with those highlighted by

Wieczorek, Vos and Bons (2014) for their proposed industrialised software quality assurance approach.

Industrialisation in software development, which is documented and studied by several researchers and authors such as (BITKOM 2010) and (Greenfield and Short 2003) and software quality assurance follow similar rules (Wieczorek, Vos and Bons 2014). In fact both flows are aligned but independently performed. Wieczorek, Vos and Bons (2014) assert that all activities taken for the five dimensions of industrialisation also must be considered with respect to their impact on the business layers of strategy, planning and operational. This section further discusses how the business layers relate to the industrialisation dimensions and how it may be applied in the context of the software quality assurance process and how the research questions for this study may be answered. Table 1 shows this relationship between the layers and the industrial dimensions.

Based on 9001: (2015) quality management standard, a TQM framework for small businesses (Kalpande, Gupta and Dandekar 2012) and matrix developed by Wieczorek, Vos and Bons (2014), the minimum requirements for the potential implementation of industrialisation of SQA at the different business layers have been filled into Table 1 in the form of keywords.

Table 1 - Industrial Dimensions – Business Layer Matrix (Source- Developed by Author)

		Business Layers		
		Strategy	**Planning**	**Operational**
Industrialisation Dimensions	Modularisation	SQA strategy Critical success factors KPI strategy	Requirements gathering	Testing strategy Intermediate artefacts Project specific processes
	Standardisation	Reference standards Quality budget	Right quality Project management Processes Benefits Constraints Reporting and measurement database	System structure SQA processes Templates Re usable – system Re-usable – testing Styling guides Standardised working environment
	Specialisation	Automation software	Who does SQA	Skill, training Quality gate
	Automation		Automation tools	Savings
	Continuous Improvement	Organisational structure Quality governance	KPI measure Quality Management	Measurement database Least effective Most effective

To read the table an example is given: For an organisation to achieve industrialisation of SQA of its standardisation dimension for its operational layer (the shaded cell in Table 1), the following criteria must exist:

- A system structure for the level of operations
- Existing SQA processes
- Templates for its repetitive activities
- Re-usable software testing components
- Re-usable components for system level testing and development
- Styling guides for software development
- Standardised working environment

These criteria indicate a level of readiness for industrialisation of a software quality assurance process that may be industrialised.

The *strategic layer* is where the software quality risk model is defined. The quality governance approach from a strategic perspective is defined and updated at this level. This layer then provides the quality risk model to the planning layer. The planning layer involves the quality governance approach from a planning point of view. Verification and validation assets of a project are provided at this layer. The operational layer defines the detailed quality requirements, conducts verification and validation activities, and provides feedback to the planning layer.

Modularisation is concerned with structuring and decomposing processes. To be able to structure and decompose workflows it is necessary to structure and decompose products to intermediate artefacts that can be worked on in different steps. Some aspects to be considered in modularisation for software quality assurance are:

- Testing strategies (as a whole and related test stages
- Intermediate artefacts (input and output)
- Software components like packages for test environments
- Test cases, test scripts, test results and reports
- Interfaces between SQA and change and release management.
- Interfaces between SQA and development processes

Standardisation provides the opportunity to adjust the individual approaches or activities of SQA to a common approach. This will avoid the intervention of new workflows every time a new project or task is started and will reduce the probability of defects. Standardisation assures the re-usability and interchangeability of products. Standardisation also reduces the number of processes that create the same output from the same input. Aspects to be considered in standardisation of SQA are:

- Verification and validation as a whole
- Analysis and improvement processes
- Standard test case portfolio
- Standard test scripts for functional testing

The *specialisation* dimension of industrialisation focuses on the skill and qualification of an individual within an organisation for the task they must perform. It covers to aspects: areas with no overlap and roles that are defined by subject matter experts. In small organisations, it is very unlikely that the resource pool will allow for specialists due to the limited size of the businesses.

Typically, some specialisation considerations in SQA are:

- Specialist verification and validation role
- Team building for specific roles

Automation always means replacing manual activities by automated ones. Automation is most efficient if functions, modules, and components can be re-used. Repetitive part of a process is ideally considered for automation. Automation can expand from individual activities to the automation of an entire process. The degree of automation implemented in an organisation depends on budget and time parameters, on skills of people and on the lifecycle of the current product. This makes it practical for application in small businesses software businesses. Aspects of automation that need consideration include:

- Analysis of processes with respect to stability and repeatability else automation is not feasible.
- Automation is a long-term approach so maintenance and operation of the automated system must be assured.
- Validation and verification assets are typically automated at the operational layer of the organisation.

Continuous improvement is not a value-added dimension. The previous 4 dimensions of industrialisation are based on business goals. Quality management ensures that these goals are achieved, and continuous improvement provides a way to regularly and systematically analyse the business, its processes and its products to ensure that the continuous improvements are achieved. Feedback and experiences need to be collected and analysed for sustainable decisions to be made to generate improvement activities. These are done typically by capturing performance data into database for measurable processes.

3.4 Conclusion

The purpose of this chapter was to explain the theoretical underpinning of industrialisation for this study. Industrialisation has been successful in many industries worldwide and has also delved into the realm of software development. This chapter has given a theoretical lens through which the quality assurance of software may be viewed. The importance of understanding how industrialisation can be applied to the software quality assurance industry is key to understanding how the data collected may be used to analyse the results of this study. The objective of the presented approach was to find a way by which the industrialisation dimensions may be utilised in small businesses. It is not the aim of this study to develop an industrialised approach but to investigate its effectiveness in small businesses.

The concepts explained in this chapter will help bring meanings to the results. The next chapter outlines the research methodology used in this study.

4 CHAPTER 4 – RESEARCH METHODOLOGY

"If debugging is the process of removing software bugs, then programming must be the process of putting them in." ~ Edsger Dijkstra

4.1 Introduction

In the literature reviewed thus far, only (Wieczorek, Vos & Bons, 2014:Chap 5) have defined an industrialised quality framework for software quality assurance. The notion that the software quality assurance process can be industrialised is dependent on how we define industrialisation as well as on the current state of SQA. The literature researched thus far do not indicate widespread implementation of industrialised SQA processes in companies, especially not in small software businesses.

To answer the stated research question below (as in section 1.4), it is not necessary to prove that the SQA industry is currently industrialised but to demonstrate that it can, in the future, be industrialised.

Can Industrialisation of the SQA process help increase the effectiveness of SQA implementation in small software businesses?

The outcome of this study will hence seek to conclude if small software businesses will benefit from an industrialised SQA process. This was be done by adoption of a research design and methodology served to develop an understanding of this phenomenon and attempt to address it.

This chapter discusses the philosophical assumptions, research approach and design, methodology and method for data collection and analysis method for this study.

4.2 Methodology

4.2.1 The research process

To arrive at a research conclusion, a research philosophy and a research design is required. A framework developed by Saunders, Lewis and Thornhill (2009), the research onion, in Figure 8, guided this research process, starting from the selection of the

research philosophy to the analysis of the data. This enabled the research to be designed and conducted in a structured and systematic way, by selecting a research philosophy, justifying a methodological choice, research strategy and time horizon and finally the data collection methods and data analysis.

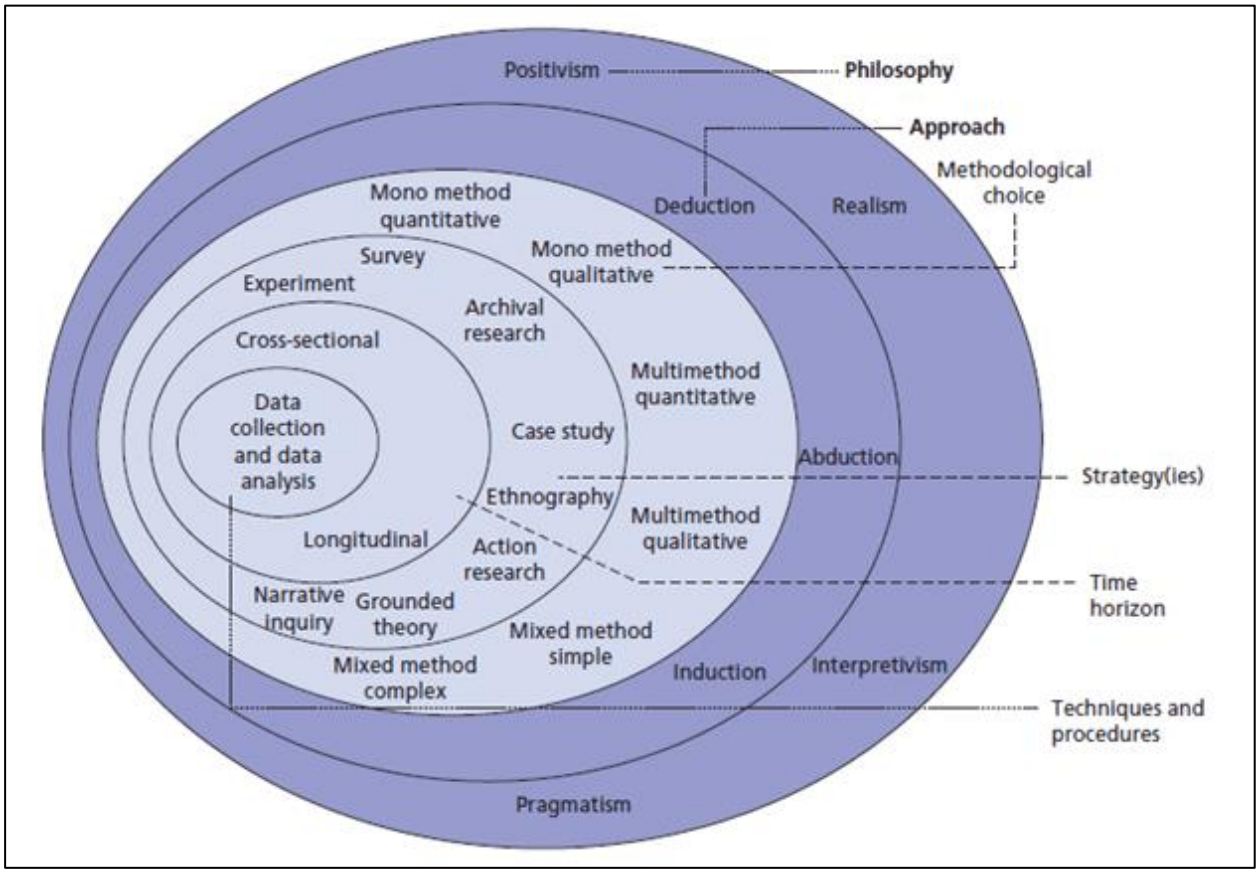


Figure 9 - The research onion

Source: Adapted from Saunders, Lewis and Thornhill, 2009

4.2.2 Research philosophy

A research philosophy is a set of beliefs and principles that guide the researcher's approach to the research process. It provides the researcher with a framework for understanding and interpreting the research phenomena and determines the researcher's perspective on how knowledge should be generated, and what methods should be used to gather data.

The literature review presented in chapter 2 highlighted that an industrialised SQA process does not exist in a small software business. The objective of this study is to

investigate the effectiveness of industrialising such an SQA process for small businesses by doing an in-depth investigation.

The positivist paradigm is a methodological philosophy where the methods of natural sciences apply to an observable social reality to produce law-like generalisations. It promises unambiguous and accurate knowledge (Pham 2018). Research studies using this phenomenon investigate the relationship between an independent variable and one or more dependent variables and are discovered by causal inferences as the results of experimental design. For the purposes of this research study, the positivism philosophy was not applicable since an industrialised SQA process under investigation does not currently exist.

Section 2.7 culminated in a discussion of the benefits of industrialisation of SQA on small businesses, but clearly in the absence of data, an exploration of the concept to gain further knowledge is necessary. The purpose of this research study is to create new, rich understandings and interpretations of the research question in its social world and its context. Research data wasl therefore needed to be collected based on what was meaningful to the social actors of the study. For these reasons, the interpretive philosophy was chosen as it aims to produce an understanding of the social context (small software businesses) of the phenomenon (SQA industrialisation) and the process whereby the phenomenon is influenced by the social context (the effectiveness of industrialisation of SQA in small businesses)(Rowlands 2005).

4.2.3 Research approach

The research strategy encompassed investigating companies that perform the SQA function to better understand the nature of the SQA problem. The task then involve analysing this information and formulating a theory (Creswell 2014). When making empirical observations about some phenomenon of interest and forming concepts and theories based on them, an inductive approach is employed. The deductive approach on the other hand, starts from a theory, derives and tests a hypothesis and then revises the theory (Woiceshyn and Daellenbach 2018). This research therefore uses the inductive approach, where according to (Saunders, Lewis and Thornhill 2009) a more informed decision about the research design can be determined. Since this research is concerned about understanding if industrialising SQA can be effective in small software organisations, it is more appropriate to undertake the research inductively.

Furthermore, the industrialisation of the SQA process was investigated regardless of whether the software development process has been industrialised, in other words, independent off the industrialisation of the software development process.

4.2.4 Research strategy

The choice of research strategy was guided by the research question and objective of this study, the extent of existing knowledge, the amount of time and other resources available as well as the researcher's philosophical underpinnings.

Saunders, Lewis and Thornhill (2009) point out that although the grounded theory strategy is often seen as the best example of an inductive approach, the research starts with a theoretical framework and the theory is further developed from data generated by a series of observations. Software development projects have varying timelines and can go on for several years on some projects. For this reason and the limited time available for completion of this study, grounded theory is not suitable. On the other hand, Woiceshyn and Daellenbach (2018) assert that, the ethnographic strategy, which is rooted firmly in the inductive approach, requires the researcher to be immersed in the social world being researched over a long period of time. Ethnographic strategy is also not a suitable choice for this study as it is not feasible for the research to observe SQA in progress during an entire software project.

The strategy, hence, involved understanding the components of industrialisation and then seeked out small businesses to see if they had these components or similar components to help them industrialise the SQA process. Asking existing actors in the software industry, together with secondary information provided by these actors assisted in arriving at an answer to the research question. This justified the use of exploratory method in this study. An exploratory study helped to understand the nature of the problem and to explore the topic at different levels (Casula, 2021). Exploratory research is conducted when the researcher has little, or no prior knowledge of the phenomenon being studied.

4.2.5 Research choice

Quantitative research focuses the relationship between variables through the testing of a hypothesis (Antwi, 2015). Since project timelines vary from project to project, it was not

practical to perform a quantitative study of the research questions in the timeframe of this research study. This was further emphasised by the fact that to measure the effectiveness of SQA the research study would have to be involved in several projects from its inception to measure the quantitative data which was outside the scope of this study. The timeframe of the research hence eliminated the use of the quantitative method as a research design choice. Qualitative research explores the phenomenon under study through the participation of the research participants (Kapoulas, 2012) in their natural surroundings and setting. Qualitative research focuses on the how and why of the issue at hand. The choice for this project was therefore a qualitative choice. According to (Jain 2021), interviews, instead of surveys, offer an opportunity for the researcher to ask probing questions. They were also found to be useful in getting a broader understanding of how and why certain things happen and what the opinions, motivations, interests, and feelings of the people involved in the research study. The chosen research method therefore used interviews as a data collection method.

Since the problem being studied in this research has not been deeply investigated before, both primary and secondary research methods was pursued. Primary research involved interviewing the participants of the research whereas secondary research involved reviewing the literature provided by the participants and online literature and data to arrive at an analytical conclusion.

4.2.6 Research techniques and time horizon

The study of the state of software quality assurance in a small software business was done at a particular point in time. Time constraints and the fact that software projects vary from project-to-project in duration and complexity, dismissed the use of a longitudinal study. Therefore, the use of a cross-sectional study was chosen.

4.3 Research Population Target

The target population in this study were software developers and quality assurance leaders in small software businesses. Employees, in privately owned small businesses in the software sector, were selected on their active participation, experience and knowledge in the software development and software quality assurance industry, thereby contributing a rich understanding of the software quality assurance process. Small

businesses with under 15 employees were pursued for this study. This limit placed very small enterprises (VSEs) under focus in this research study. The Institute of Information Technology Professionals of South Africa (IITPSA) was also contacted for a list of software businesses to engage in this research study.

The researcher has accumulated at least 15 years' experience in the field of Software Quality Assurance. This experience was instrumental in securing the participation of the research study participants. Saunders, Lewis and Thornhill (2009) indicate that research has shown that a researcher is more likely to gain access to participants when an existing contact is used. Using a network of established peers and colleagues from the industry, the researcher had gained the trust and willingness of participants to contribute to the study.

4.3.1 SoftwSampling and Sample Size

Saunders, Lewis and Thornhill (2009) and Guest, Bunce and Johnson (2006) suggest that when the aim of the research is to understand or investigate common themes within a fairly homogenous population that twelve in-depth semi structured interviews should be sufficient. This allows the group to be interviewed in detail to achieve the answers to the research questions. Saunders, Lewis and Thornhill (2009) also indicate that data saturation will likely be achieved at this stage. A comprehensive, in-depth interview guide for the semi-structured interviews was compiled for the purpose of this study (See appendix A)

In this research study the software developers and quality assurance individuals had similar exposure to work activities, often, wearing two 'hats' in small businesses due to their small and limited resource pool. It is common for roles and responsibilities to be shared in small businesses. Since one developer and one quality representative per organisation was necessary for this study, six organisations were planned for participation. At the time of writing this dissertation, the sixth organisation indicated that they were unable to participate due to an overload of work commitments. This further emphasized the difficulty for small businesses to spread their resources.

4.3.2 Participant recruitment

The recruitment procedure was done incrementally:

a) A list of potential candidates was acquired:

A list of small software businesses was identified from IITPSA (Information Technology Professionals of South Africa). A second list of potential participants was compiled from the researcher's network of software development companies.

b) Request to conduct interviews:

- Contact was made with the organisations (via email or telephonically) and their participation was requested through the management (the potential gatekeeper) of the business.
- Where the request was accepted, the appropriate software developers and software quality assurance personnel was identified by discussion with their management.
- Once identified, the participants were contacted for their consent to the interviewed.

This method of recruitment allowed for a positive relationship to develop with the individuals who granted access to the researcher and improved the credibility of the researcher and the research itself.

4.3.3 Medium to conduct interviews

The interviews were conducted in two ways:

- In-person for participants locally available (geographically). This was done at a mutually agreed location.
- Online via Microsoft Teams for participants not geographically located in the same area or if the participant preferred the online method.

All necessary Covid 19 regulations were considered for in-person interviews.

4.3.4 Inclusion and exclusion criteria of participants

Inclusion and exclusion criteria increase the likelihood of producing reliable and reproducible results. The following criteria was used:

For inclusion:

- Participants must have software development and/or software quality assurance in their current role.
- Participant organisations must be in the software development business.
- Small businesses to have fewer than 15 employees.
- English speaking participants were required to facilitate the interview (since the researcher is fluent in English)

For exclusion:

- Software corporates such as Google, Microsoft were to be excluded.
- Software vendors (these are businesses that sell software but do not develop the software themselves) were not part of the research study.

4.3.5 Obtaining informed consent

After the participant recruitment process had identified the potential research candidates, and the participant had expressed a willingness to participate in the research, a consent letter, detailing the agreement to participate in the study was issued for signature by the researcher and participant (Appendix B). Together with the letter of consent an information letter (Appendix C) was attached. The information letter informed the participant with relevant information about the research such as:

- Information about the researcher
- the research purpose,
- the procedure of the participation (interview in this case)
    - including permission to audio record the interview
    - duration of the interview (approximately 45-60 minutes)
- any risks or discomforts to the participant
- the right to withdraw from the research
- any benefits, remuneration and costs to the participant
    - Reports to be shared
    - No remuneration for the participant
    - And no costs to the participant
- confidentiality and anonymity of the participant
- storage of data
- contact information for any problems or queries

4.3.6 Interview time and duration

Interview session times were set up during weekdays during work hours or a time that was suitable for the participant. The interview duration was approximately 45 -60 minutes. Permission for the audio recording was sought from the participant (as stated on the consent form).

4.3.7 Participants anonymity

The information letter confirmed that the participant would remain anonymous, and their information would be kept confidential. The interview transcripts and audio recordings were stored in different locations and would only be linked together via a key or a code. This key would not be recorded in the same location as the interview data. Contextual data such as location of interview, date/time and participant information (business role, experience) was also recorded.

Both forms are attached to this document (Participant Consent letter and Participant Information letter) in Appendices B and C respectively.

4.3.8 Data Storage

The data was stored in a cloud-based storage. One folder was created for interviews, and one for documents received from the organisation and interview transcripts. All interview audio files have the naming convention Org 1-Participant 1, Org 1-Participant 2, Org 2–Participant 2, etc. All electronic data will be stored for 5 years and then deleted. All data stored physically will be destroyed.

4.3.9 Gatekeeper consent

Primary data for this research study was collected via interviews. This required physical access to individuals that work in businesses who most often have limited time and resources. To allow the researcher to undertake research it was necessary to contact the person that permitted access to these individuals. In most small businesses these are the business owners or managers. The gatekeepers in this research study were the owners or management of the intended participating businesses.

Two letters were developed:

- A letter seeking permission from the management of the participating business from research supervisor (on behalf of DUT) - see Appendix D
- A letter seeking permission from the researcher (student) of the participating business – see Appendix E

4.3.10 Participant's incentives

There were no financial or other incentives for the participants. This was stated on the Information letter that was given to the research participant together with the letter of consent.

4.4 Ethical Considerations

The lack of standardisation of semi-structured interviews (which was used in this research) may have led to concerns about the reliability, credibility and trustworthiness of the data collected. For this reason, the researcher took the following steps to ensure the quality of the data was not compromised (together with the letter of information for the participant):

The level of knowledge by the researcher on the research topic and the business participating in the research helped demonstrate credibility. The researcher has had several years of experience in the software quality assurance field, and this contributed to the topic under discussion. The participants were provided with the themes of the interview before the event took place. For in person interviews the location for the interview were asked for their preferences. The interviewee was assured that confidential information was not being sought from them. This ensured the trustworthiness of the research. There was also no mention of the other organisations participating in the research. Any concerns that the interviewee had were clarified before the interview commenced. Where concern regarding anonymity was highlighted, the researcher assured the interviewee that data will be stored anonymously, and the interview transcripts will have a code to relate the data and the transcripts. The participants were re-assured that the interview was fully voluntary, and they could withdraw participation at any time.

A request for ethical clearance of this research study was accepted by Durban University of Technology's Institutional Research and Ethics (IREC) committee (see Appendix F)

4.5 Data Collection Technique

Structured interviews are associated with the quantification of research data. This research study used semi-structured interviews instead as these are an appropriate instrument for qualitative research (Saunders, 2009). Semi-structured interviews allow questions to be asked on the themes of this research study (SQA, Software Development, Industrialisation, and small software business) as stated in section 2.1 and Figure 3. Semi-structured interviews also allow for open-ended questions, and this was helpful to find out more details about what the participants experience in the SQA environment in their small businesses. Semi-structured interviews were therefore appropriate for this exploratory study.

In pursuit of primary data via semi structured interviews, secondary data would also be collected. These were in the form of process flows, policies, procedures, and other documents used by the participants in their activities. Secondary data provide an additional source from which to answer the research questions (Saunders, 2009).

4.6 Data Collection

The interview questions were designed to enhance the understanding of the themes of the research study as outlined in section 2.1 and to answer the objective of the research questions and the associated sub-questions.

Research Question: Can Industrialisation of the SQA process help increase the effectiveness of SQA implementation in small software businesses?

Research sub-questions:

- What costs are incurred by the organisations to achieve their desired quality level?
- Do small software development businesses have a defined SQA process in place?
- How effective are the SQA activities in these organisations?

- How could the SQA process be industrialised in its application?
- Will this industrialisation improve the effectiveness of the SQA process?

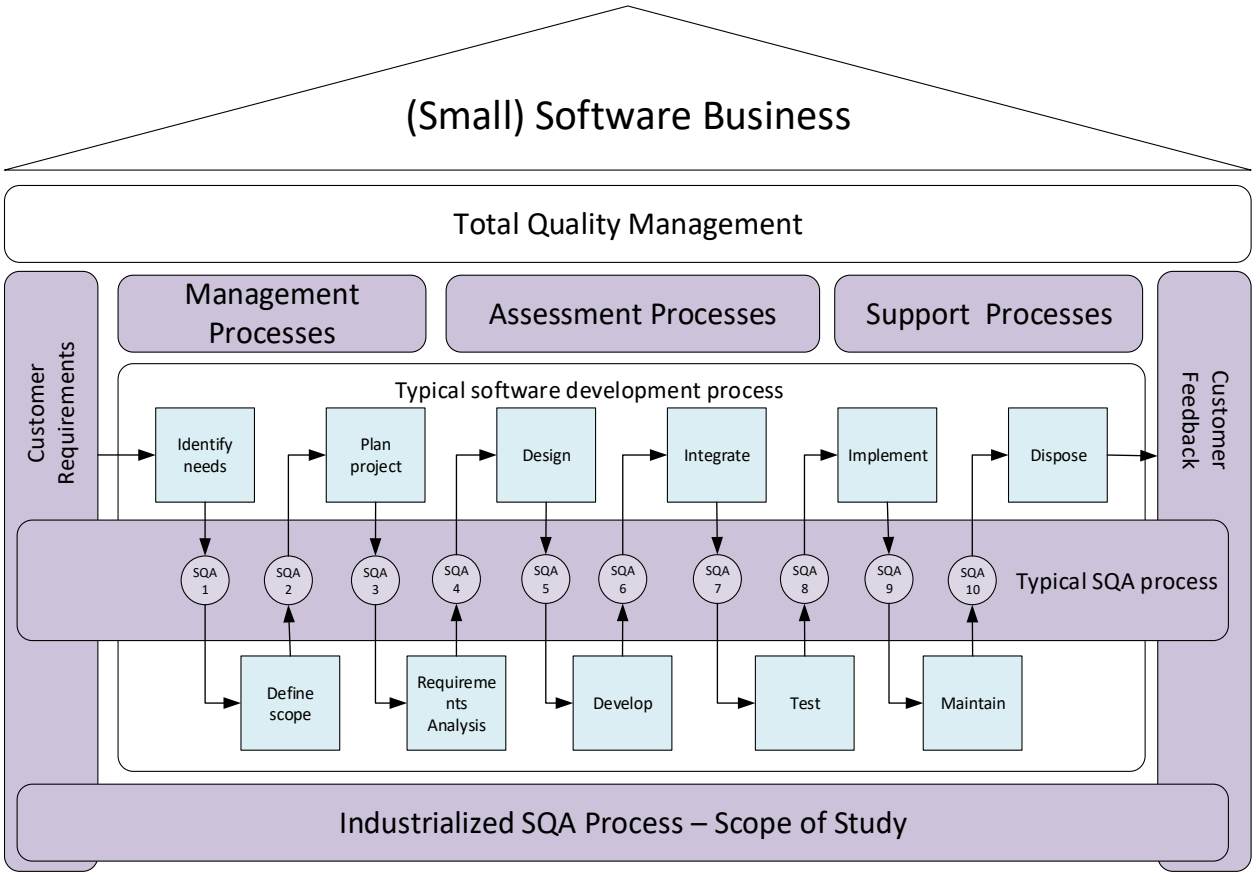


Figure 10 - The small business quality house (showing interview focus)

Source: adapted from Wieczorek, Vos and Bons, 2014

To further enhance the data collection via the interviews for an in-depth understanding of the topics, some questions were targeted for the SQA leader and others were created for the software developer. These questions were structured into sub- categories of business strategy, planning and operations. The sub-categories align to the business-wide TQM framework as introduced in section 1.5 and further reviewed in section 2.4. Typically, the SQA leader was positioned to answer the strategic and planning questions the developer could answer questions from the planning and operational point of view. Further building on the house of quality, in Figure 10, the areas highlighted in purple were the areas around which the interview questions were developed.

The population of the participating organizations for this study was small software development companies within the country. Based on the research questions, the

interview candidates were personnel in Software Quality Assurance leadership, and software developers within these companies.

The interview questions were the primary tool for collecting information from participants to answer the central research question. Themes that run through the data collected, which relate to the research questions in this study, were uncovered.

4.7 Data Collection Instrument

Table 2 below shows a list of questions used in the interviews with participants in the research study.

Table 2 - Interview questions

No	Interview Questions
	Section A **Strategic – How does the business landscape accommodate strategies for SQA, standards and regulations.**
1	Describe the strategies used for managing software quality in your organisation? *Probe: How does your organisational structure support this strategy*
2	How are the critical success factors for measuring software quality described? *Probe: Can you give an example?*
3	What strategy is in place to manage and measure these KPI? *Probe: How are these measurements recorded and tracked?*
4	What training and communication strategies are in place for the SQA? *Probe: Does this strategy include areas of technical training?*
5	What reference standards or international standards are used?
6	What strategies are least effective for managing your software quality?
7	What strategies are most effective for managing your software quality?
	Section B **Planning – How does the business plan, monitor and control the SQA strategy**
8	How does the organisation know whether products have the correct quality? *Probe: Does a reporting and measurement database exist?*
9	How is the project management methodology used in the organisation described? *Probe: Does this include project and product governance procedures?*
10	Are budgets specified for quality and risk management?
11	What processes, procedures and instructions are in place? *Probe: Are examples available please for sharing?*
12	What role does automation tools play in the organisation? *Probe: Can you give examples of this*
13	How would you describe the strategy for testing during the software development cycle?
14	What are the benefits and constraints in the SQA process?
	Section C **Operational – How are the plans for SQA executed?**
15	Describe the system structure for developing software? *Probe: Are standardised modules and styling guides used to develop software?*
16	How are SQA processes defined? *Probe: Are these processes defined by product type or project type?*
17	Who performs the Quality assurance function?
18	What templates are used for SQA?
19	What re-usable components available at a system level?
20	What re-usable components are available at testing stage?
21	What processes are used for requirements gathering?

Questions were categorized into areas of Strategy, Planning and Operational but differentiated by the 5 stages of industrialisation stated in section 2.4 (Modularisation, specialisation, standardisation, automation, and Continuous Improvement). Since this research study is based on the interpretivist philosophy, the researcher is concerned with understanding the meanings that participants ascribe to various phenomena (Saunders, Lewis and Thornhill 2009). To gain maximum effectiveness from the interview some questions had one or more probing question to enable the participant and researcher to build on the main answer.

An interview guide (in Appendix 6.1) was prepared for use at the interview. The guide included a few steps to explain the interview process to the participant prior to the commencement of the interview as well as the interview questions as detailed in Table 2.

4.8 Data Analysis

The interview questions were designed to explore answers to the central research question. The data collected during semi-structured interviews are at risk of reliability, bias and validity issues (Saunders, 2009). These challenges were taken into consideration when reporting the results and drawing conclusions.

Inductive data analysis allows for the development of an answer to the research question in this study. According to Ngulube (2015) data analysis using themes is considered to be the foundational approach to qualitative data. Thematic analysis, as it is called, is a method for identifying themes and patterns in a dataset in relation to the research question of the study. The conclusions are then drawn based on observations from the transformed data.

The steps outlined by Isaacs (2014), Ngulube (2015) and Aronson (1995) were used as guideline for this study. The steps are as follows:

- Read and re-read the interview transcripts and listen to the audio recordings to obtain a sense of what was being said while consciously looking out for what stands out as being part of the answer to the question (Isaacs 2014). This may require going back and forth between the study aim and theoretical framework.

- Identify all data and relate to the already classified patterns (Aronson 1995). In this study high level themes were introduced in the literature review (Chapter 2)
- Assign codes to chunks of data if necessary to find categories or themes in the data.
- Combine related patterns into sub-themes. According to (Aronson 1995) these patterns are drawn from conversations or vocabulary, meanings or feelings.
- Link the different themes or categories to develop a logical explanation for the answer to the research question. This is done by reading the related literature and allows the researcher to make inferences about the interview.

The computer aided qualitative data analysis tool, Nvivo, was used to do the analysis of the data collected from the semi-structured interviews. Chapter 5, section 5.3.2, presents a detailed procedure for thematic analysis of qualitative data.

4.9 Conclusion

This chapter outlined the research methodology, approaches, and ethical approaches to collecting and analysing data for this study. A qualitative study, using the interpretivist philosophy, an inductive approach and exploratory strategy by utilising semi-structured interviews to collect data was employed. Five small software businesses were used as participants. The next chapter, Chapter 5 – Analysis and results will present the findings of the study.

5 CHAPTER 5 -ANALYSIS AND RESULTS

"It's hard enough to find an error in your code when you're looking for it; it's even harder when you've assumed your code is error-free." ~ Steve McConnell

5.1 Introduction

This study provided valuable insights into the capability of small software companies to industrialise their SQA processes. Specific insights addressing their current strategies and structures and behaviours uncovered in the study revealed whether the companies could potentially have an effective industrialised SQA to improve their sustainability.

Chapter 3 discussed the research design, methodology, data collection and analysis. Data was collected, using semi-structured interviews. The interviews were conducted in small software businesses with fewer than 15 employees. As mentioned in section 3.3.1, in small businesses, employees often share roles due to their limited resource pool and therefore the quality representative and software developers in some of the interviews conducted were done with the same individual. The entire set of questions (quality representative + developer) were then directed to this individual.

Five companies participated in this study. In section 4.3.1 six companies were deemed sufficient but the sixth company became unavailable for interviews in the timeframe of this study due to their business constraints – once again highlighting the difficulty of small businesses to effectively provide resources for their commitments. Effectively, the full set of interview questions were asked to five participating small businesses. Some interviews were done with participants individually (that is, one developer and one QA representative), but some were done with both participants in one sitting of the interview. This allowed some participating businesses to minimize the disruption to their business operations.

The research questions that guided this qualitative study were geared towards addressing the problem statement of investigating the effectiveness of industrialising SQA in small businesses. This study included five research questions:

RQ1:

What costs are incurred by the organisations to achieve their desired quality level?

RQ2:

Do small software development businesses have a defined SQA process in place?

RQ3:

How effective are the SQA activities in these organisations?

RQ4:

How could the SQA process be industrialised in its application?

RQ5:

Will this industrialisation improve the effectiveness of the SQA process?

This chapter presents the findings from this research study. The chapter begins with a presentation of the descriptive findings (section 5.2), including participant demographics. The procedures for data analysis follows (section 5.3). Section 5.4 provides the information to interpret the data and then the results are presented (section 5.5).

5.2 Descriptive findings

This section provides a narrative of the characteristics of the data as well as a description of the type and amount of data collected in the study. A purposeful sample of five companies were selected. Participants included those who met the following criteria:

- Worked in a small software business as a software developer.
- Worked in a small software business as a software quality assurance representative.
- Or worked in a small software business as both a developer and software quality assurance representative.
- Participants were willing to participate in an audio recorded semi-structured interviews.

The owners or CEOs (gatekeepers) of the participating organisations were approached initially for permission to use their organisations in the study. All interview participants were thereafter identified by the gatekeepers of the respective organisations. Contact was then made with interview participants to set up meeting times and settings that were suitable for the participants. A total of eight participants were interviewed, who were in the position to answer the full set of twenty-one research questions per participating organisation. Table 3 describes the organisation and participant demographics, in terms of organisation size and participants role in the organisation.

Table 3 – Organisation and Participant Demographics

Organisation code	**Company size (# employees)**	**Participant**	**Participant Role**
Org1	< 5	Participant 1 Participant 2	SQA representative Developer
Org2	<12	Participant 1	SQA representative + Developer
Org3	<6	Participant 1 Participant 2	SQA representative Developer
Org4	2	Participant 1	SQA representative + Developer
Org5	<5	Participant 1 Participant 2	SQA representative Developer

5.2.1 Settings

The study was conducted with organisations based in two provinces, KwaZulu Natal and Western Cape, in South Africa. The interview settings were conducted in-person and online. The participating organisations are all privately owned software development organisations who develop computer-based software and not firmware. The online interviews were conducted using the Microsoft Teams collaboration software. All interviews were audio recorded. Audio recordings were done using the Teams meeting recording functionality and a backup recording was done using a voice recording application on the computer. Table 4 details the settings of the interviews. The owners or CEOs (gatekeepers) of the organisations were approached initially for permission to

use their organisations in the study. All interview participants were thereafter identified by the gatekeepers of the organisations contacted.

Table 4 - Interview Settings

Org	Province	Interview setting
Org1	KwaZulu Natal	In person
Org2	Western Cape	Online
Org3	Western Cape	Online
Org4	KwaZulu Natal	Online
Org5	KwaZulu Natal	Online

5.2.2 Data Sources

The primary data source for this study was the data from the unstructured interviews.

Secondary data was requested from the participants. Only two organisations (Org 1 and Org3) were in possession of and able to share their data. This data included procedures and process flows of their SQA process (Appendix 7.11). These documents were used to look for information related to the research questions and in some cases supported the responses to the interview questions. Analysing the data from this source was important because it provided additional details to answer the research questions.

5.3 Data Analysis Procedures

The purpose of this qualitative study was to investigate the effectiveness of industrialising the SQA process in small software businesses. The phenomenon of industrialisation of the SQA, particularly in small businesses was explored and examined through the lenses and experiences of the study's participants. Data gathered from interviews with participants and some secondary documents obtained from participants provided insight and understanding of the phenomenon under study. The data collected was organised, prepared, and analysed to answer the five research questions laid out in the study.

For this qualitative study, inductive thematic analysis was used, as suggested by Braun and Clarke (2006), to analyse data from semi-structured interviews. Themes were generated inductively from the raw data. The themes identified were strongly linked to the data as higher order codes were identified to help organise the data without trying to

fit them into any pre-existing frame or bias. Sub-themes were also developed without trying to fit them into pre-existing coding framework. Nvivo and hard copies of the transcript were used to code data. The data analysis consisted of organising and preparing the data for analysis, followed by the analysis.

The audio recordings of the interviews were transcribed into text files using Microsoft Words transcription function. The correctness of the transcripts was verified by listening to the audio and confirming the text files conversion.

The interview questions were designed to investigate whether small businesses had the foundational elements of quality governance according to Total Quality Management (TQM) principles and industrialisation dimensions, which could potentially pave the way for the industrialization of their processes. These questions were formulated by identifying aspects of approaches that indicated the presence of industrialization prerequisites. The interview sections commenced with a broad introduction to the topic, followed by a structured exploration of industrialisation concepts as introduced in section 2.7.2.

5.3.1 Preparing the data

All interviews were audio recorded. Microsoft's transcription function was used to transcribe the audio files. Transcripts were checked and re-checked against the audio files for correctness. The data was at first hand-coded and was then uploaded into Nvivo for coding and analysis. According to Braun and Clarke (2006) , thematic analysis is a process through which patterns or themes are identified within a qualitative data set with the goal of addressing the research questions by interpreting and making sense of the data.

5.3.2 Analysing the data

A six-step process of thematic analysis to analyse semi-structured interview transcripts was followed as outlined by Braun and Clarke (2006). The data was analysed in two stages: manual coding of the data and the use of computer assisted qualitative data analysis software (CAQDAS), a software tool called Nvivo. Nvivo was used to organize the qualitative data.

In the first stage of manual coding, Microsoft Word was used to comment and generate initial codes. Codes refer to words or textual data salient to the research questions. The codes generated through manual coding were descriptive codes. These are brief descriptive titles to each coded passage. When the process of manual coding was complete the interview transcripts were uploaded into Nvivo for further analysis. Nvivo version 12 was used to assist with the thematic analysis of this study.

The second step of the thematic analysis is the initial coding process. Although coding was done manually, at this stage the coding process involved organising all data based on the research questions. Coded passages were pulled out form the manual coding and captured into Nvivo, in this way maintaining the structure of organisation of the data by the research question. All the transcripts were then coded into Nvivo ensuring pertinent interview questions related back to the research questions. This process yielded a list of codes that were then used further in the analysis.

Table 5 below presents an example of how the coding of passages of text occurred.

Table 5 - Coding of Raw Data

Raw Data	Code
“We allocate 5% of the total budget to quality, but. It's way more than for that, right? No, we definitely underfunded it” – (Org 3)	Budget for quality
“So, but them being in the loop all the time, that's how we actually ensure that the quality of the software is good, and we achieve those.” (Org 1)	Customers tell us about our quality
We did attempt to track that KPIs last year. But then it fell off the wagon again. But yeah, well, we do track the number bug fix requests that we get from our OPS team (Org 5)	Do not use KPIs
Yeah, that's a difficult one in a small company. Yeah, we did try to get people to sort of give us a spec or something. And we do try to have like a 30-minute design session with all the stakeholders. (Org 5)	Requirements gathering a challenge
We've got automated tests run for stress testing or load testing our units for the DCU (Org 1)	Testing tools

Step three in thematic analysis according to Braun and Clarke (2006), is to examine all the codes for themes. This is done to reduce the data further into larger categories of related codes with descriptive titles. In this way a hierarchy of codes, sub-themes and themes started.

When this was completed, step 4 allowed for review of these codes and categories. Braun and Clarke (2006), recommended that doing a review as this step guaranteed that all categories generated to this point in the analysis addressed one of the five research questions. Further larger categories found were also given a descriptive title at this stage. Appendix H shows a screen capture of the review and re-coding steps as performed in Nvivo.

Braun and Clarke (2006), encourage the final refinement of themes in step 5. In this way all themes and categories were placed in a thematic structure that told a comprehensive story.

Finally, in step 6 the results and findings were presented. Section 5.5 describes the results of the study.

5.4 Interpretation of data

This section describes how the findings of this study relate to existing theories or concepts in the industrialisation field. This demonstrates how the data interpretations contribute to or challenge the existing body of knowledge.

ISO 9001: (2015) defines the effectiveness of a process as:

“The extent to which planned activities are realized and planned results are achieved”.

Effectiveness is measured by comparing the target or desired state against the actual state of an activity or process. The effectiveness of industrialisation of the SQA process in small businesses may be interpreted by relating the findings of this research study to the fundamental dimensions of industrialisation. These dimensions have evolved over the past 100 years form work done by Frederic Taylor (Kambhampati 2017) in the Principles of Scientific Management originally published in 1911, and by Edward Deming, Joseph Juran, and Kaoru Ishikawa in the mid-20th century, for their work in quality management (Watson and DeYong 2010). The main principles of industrial engineering are deduced form the principles of scientific management (Kambhampati 2017).

The dimensions of industrialization in the context of processes and systems, namely modularization, standardization, specialization, automation, and continuous improvement, have evolved from principles in various fields such as manufacturing, quality management, and business administration.

As stated in section 2.7.2, knowledge about how the industrialisation dimensions (modularisation, standardisation, specialisation, automation, and continuous improvement), relate to strategic, planning, and operational layers of the participating organisations were extracted from the participants via the interview questions. This section sets out an interpretation of the research data as it relates to the research questions. Table 6 (an expanded version of Table 1) shows this relationship (relevant RQ shown in bottom right hand corner of the cells) and the how they contribute to the answer of this research study.

The research questions to be answered are:

- RQ1: What costs are incurred by the organisations to achieve their desired quality level?

- RQ2: Do small software development businesses have a defined SQA process in place?
- RQ3: How effective are the SQA activities in these organisations?
- RQ4: How could the SQA process be industrialised in its application?
- RQ5: Will this industrialisation improve the effectiveness of the SQA process?

Table 6 - Industrialisation Dimension - Business Layer Matrix (with Research Questions)

Source: Developed by Author

		Business Layers		
		Strategy	**Planning**	**Operational**
Industrialisation Dimensions	Modularisation	SQA strategy Critical success factors KPI strategy RQ4	Requirements gathering RQ2	Testing strategy Intermediate artefacts Project specific processes RQ4
	Standardisation	Reference standards Quality budget RQ1	Right quality Project management Processes Benefits Constraints Reporting and measurement database RQ2	System structure SQA processes Templates Re usable – system Re-usable – testing Styling guides Standardised working environment RQ4
	Specialisation	Automation software RQ4	Who does SQA RQ4	Skill, training Quality gate RQ3
	Automation		Automation tools RQ4	Savings RQ3
	Continuous Improvement	Organisational structure Quality governance RQ4	KPI measure Quality Management RQ1	Measurement database Least effective Most effective RQ3

Firstly, to understand the effectiveness of industrialisation it was necessary to understand what the businesses considered their criteria for success. It was also necessary to

understand the quality costs incurred. This information would lead to the answers for RQ1. Table 6 shows the cell in the Industrialisation dimensions-business layer matrix where emerging themes for answering RQ1 could be found.

Secondly, the investigation into the research question needed to uncover whether small businesses have an existing SQA process in place. This would provide a starting point for small businesses to effectively implement an industrialised scenario. Hence the need for answers to RQ2.

RQ3 searches for the effectiveness of the existing processes in the participating businesses. The answers to this question would allow the researchers to further determine the potential for industrialisation of SQA in small businesses.

Moving forward, Research Question 4 (RQ4) delves into investigating how small businesses may apply their existing processes to enhance their sustainability through industrialisation.

Finally, RQ5 contributes to answering the overall research problem to search for the effectiveness of industrialising the SQA process in small businesses.

The completion of Table 6 involved populating it with the research questions that would be answered through the amalgamation of the dimensions of industrialization and the layers within the business structure.

5.5 Results

This section presents the results of the analysis of data collected through the interviews and secondary information. The participating organisations were assigned codes, Org 1 to Org 5. The presentation of the data analysed in this study is organised by research question as examined through the themes and sub-themes as derived from the data. The data analysis was continued until no new themes emerged. Findings generated through the categories, themes and sub-themes embedded in the data answered the five research questions in this study.

The purpose of this qualitative study was to explore the effectiveness of the SQA process in small software development organisations. The analysis of data collected in this study revealed that small software development businesses find difficulty in setting targets and measuring their costs incurred to achieve a desired level of quality. They do have processes in place for some elements of their quality assurance process. Some of these processes may be industrialised, especially their testing and bug tracking processes. If this is done, they will see an increased effectiveness of their SQA process. Data gathered from interviews and some organisational documents from two businesses were coded and analysed to answer the research questions. Five major themes and 21 sub-themes emerged from the data and the findings from these were applied to the five research questions.

Table 7 shows the themes and subthemes extracted from the research and how they aligned to the research questions.

Table 7 - Thematic Structure of Study Results Aligned to Research Questions

RQ	Theme	Subtheme
1	Quality costs in small businesses are not measured	1a. Regulatory standards used for compliance but not governance. 1b. Setting and measuring KPIs is difficult. 1c. Small businesses have no budget or KPI for quality cost tracking
2	Small businesses have some elements of an SQA process	2a. Existing processes have some benefits and constraints. 2b. Processes, procedures and instructions somewhat used but not much is documented. 2c. Requirements are gathered by constant communication with customers. 2d. SQA role played by the developer in most cases. 2e. The correct quality mainly dependent on the customers satisfaction at handover
3	SQA in small businesses work but effectiveness is not measured	3a. Effectiveness of strategies are not evaluated in small businesses. 3b. No strategy for training and Communication 3c. Some strategies in small businesses are effective processes. 3d. ISO 9001 Quality Management System works well
4	SQA Industrialisation is best applied to testing phase in small businesses	4a. Automation tools used in development, testing and logging. 4b. Processes not mature. 4c. SQA management strategies vary. 4d. System level and testing re-usables. 4e. Start with customer requirements. 4f. Testing strategy involves developers and management
5	Industrialisation will improve effectiveness of SQA	5a. Critical Success Factors are customer based. 5b. Same quality standards for everyone 5c. Small businesses meet customer requirements. 5d. Defect tracking and management system are good starts. 5e. Communication and training are important

Word clouds visually represent text data by displaying words in different sizes based on their frequency or importance within the given text. They provide a quick and intuitive way to identify patterns, trends, and prominent themes within a body of text. Figure 10 shows a word cloud developed after the coding analysis was completed. Words such as 'software', 'quality', 'one' and 'process' emerge as the most used words during the interviews. These terms suggest that the data centres around existing processes in software quality in the participating organisations. While the word cloud offers insights into the primary themes, it is important to note that the word industrialisation did not appear. This relates to the discussion focusing on existing processes which were not industrialised. The analysis of the data would lead to a conclusion in the possibility and effectiveness of industrialisation.

Figure 11 - Theme word cloud

Source – Generated by Nvivo with research data

5.5.1 Theme 1: Quality Costs in small businesses are not measured

Theme 1 addressed the RQ1: What costs are incurred by the organisations to achieve their desired quality level?

There is no doubt that good software quality takes time and money to achieve. The lack of quality also costs money to both the users and the developers. Maxim and Kessentini (2016) state that the cost of quality includes the cost incurred by all quality-related activities pursued by the development team plus any support and maintenance costs that can be related to quality. The importance of quality costs is further highlighted by Shrestha (2016), who state that the costs of quality should be identified, measured, analysed and controlled to meet customer's requirements at the lowest cost. Theme 1 was defined by an organisations ability to measure their costs to achieve their specified quality levels for the software they develop. Table 8 presents a description of subthemes for Theme 1.

Table 8 - Description of Theme 1 Subthemes

Subtheme	Description
1a. Regulatory standards used for compliance but not for governance	Small businesses use standards to ensure that their products meet customer's requirements for compliance but not for governing their SQA and development processes.
1b. Setting and measuring KPIs is difficult	KPIs, which lead to metrics are difficult to set and measure in small software businesses.
1c. Small businesses have no budget or KPI for quality cost tracking	Budgets for quality cost tracking and risk are not set therefore the cost of assuring software quality is unknown.

Subtheme 1a – Regulatory standards used for compliance rather than governance.

Software quality standards and models provide a basis for specifying quality requirements and assessing quality of software (Miguel, Mauricio and Rodríguez 2014). Quality standards are helpful to set quality goals to support quality management for a software product. Process governance in software development organisations ensure that a software project aligns with the strategic business goals and compliance with external regulations. It offers a framework for achieving measurable progress towards these goals. Of the five participants, only one participating business (Org 1) had a process management governance system – the international quality management system (ISO 9001:2015) in place. Secondary data from Org 1 supported their statements regarding their management system. They presented a flowchart of their software development process as developed for their quality manual. They, however, indicated their difficulty in accurately measuring and setting KPIs for software quality.

Subtheme 1b – Setting and measuring metrics in small software businesses is difficult.

Most small software businesses do not keep metrics. As stated by (Pusatli and Misra 2011), expressing software quality by numbers is still a challenge. These authors also say that software quality between the developer and the customer needs to be

acknowledged and should be promoted against the increasing development costs due to additional programming costs and effort. This is very clearly seen by the analysis of the interview data. Org 5 clearly expressed their difficulty as stated below:

"We did attempt to track that KPIs last year. But then it fell off the wagon again"

Another participating organisation said that their customers tell them about their quality.

"We use the user! It's actually a very good question. If I can go first. But I think the problem that small companies have is you don't normally have the privilege of having a QA Team as such."

Participants also stated that resource constraints left them in a position where individual developers had to "double up in multiple roles in small organisations to do the testing".

Subtheme 1c - Small businesses have no budget or KPI for quality cost tracking.

All but one participating organisation said that they had set a budget for quality costs. Org 3 said that they "allocate 5% of their total budget to quality, ……but we definitely underbudgeted". Whereas Org 2 stated that when quality becomes a problem then "it's all hands on deck and it's just about let's just resolve the issue". They stated that they spent long hours and early mornings to resolve issues, but these hours are not tracked in terms of quality costs. Org 5 said that they thought tracking quality costs were for larger 'development' houses whereas they worked on billable hours. Org 1 were not sure if there was a budget for quality or that quality costs were tracked. Org4, vehemently said that "I haven't done this" when asked if he tracked quality costs.

5.5.2 Theme 2: Small businesses have some elements of an SQA process

Theme 2 addressed research question RQ 2: Do small software development businesses have a defined software quality process in place?

The fear of risking resources and not having anything in return cause small software businesses to be reluctant to implement processes and standards for software quality assurance (Pusatli and Misra 2011). It is understandable that organisations need to select an appropriate approach to automation tools, and they need to implement a plan

that best suits the improvement of their processes. Small businesses can develop agile methods very quickly as people are accessible and are flexible to switch roles (Gonen and Sawant 2020). Theme 2 was defined by the participating organisations' existing state of software quality assurance and their process capability. Table 9 describes the subthemes that emerged from the analysis.

Table 9 - Description of Theme 2 Subthemes

Subtheme	Description
2a. Existing processes have some benefits and constraints	Existing SQA processes in small businesses benefits the testing phase however more extensive testing is still needed
2b. Processes, procedures and instructions somewhat used but not much is documented	Small businesses use agile processes during some parts of their lifecycle and some processes to track and debug defects during testing.
2c. Requirements are gathered by constant communication with customers	Constant communication takes place with the customer, but the requirements are rarely documented
2d. SQA role played by the developer in most cases	The SQA role is carried out by the developer in small software businesses.
2e. The correct quality mainly dependent on the customers satisfaction at handover	Small software businesses use their customer to determine their quality levels

Subtheme 2a – Existing SQA processes in small businesses benefits the testing phase however more extensive testing is still needed.

All the research participants indicated that testing was a huge part of their informal process. As mentioned in section 5.5.1 subtheme 1a, Org 1 was the only participant that had a formal process documented. They presented their process flow as secondary information (see Appendix K). Org 5 went as far as to say that they did spend time on testing but their documentation for testing was non-existent:

"I would say the constraints or the downfall of what we do and how we do it is probably the lack of documentation. Because in a good test environment, you've obviously got documentation to test against, whereas in our case that that doesn't exist. Benefits quicker, quicker to production."

This implies that requirements specifications that are typically used to generate test cases are not available, and that the product may not be optimally tested.

This was further emphasised by what Org 1 believed to be their constraints. They felt that their test data was “too clean”, and this test data did not identify potential defects early enough.

Subtheme 2b – Small businesses use agile processes during some parts of their lifecycle and some processes to track and debug defects during testing.

The development methodology most used by software development in small software development organisations is the agile method. In saying that, it was noted that agile was used in some parts of the development (and subsequently), their SQA process. This was clearly stated by Org 3, who said:

“So we we're very agile at the start of the project, we very agile in certain developments, right, like the wire frames and so on because they are very easy to change. But then once we've got sign off on the wire frames and we approve the wireframes then the developers, they, you know, they've got to go through the development execution lifecycle there, right? So we use, basically I'd say like 2 methodologies right? So agile at the beginning, but then a very scrum based methodology in the actual development team.”

Org 1, the organisation who benefitted from using ISO 9001 in their organisation also indicated that they were able to use project management tools such as project plans to manage their team:

“So we have a project plan that we use. Okay. Uh, for checking of the time for the projects. It also gives us a, overview of when roughly the project will end so we can let the customer know that they can have their software solution at that point in time. Okay. Uh, so we, we try and work off the plan Yeah. So, to ensure that the project's on track as well.”

Subtheme 2c – Constant communication takes place with the customer, but the requirements are rarely documented.

Requirements needed from the customer to develop a product are not documented formally by small software development businesses. This affects the adequacy of testing on a software product, ultimately affecting its quality. Org 4 said exactly this in his response to a question regarding the documentation of customer requirements:

"I maybe should write this down, but I haven't written it down. So what I do is I after that initial meeting and what I told you like analysing the company's business and what they do. I then send an e-mail or an A message for all their website requirements after knowing what they would want, for example."

On the other hand, some organisations (Org 1 and Org 3) capture their customers' requirements into a requirement specification document but still struggle to give sufficient attention to developing test cases (test cases being an important testing artefact) during the testing phase. Org 2 spent more time on getting the requirements from customers and said that they attended a "user conference every year" and that they "sort of get our user requirements" from there.

Subtheme 2d – The SQA role is carried out by the developer in small software businesses

Org 5 said in no uncertain terms – "we use the user to test". They highlighted the fact that in small software businesses the developers wear many hats. They said:

"But I think the problem that small companies have is you don't normally have the privilege of having a QA Team as such. So the challenge often lies with developers and people almost have to double up in multiple roles in small organization to do testing.

The organisation that had ISO 9001 implemented, Org 1, shared that they had a junior developer that they used occasionally to do their testing. This provided a path for 'external' testing and an opportunity to train the junior developer. Their developer would do the testing after coding and then handed over the software to the junior developer. In another participating organisation's case (org 5), they used their owner to test since he had extensive experience in the products usage domain. The developers praised the owner's ability to identify defects: *"… once it is done, it goes into a staging environment and then normally if it's quite technical, then someone like Dennis will test, he's actually a brilliant, brilliant tester. So he's our tester then",* they said. This further raises the 'many hats' scenario that small businesses endure to remain sustainable and effective.

Subtheme 2e – Small software businesses use their customer to determine their quality levels.

Quality needs to be embedded at all layers of the business, namely, strategic, planning and operational. This together with balancing the cost of quality and cost of risk (damage) leads to the notion of the correct quality (Wieczorek, Vos and Bons 2014). Small businesses, however, argue that satisfaction of customers is their primary priority, and although quality standards and frameworks may achieve this goal the customer can be satisfied with responses to their issues (Pusatli and Misra 2011). This is very evident in the responses from the participants. Org 1 states it clearly:

"...so currently we work of the customer being happy. Uh, that's how we know that our software is of good quality".

5.5.3 Theme 3: Effectiveness of the SQA activities

Gonen and Sawant (2020) have undoubtedly said that although automation helps industries, it is difficult to implement. They also clearly state the small businesses cannot afford to buy and maintain automation tools that support the agile process, but it does give the team some flexibility. Meeting the customers' expectation was an important part of all participant's goals. Table 10 presents the sub-themes for answering RQ3:

How effective are the SQA activities in these organisations?

Subtheme 3a – Effectiveness of strategies are not evaluated in small businesses

The participating organisations found that formal evaluation of the effectiveness of their SQA processes was not practical, instead they were able to give some sense of their effectiveness to satisfy their customers. Org 3 recognised that their quality levels 'differed' between different employees - saying that "the deployment team and the development team currently don't have the same quality standards". Org 5 believe that they are effective as they do not create new architecture for every project – and a lot of their work is "just adding forms and functionality to the existing system", thereby reducing the risk of defects.

Table 10 - Description of Theme 3 Subthemes

Subtheme	Description
3a. Effectiveness of strategies are not evaluated in small businesses	Ongoing evidence of project performance, benefits realisation and achievement of organisational goals are not measured.
3b. No strategy for training and Communication	Small businesses are better at communication between employees and customer although no formal processes are defined
3c. Some processes in small software businesses are effective	Small businesses have found effective strategies and have embraced its usage and improved on it.
3d. ISO 9001 Management System works well	Employing a management system is evident that goals are achieved effectively

Subtheme 3b – No strategy for training and Communication but constant contact with the customer

All participants acknowledged that they did not have a training strategy. Org1, said that they would offer training to the staff on a "as needed" basis and that their ISO 9001 quality management process provided sufficient guidance for the employee. Org 3, however, acknowledged that it was difficult to integrate new employees into their business without a training strategy. Org 5, without mincing words, said that they could not do much else but actual software development:

"…in terms of strategies for quality assurance, no we don't have. Especially because as a small company, as a small Dev team, we must do everything software related, so in terms of the software, there's many nuts and bolts that puts that all together"

Subtheme 3c – Some processes in small software businesses are effective

Even though the organisations said that they were not able to do much in terms or measuring and evaluating their SQA processes effectiveness to meet their goals, they also said there were some positives aspects to the way they worked. For example, Org 5 said that their most effective strategy was spending two years redesigning the architecture of the software they develop by "*defining and creating a repeatable architectural base for our system*". They claim that this has prevented new developers from "*re-inventing parts of the wheel*" and thereby prevented defects.

Org 3 believed that their innovative approach of understanding their industry by doing a literature review on the product requirements gives them a better understanding of the customers' requirements. Org 4 said that he had no documented processes at all, but he gets sufficient information from the customers.

Org 2 use a bug tracking system which helped them track defects and their resolution thereof.

Subtheme 3d - ISO 9001 Quality Management System works well

Org 1 believed that a lot of their success is attributable to the alignment to the ISO 9001 Management system and they do not need another system to help them as clearly said by the participant:

"I think the ISO process is working for us. Uh, so I don't think we actually need anything else at the moment."

Org 1 presented their process flow as secondary information for this study (Appendix K). Their software development process provides a stable and repeatable framework for conducting their software development and quality assurance activities. They have indicated that measuring the performance of this process is still a challenge and are currently investigating process metrics for this process.

Org 2, while not ISO 9001 certified, are able to track their bugs and debugging during the development process. This gives them some control of their resources and their customers' expectations. The application of a customer relationship database is another tool used by Org 2 to "*track our operational stuff*". Org 3 believed that the management style was important and appreciated the ISO 9001 as a method to work. Their documented process interaction diagram (Appendix K) showed a high-level mapping of their processes. It is evident that the organisations that were most optimistic about processes had some documented versions of these processes. Org 4 on the other hand, did not have any documented processes found that the SQA activities were limited to the testing of the system that was developed.

5.5.4 Theme 4: Possibility of industrialisation

Software quality assurance can be executed through manual means or facilitated using automation tools. An organization's capability to adhere to a defined process naturally

leads to a method that can be consistently replicated and reliably applied. Although automation can boost productivity and improve effectiveness of a process it is difficult to implement. Initial efforts of converting manual efforts into automation costs a significant amount of money and effort (Gonen and Sawant 2020). Maintenance of automation tools is prohibitive too. Gonen and Sawant (2020) also say that large businesses can afford automation tools and they can maintain SQA automation by having a dedicated team unlike for small businesses. On the other hand, small businesses can develop methods very quickly as people are accessible and are flexible to switch roles and contribute to the team. Theme 4 provided an insight into existing small businesses ability to industrialise their SQA processes based on their current and future potential. Table 11 shows the subthemes that emerged to answer research question 4: How could the SQA processes be industrialised in its application?

Subtheme 4a - Automation tools may be used successfully in development, testing and logging.

Participating organisations used automation tools for developing their software (backend and front end) and in the testing phase (for logging defects and responses).

Some organisations shared that they saw the benefits of automation in their development of the back-end. Org3 admitted that automation had a positive impact and said that "*now we've seen that it makes our work easier and we can focus on other work instead*". Org 4 also benefitted from tools for automation of their development work. Specifically in the development of websites, work was reduced by 75%. Org 5 used automation to scale their systems resources on the web services. They also automated reports for their customers and stakeholders.

Table 11 - Description of Theme 4 Subthemes

Subtheme	Description
4a. Automation tools used in development, testing and logging	Small businesses use development, backend, automated testing and logging tools to improve productivity and money.
4b. Processes not mature	Processes differ for different product types indicating immature process design
4c. SQA management strategies vary	SQA strategies may be applied in many different ways in small software businesses.
4d. System level and testing re-usables	All small software businesses focus on customer requirements before commencing development
4e. Start with customer requirements	Small businesses use re-usable templates in development and test cases during testing
4f. Testing strategy involves developers and management	Testing processes in small software businesses are well defined for automation

Org 1 said that they used automated testing for stress and load testing their software. Org 2 on the other hand, used automated logging of their software's "*health*"

"So we've got a bunch of logging in there that we log raw data that we can actually process it in terms of the physical data that we capture the radar data. Then we've got lots of physical components that we can measure. We've got something that's called an exception file, which is exception logging on the system as well, and we build tools for our technicians to be able to visualize all of that".

Subtheme 4b - Some differentiation between product type and process type

When asked whether differentiation between their processes for different products was present, Org 3 said that their firmware development process differed from their computer-based software development whereas Org 1 said that the processes were the same for the different types of software development.

Subtheme 4c - SQA management strategies vary vastly

Interestingly, almost all participating small businesses had a different strategy to manage their software quality assurance processes. Org 2 pointed out that *"depending on the context of what we're developing, our strategy would differ potentially*". They went on to say their differing approaches for software quality assurance was essentially a switch between a quick fix strategy (for front end development) and test-driven strategy (for back end work). The participant said the following:

"We're not completely Dev Ops, but if there's a mistake there we can rectify quite quickly. And it's not going to be fatal to the user as long as the application doesn't crash. So the difference between, say, functionality and then stability if I can separate those two. But on the back end side of things when it comes to safety-critical type of things, we almost use sort of a test-driven design"

Org 3 did not have a documented strategy but went on to say that they start by finding the right people for the job. They must have combination of "*hard skills, soft skills and different experiences*". According to Org 3, if they secure the correct people, they are able to reduce the management interventions in quality management. They believe that every member of the team is responsible for the quality in their specific domain.

Org 1 rely on their ISO 9001 Management system to set up and control their software quality assurance process:

"We currently have a ISO process in place that we use to manage quality in the organization. So we have a, a software process where we capture the requirements, the analysis, the design, and then we plan out a project plan for the implementation. And, uh, we do also have documentation for sign off for pilot and handover".

The participant interviewed in Org 4 felt that even though he did not have a defined strategy for software quality management, the activities that were in place provided the quality in their product which the customers approved of. Small software businesses show an ability to be sufficiently flexible to implement activities that meet their needs to industrialise or at least, automate their organisation.

Subtheme 4d - System level and testing re-usables are available plus some templates

None of the businesses interviewed considered industrialisation as a solution to improving their business's effectiveness but inherently their actions provided an insight into their ability to, at least, automate some activities. To standardise on some activities (so as not to 're-invent the wheel'), Org1 and Org 3 said they have written modules that may be re-used in multiple projects. They have also written test application and tools that are used on multiple projects. Org 3 went a step further and said that their re-usable modules, 'in special in-house repositories', were documented well by commenting the code so that other users could "*understand what is does and how to run it'*:

"We've created a hook for each person. Each person would use and it works for all the features that are in the project. So for testing if we have wrote functions we wrote test functions for certain hooks which can be reused"

Org 5 said that they re-used datasets to test their product:

"so that that helps us a lot to make sure that any changes that we made did not break any in anything fundamental in the functionality of that particular product."

Subtheme 4e - System structure for software dev starts with customer requirements

All participating organizations start their software development off with gathering requirements from their customers. These requirements are used to do their analysis and their feasibility study according to Org1. They then draw up detailed specifications, detailed design and use the requirements to develop test cases. Bear in mind that Org 1 is the organisation with the ISO 9001 implementation in place. Regardless of the whether the participants had a management system, their first step to developing was the collation of customer data to build their product. The methods to gather these requirements varied. Org 2 said that they attended conferences to detect potential requirements for future products. They also use their customer-facing technicians to interact and communicate with clients regarding their needs. Small businesses have found effective ways to extract requirements from the new, existing and potential customers. Org 4, confirming that a visit with a customer is important.

Subtheme 4f - Testing strategy involves use of the developers and management in small businesses

Since small businesses have limited resources, using a dedicated testing team is a luxury and is hence not affordable. However, they do realise the implications of not conducting sufficient testing and it is ensured that everyone in their organisations is mindful of the associated risks. The strategies amongst small software development businesses for testing show that maximum usage will be made of their resources. To this end, developers and management take on the testing responsibility. Org 5 believed that their owner/director, who started the business, had a wealth of knowledge on their customer's needs and was used to test their product after the development team had done their testing. Org 3 sometimes use a junior developer to test. This allows the junior developer to become familiar with the product for future maintenance. The extent of standardisation or modularisation in terms of SQA in small businesses is strongest in the testing process.

5.5.5 Theme 5: Industrialisation will lead to improved effectiveness of SQA in small businesses

The SQA process is the subject of this research study. It is a significant process that runs in parallel with the software development process. Evaluation of the SQA's effectiveness can be divided into several dimensions (Gonen and Sawant 2020):

- cost of implementing SQA (quality costs),
- effectiveness of SQA and
- quality of the software product.

Automation of SQA will improve the effectiveness of the process and boost productivity but it is very difficult to implement, especially in small software organisations. Many automation tools are available in the market that support the SQA process. The cost of the tools and the maintenance adds extra burden. Small businesses have found ways of modularising, standardising and automating activities that work for them. Small businesses are very aware of the limited resources (human and financial) and are therefore 'creative' in their implementation of industrialisation. Table 12 gives a description of the subthemes that emerged for theme 5.

Table 12 - Description of Theme 5 Subthemes

Subtheme	Description
5a. Critical Success Factors are customer based	Critical Success Factors in small businesses are based on meeting customer requirements
5b. Organisational quality culture	Holding all employees to the same standard will improve effectiveness of SQA
5c. Small businesses meet customer requirements	Meeting customer requirements shows effective implementation of SQA
5d. Defect tracking and management system are good starts	Small businesses may start industrialisation journey with defect tracking and a business management system
5e. Communication and training are important	Training and communication at every level from customer to management to development essential

Subtheme 5a – Critical success factors in small businesses are based on meeting customer requirements.

Bogopa and Marnewick (2022), in their study, found that software development projects need: a) committed team; b) client involvement; c) clear requirements, d) good leadership and e) well defined project goals to consider their projects successful. In this research study, very similar responses were given by participants when asked about their critical success factors. Most responded that meeting their customer requirements and happy customers determined their success. Org 2 in particular, said that they monitored their quality by tracking defects. Org 3 believed that they would get benefit from measuring perceived quality and defect fix rate. They did not, however, have anything in place at the time of the interview to measure these factors.

Subtheme 5b – Holding all employees to the same standard will improve effectiveness of SQA.

Small software development organisations are typically run by owners or senior managers/ partners. Strategy for the business is driven by these owners/managers. Most research participants found this to be advantageous. Org 5 goes as far as using their owner to test their software before releasing their software to the client.:

"*Once it is done, it goes into a staging environment and then normally if it's quite technical, then someone like Dennis will test, he's actually a brilliant, brilliant tester. So he's our tester then*"

Owner and manager of Org 3, on the other hand, recognised that some employees were held to a lower standard of quality and that affected their output to the customer, admitting the following:

"..and the deployment team and the development team currently don't have the same quality standards as well, right? No, this is a big gap for us right now."

Subtheme 5c – Meeting customer requirements shows effective implementation of SQA

All the research participating software businesses agreed that meeting their customer's requirements was their main responsibility and as such their existing processes (documented or not) provided with the means to meet their customers' expectations. To this end, every participating organisation has emphasised that they have a method for gathering these requirements. In response to interview question 21 – "What processes are used for gathering requirements?

They have admitted to not having the means of determining the effectiveness or efficiency of these processes but certainly have happy customers.

Subtheme 5d – Small businesses may start industrialisation with defect tracking and a business management system.

Almost all small businesses interviewed were able to monitor and log their defects, during development and post development. They have not set acceptable targets for these defect rates. Setting up key performance indices for their performance and creation of a measurement database are the first steps in setting up a continuous improvement program. Continuous improvement is the key factor in defining and implementing the right quality. Once continuous improvement is in place, different and new questions will arise, such as

a) why is delivery so late?

b) why are there so many changes to the code?
c) are these due to requirement changes or defects or delayed testing?

The organization can then scrutinize its decision: should they invest test automation, or should they invest in process optimisation on delivery or earlier testing. This would perhaps lead to modified tasks.

For Org 1, their ISO management have provided the essentials of quality governance that have delivered and effective SQA process. They have, however, not been able to define a process performance measurement at the time of the interview. ISO 9001 provides a quality governance framework for all three levels of the organization: strategic, planning and operational. This in return provides the business with focusing on quality in a holistic business-driven approach.

Subtheme 5e – Training and communication at every level from customer to management to development essential.

In software development, communication takes place everywhere and all the time. Small businesses are especially good at communication, due to their accessibility to all employees and their 'flattened' management structure. Developers are often able to directly communicate with the customers to clarify their requirements and directly with management about their ability to meet these requirements within the constraints of time and budget. Requirements are the strongest drivers of a project's direction. It is therefore important to be clear about them. All participants in this research study indicated that they had an effective communication channel to their respective stakeholders.

5.5.6 Summary of findings

Sections 5.5.1 to 5.5.5 presented the themes that emerged from the interview data using thematic analysis. The data from these sections also enable the researcher to answer the research questions set out in section 1.4.

Table 13 has a summarised view of the results and show existence of the industrial dimensions in the software businesses that participated in the research study. Some elements clearly do not exist, while some partially exist, and some elements are currently

implemented. The gap between their current process and an industrialised process may then be identified and a pathway may be mapped.

Table 13 - Existence of industrialisation dimension in participating organisations – (Source – Developed by Author)

		Business Layers		
		Strategy	**Planning**	**Operational**
Industrialisation Dimensions	**Modularisation**	😐 SQA strategy 😐 Critical success factors 😐 KPI strategy RQ4	🙂 Requirements gathering RQ2	🙂 Testing strategy 🙂 Intermediate artefacts 😐 Project specific processes RQ4
	Standardisation	☹ Reference standards ☹ Quality budget RQ1	🙂 Right quality ☹ Project management 😐 Processes 😐 Benefits 😐 Constraints ☹ Reporting and measurement database RQ2	🙂 System structure 🙂 SQA processes 😐 Templates 🙂 Re-usable – testing ☹ Styling guides ☹ Standardised working environment RQ4
	Specialisation	🙂 Automation software RQ4	Who does SQA RQ4	☹ Skill, training ☹ Quality gate RQ3
	Automation		🙂 Automation tools RQ4	Savings RQ3
	Continuous Improvement	☹ Organisational structure ☹ Quality governance RQ4	☹KPI measure ☹ Quality Management RQ1	😐 Measurement database 🙂 Least effective 🙂 Most effective RQ3

Key:

🙂 - dimension exists
😐 - dimension partially exists
☹ - dimension does not exist

In summary, the themes presented the following findings:

- Quality costs in small businesses are not measured.

- Small businesses have some elements of an SQA process currently.
- SQA process in small businesses work but the effectiveness is not measured.
- SQA industrialisation is best applied to the testing phase in small businesses.
- Industrialisation will improve effectiveness of SQA in small businesses.

5.6 Conclusion

The industrialisation of software quality assurance in small businesses entails the implementation of the dimensions of industrialisation by determining the factors that exist in small businesses that affect modularisation, standardisation, specialisation, automation, and continuous improvement as outlined in Chapter 3. The results were analysed through the lens of these dimensions of industrialisation and the interview questions were designed to explore the information from the research participants in these dimensions.

This chapter outlined the results and interpretation of the findings generated from the qualitative approach. Five organisations were interviewed, with software developers and Quality assurance representatives making their contribution. The data was analysed using thematic analysis. Based on the underpinning concepts and dimensions of industrialisation, key themes were extracted through which the research questions were addressed and answered.

6 CHAPTER 6 -CONCLUSION

"A computer is the most incredible tool we've ever seen. It can be a writing tool, a communications centre, a super calculator, a planner, a filer, and an artistic instrument all in one, just by being given new instructions, or software, to work from. There are no other tools that have the power and versatility of a computer." ~ Steve Jobs

6.1 Introduction and Summary of study

Researchers and authors have demonstrated the positive impact of total quality management and industrialisation in all industry types, software development has been no exception. Small businesses and their challenges have also been a large topic of research, especially since small businesses form a large part of the global economy. The challenges facing these small businesses are well documented with South Africa seeing a failure of 70% in small businesses in the first year of business (News24 2022). Software usage and its development has seen exponential growth in the recent decades. Software products are often developed by small software businesses. The adoption and awareness of total quality management and industrialisation have provided the methodology for large business to succeed, however, the adoption of these methods have been challenging for small software development businesses and specifically their software quality assurance (SQA) process. Wieczorek, Vos and Bons (2014) have proposed methods for industrialising the software quality assurance process but have focused on large businesses. Further research was hence warranted to explore and inform how industrialisation of the SQA process may be effective in small software businesses.

The current gaps and the limited number of studies on industrialisation in the SQA process demonstrated the need to identify how the SQA processes in small businesses was practiced and how the dimensions of industrialisation could contribute to an effective industrialised SQA process.

The aim of this study was to explore the effectiveness of industrialising the software quality assurance process in small software businesses. The research was based on interviews conducted with five South African software development businesses and participants comprised of software developers and quality assurance representatives.

Thematic analysis was used to interpret the findings and the industrialisation dimensions of modularisation, standardisation, modularisation, automation, and continuous improvement on the three business layers of strategy, planning and operations were used as the theoretical factors to interpret the findings. Several themes emerged which included:

- Quality costs are not measured in small businesses.
- Small businesses have some elements of an SQA processes.
- SQA in small business work but the effectiveness in not known.
- SQA industrialisation is best applied to the testing phase in small business.
- Industrialisation will improve effectiveness in the long term.

This research study has six chapters.

Chapter one outlined the research study conducted and the research context. It introduced the importance of quality software product delivery and its benefits and the struggles of small software businesses to achieve the levels of quality that are expected by consumers. The chapter briefly introduced the software development process and how it relates to the quality assurance process. The chapter also described the aim of the study: The aim of the study was to investigate if industrialisation, as proven in other industries, will provide an effective solution to software quality assurance challenges in small businesses.

Chapter two addressed the literature related to the research problem. This chapter provided and overview of the state of quality assurance in small businesses. The chapter also sketched an understanding of the software quality assurance domain within small businesses with TQM as a framework. Scholars argue that small businesses are essential to the economy in any country and that quality of software (in software development businesses) is a necessity for the sustainability of these businesses but not many have determined the most effective method to perform software quality assurance. This gap enforced purpose of this study - to highlight what software businesses are currently doing in their SQA domain and investigate if an effective industrialised process for SQA in these small organisations can exist.

In Chapter Three, a theoretical lens through which the industrialisation of quality assurance of software may be viewed, is presented. The importance of understanding how industrialisation can be applied to the software quality assurance industry is key to

understanding how the data collected may be used to analyse the results of this study. The objective of the presented approach was to find a way by which the industrialisation dimensions may be utilised in small businesses. The concepts explained in this chapter help bring meaning to the analysis of the results.

Chapter Four outlined the research methodology, methods, approaches, and ethical approaches to collecting and analysing data for this study. A qualitative study, using the interpretivist philosophy, an inductive approach and exploratory strategy by utilising semi-structured interviews to collect data was employed. Five small software businesses were used as participants.

Chapter Five outlined the results and interpretation of the findings generated from the qualitative approach. Five organisations were interviewed, with software developers and Quality assurance representatives making their contribution. The data was analysed using thematic analysis. Based on the underpinning concepts and dimensions of industrialisation, key themes were extracted through which the research questions were addressed and answered.

Finally, Chapter 6 provided a conclusion to the research study and recommendations. To conclude this research report, this chapter is further organized as follows: the overview of the research, Research questions revisited, Research contributions, recommendations, Limitations, the conclusion and finally, further research recommendations.

6.2 Research questions re-visited

The research questions that guided this qualitative study were geared towards addressing the research problem to investigate if industrialisation of the SQA process can help improve the effectiveness of SQA implementation in small software businesses. To address the problem, it was necessary to examine the phenomenon from the perspective of software developers and quality assurance representatives in small software businesses. These individuals work in environments where the challenges of small businesses are experienced regularly. Some of these participants are owners and managers of these businesses. The qualitative research study included the following questions:

RQ1: What costs are incurred by the organisations to achieve their desired quality level?

RQ2: Do small software development businesses have a defined SQA process in place?

RQ3: How effective are the SQA activities in these organisations?

RQ4: How could the SQA process be industrialised in its application?

RQ5: Will this industrialisation improve the effectiveness of the SQA process?

These five research questions guided the investigation, understanding and explanation of the participants' insights and experiences into the effectiveness of their software quality process as it currently exists.

6.2.1 RQ1: What costs are incurred by the organisations to achieve their desired quality level

Small businesses used standards for software product compliance but did not use standards or management systems to govern their quality. They found it difficult to set and measure key performance indicators. As an outcome, they did not allocate specific budgets to achieving quality. They, therefore, were not able to measure the costs (appraisal and failure costs) they incurred on achieving their quality levels. Costs incurred by organisations to achieve their desired quality level in unknown.

6.2.2 RQ2: Do small software development businesses have a defined SQA process in place?

Small software businesses have established certain operational procedures, although these processes remain undocumented. While they do observe certain advantages from these practices, they also recognize a number of limitations. The agile methodology proves advantageous for specific aspects of their workflow, yet it becomes unwieldy when applied to the entirety of the development cycle. Nevertheless, these businesses exhibit the capacity to adjust and switch roles as needed.

The absence of proper documentation implies that their testing protocols lack the comprehensiveness required, and their test data is insufficient for timely defect detection and resolution. While these businesses maintain regular communication with their

customers, this communication is not formally recorded. In small enterprises, the responsibility of Software Quality Assurance (SQA) often falls to the developers due to resource constraints that hinder the establishment of an independent quality assurance department. This arrangement restricts the impartiality in testing and the execution of quality assurance activities.

The implementation of the ISO 9001 standard furnishes these businesses with a robust framework for governing quality, enhancing their overall quality control structure. Only one business interviewed had an working ISO 9001 system in place.

6.2.3 RQ3: How effective are the SQA activities in these organisations?

Small businesses anchor their quality standards to the satisfaction of their customers. Conducting a thorough assessment of their operational processes and the efficacy of their products is not feasible due to resource limitations. Small businesses generally lack a structured approach to training and communication, often providing training only when a specific need arises. Communication within small businesses is more straightforward compared to larger enterprises, as employees are more easily reachable. In the realm of small software development businesses, they diligently monitor defects to efficiently allocate resources during product development. The introduction of the ISO 9001:2015 quality management system serves as a foundation for necessary quality oversight. The management of Software Quality Assurance (SQA) activities exhibits its highest effectiveness when supported by a well-defined management system.

6.2.4 RQ4: How could the SQA process be industrialised in its application?

An organization's ability to adhere to a defined process inherently results in an approach that can be consistently duplicated and dependably executed. Small businesses employ tools for both their back-end and front-end development, leading to observable enhancements in productivity. These businesses have also experienced notable productivity improvements due to the use of automation tools for stress and load testing their software products. The strategy for Software Quality Assurance (SQA) displays

diverse applications within small enterprises and is contingent on the intricacy of their products.

Small businesses adopt an approach of creating reusable modules for their software products, which extends to reusable test cases and scripts as well. Additionally, small software development businesses place a strong emphasis on clarifying customer requirements. It's common for them to involve their developers, managers, or owners in testing the final product before deploying it to the customer. This hands-on approach helps ensure that the product aligns closely with customer expectations.

Industrializing the Software Quality Assurance (SQA) process involves systematizing and standardizing the application of quality assurance practices in a way that resembles industrial manufacturing processes. The SQA processes in small businesses can be industrialised in its application by:

- Standardising their SQA procedures that encompass their entire software quality assurance cycle. These procedures should cover requirements analysis, testing, defect management, and documentation.
- Implementing automated testing tools for repetitive and routine tests
- Establishing consistent testing environments to replicate real world conditions.
- Develop comprehensive and well documented test cases that cover various scenarios.
- Defining key metrics for measuring the effectiveness of the SQA process, such as defect density, test coverage, and pass rate. These metrics help to identify areas for improvement.
- Creating and maintaining a repository of quality assurance documentation, including standards, guidelines, and best practices.
- Cultivatjng a culture of continuous improvement, where lessons learned from each project are applied to enhance the SQA process for subsequent projects.
- Implementing streamlined reporting mechanisms that provide actionable insights without excessive administrative overhead.

6.2.5 RQ5: Will this industrialisation improve the effectiveness of the SQA process

In the context of small software businesses, there appears to be a deficiency in the strategic layer. However, elements of planning and governance layers are present to varying degrees, while the operational layer stands out as the most robust aspect in these businesses.

Industrialization can improve the effectiveness of the software quality process to some extent, even if the strategies are not fully in place. However, the impact of industrialization will likely be more limited and may not reach its full potential without well-defined strategies.

Even without fully developed strategies, some level of standardization can be introduced. This can help establish consistent practices and reduce ad-hoc approaches, leading to a more structured quality process. Basic automation can still be implemented, such as automated testing for repetitive tasks. While this might not cover the entire range of quality assurance activities, it can save time and improve efficiency. Implementing basic metrics and reporting mechanisms can provide visibility into the quality process. While these might not be as comprehensive as with well-defined strategies, they still offer insights for improvements. Industrialization itself promotes consistency. By introducing standardized tools and processes, variations in quality of the output delivered to the customer can be reduced. Some efficiency gains can still be achieved by automating certain aspects of testing and implementing streamlined workflows.

While industrialization can bring some benefits to the software quality process even without fully developed strategies, the impact and effectiveness will be more significant when well-defined strategies are in place.

6.3 Research Implications Contributions

6.3.1 Theoretical Contributions

The theoretical framework for the industrialization of software quality assurance is based on principles borrowed from industrial manufacturing processes and quality management

methodologies. Chapter 2, Literature review and Chapter 3, The Underpinnings of Industrialisation (of SQA) describe a framework that provides a structured approach to enhance the effectiveness, efficiency, and consistency of software quality assurance practices. Key theoretical components that contribute to this framework include: Total Quality Management (TQM), industrialisation dimensions (modularisation, standardisation, specialisation, automation and continuous improvement), metrics and data driven decision making, standardised documentation, training and skills development, customer-centric approach and management commitment. Incorporating these principles into the theoretical framework for industrializing software quality assurance results in a holistic approach that improves software quality, reduces defects, enhances overall development processes, and improves the sustainability of small software businesses.

Numerous findings within this research were consistent with the advice presented in the models and foundational principles set forth by these frameworks. The framework served as a roadmap for shaping the interview questions, enabling participants to delve into their insights and firsthand experiences. This study marked a crucial initial stride in comprehending the effectiveness of industrializing software quality within small businesses, furnishing them with a sustainable operational approach.

Through this study, it became evident that despite resource constraints, small businesses endeavour to adhere to approaches that uphold customer satisfaction. Furthermore, the investigation revealed that quality management strategies within small businesses necessitate bolstering for greater effectiveness.

6.3.2 Methodological Contributions

The empirical nature of this research study required interpretive inquiry as data analysis required the researchers and participants interpretation. A qualitative approach was employed in this study because the research information needed to be evaluated in greater detail and research data needed to come from participants' experience and knowledge. Semi structured Interviews were the choice for data collection to gain deep insights into the research problem of investigating the effectiveness of SQA industrialisation in small businesses. Thematic analysis was used to analyse the data

generated from the qualitative approach. The contribution of this study to research methodology is given by taking an interpretative stance, a qualitative approach with semi-structured interviews and thematic analysis to address the research problem and may serve as a guide to other researchers.

6.3.3 Practical Contributions

The aim of this study was to investigate the effectiveness of industrialising the SQA process in small businesses. To address the aim the main objectives

- To get rich insights into the factors that affect the effectiveness of the current SQA processes in small businesses.
- To explore how industrialisation impacts the effectiveness of the SQA processes.

The practical contribution of this study is for small businesses to implement the processes that are currently most ready for the industrialisation. This would be the testing and debugging processes.

The practical contribution of this study lies in advising small businesses to prioritize the implementation of processes that are most mature for industrialization as they currently stand —specifically, the testing and debugging processes.

6.4 Recommendations

This study was conceived to address an identified gap in the literature regarding the effectiveness of industrialisation of the SQA process in small businesses. The study contributed to the body of knowledge by highlighting the experiences and challenges in small businesses regarding their software quality assurance.

While industrialization can bring some benefits to the software quality assurance process even without fully developed strategies, the impact and effectiveness will be more significant when well-defined strategies are in place. A structured, strategic approach to industrialization maximizes the potential for improved software quality and overall process efficiency.

The results of the study recommends that small businesses focus on their strategic direction regarding quality assurance and governance. Well defined strategies ensure that all aspects of the quality assurance process are considered, leading to a more comprehensive and effective approach. Strategies guide the selection and implementation of automation tools based on specific needs. This leads to more efficient automation of key testing and quality processes.

Additionally, the following are recommended for small businesses:

- SQA leaders should define and closely monitor relevant metrics within SQA processes. Suggested metrics extend beyond the number of faults logged and should include defect density and quality costs.
- Accurately measuring time spent correcting defects will highlight lost profits, because of wasted time for SQA leaders.
- An investigation into industry best practices of Six Sigma, CMMI, and ISO 9001 may provide insight to SQA leaders that will strengthen existing SQA processes whether adopted in whole or in part.
- Software Quality assurance artefacts to be considered at the relevant stages are:
  A. Documentation – like specifications, templates, quality gates, test reports, test plans and test data to included.
  B. Methodology – such as test case design, test execution, measurement and reporting
  C. Software tools – such as management tools, performance tools, test databases, test environment
  D. Quality tools - such as validation and verification rules, test cases, test scripts

  The Small business quality house in Figure 10 is expanded to show the SQA process where the recommended SQA artefacts may be added to improve the effectiveness an industrialised SQA process. Figure 12 shows where the four Software quality assurance artefacts described above (A,B,C,D) would be used in a small businesses quality assurance process.

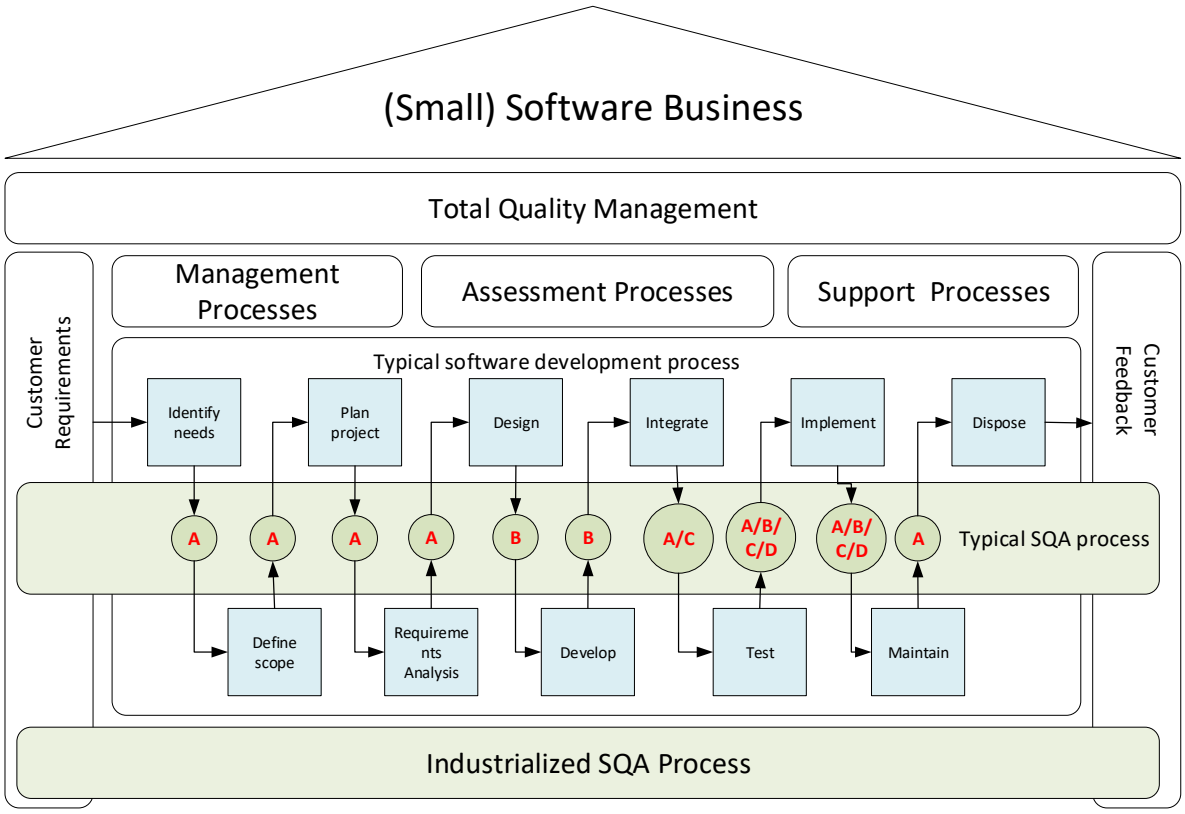


Figure 12 - The small business quality house (with recommended SQA artefacts)

Source: Adapted from Wieczorek, Vos and Bons, 2014

6.5 Conclusion

Industrialisation of software quality assurance has the potential to be effective in small software development businesses. By using the dimensions of industrialisation of modularisation, standardisation, specialisation, automation, and continuous improvement and setting strategic direction for quality assurance, small software development companies will become sustainable.

6.6 Limitations and recommendations for further research

The purpose of this qualitative exploratory study was to explore the effectiveness of industrialising SQA process in small software development organizations. The results of this study highlighted the state of five businesses in South Africa and the effectiveness of their SQA processes. The use of existing third-party tools to aid industrialisation are

numerous and were not included in this study. Further research into the benefits of these tools and their applicability is recommended.

Recommendations to improve their current processes in small software businesses were presented in section 6.4 above but a model or framework for industrialisation of SQA in small businesses was not presented in this study and could provide the objective for further research studies.

7 APPENDIX

7.1 Appendix A – Interview Guide

Interview Guide

The interview questions align with the central research study. The table below shows a list of questions to be used in the interviews with participants in the research study. Questions are categorized into areas of Strategy, Planning and Operations but differentiated by the stages of industrialisation (Modularisation, standardisation, automation, and Continuous Improvement). The questions are directed to the SQA representatives and software developers.

An interview guide is presented below to assist with interview of each participant.

	Introduction
	Welcome the participant
	Introductions (Names and Backgrounds)
	Explain the Interview process o State the interview's purpose: to explore your opinion and experience on Software quality assurance in your organisation to learn more about the current process. The outcome of the research can help identify challenges and improve the process. o Explain that there is no right or wrong answer, and they will not be judged in any way o Explain that some documents and processes may be requested if participant is willing o State that the interview will last about 45 – 60 minutes of their time.
	To gain the maximum effectiveness from the interview, questions will be categorised in 3 sections. Some questions may have one or more probing question to enable the participant and researcher to build on the main answer**.**
No	**Interview Questions**
	Section A **Strategic – How does the business landscape accommodate strategies for SQA, standards and regulations.**
1	Describe the strategies used for managing software quality in your organisation? *Probe: How does your organisational structure support this strategy*
2	How are the critical success factors for measuring software quality described? *Probe: Can you give an example?*
3	What strategy is in place to manage and measure these KPI? *Probe: How are these measurements recorded and tracked?*
4	What training and communication strategies are in place for the SQA? *Probe: Does this strategy include areas of technical training?*
5	What reference standards or international standards are used?
6	What strategies are least effective for managing your software quality?
7	What strategies are most effective for managing your software quality?

	Section B **Planning – How does the business plan, monitor and control the SQA strategy**
8	How does the organisation know whether products have the correct quality? *Probe: Does a reporting and measurement database exist?*
9	How is the project management methodology used in the organisation described? *Probe: Does this include project and product governance procedures?*
10	Are budgets specified for quality and risk management?
11	What processes, procedures and instructions are in place? *Probe: Are examples available please for sharing?*
12	What role does automation tools play in the organisation? *Probe: Can you give examples of this*
13	How would you describe the strategy for testing during the software development cycle?
14	What are the benefits and constraints in the SQA process?
	Section C **Operational – How are the plans for SQA executed?**
15	Describe the system structure for developing software? *Probe: Are standardised modules and styling guides used to develop software?*
16	How are SQA processes defined? *Probe: Are these processes defined by product type or project type?*
17	Who performs the Quality assurance function?
18	What templates are used for SQA?
19	What re-usable components available at a system level?
20	What re-usable components are available at testing stage?
21	What processes are used for requirements gathering?

7.2 Appendix B – Consent Letter

CONSENT

Full Title of the Study: An investigation into the effectiveness of industrialising Software Quality Assurance (SQA) in small software businesses

Names of Researcher: Meena Patel

Statement of Agreement to Participate in the Research Study:

- I hereby confirm that I have been informed by the researcher, Meena Patel, about the nature, conduct, benefits and risks of this study - Research Ethics Clearance Number: **IREC 127/22**
- I have also received, read and understood the above written information (Participant Letter of Information) regarding the study.
- I am aware that the results of the study, will be anonymously processed into a study report.
- In view of the requirements of research, I agree that the data collected during this study can be processed in a computerized system by the researcher.
- I may, at any stage, without prejudice, withdraw my consent and participation in the study.
- I have had sufficient opportunity to ask questions and (of my own free will) declare myself prepared to participate in the study.
- I understand that significant new findings developed during the course of this research which may relate to my participation will be made available to me.

Full Name of Participant	**Date**	**Time**	**Signature** /

I, Meena Patel, herewith confirm that the above participant has been fully informed about the nature, conduct and risks of the above study.

Full Name of Researcher	**Date**	**Signature**
Full Name of Witness (If applicable)	**Date**	**Signature**
Full Name of Legal Guardian (If applicable)	**Date**	**Signature**

7.3 Appendix C – Information Letter

LETTER OF INFORMATION

Title of the Research Study: An investigation into the effectiveness of industrialising Software Quality Assurance (SQA) in small software businesses.

Principal researcher: Meena Patel, BSc (Hons) Electronic Engineering

Supervisor: Dr Oludolapo A Olanrewaju, DTech Industrial Engineering, HOD Industrial Engineering

Good day,

I am a 2nd year student at DUT doing research for my Master's degree in industrial engineering.

I would like to invite you to participate in the research. Before you decide to participate in this study, it is important that you understand why the research is being done and what it will involve. Please read the following information carefully. Please let me know if there is anything that is not clear or if you need more information. Research is a systematic study of materials and sources to establish and reach new conclusions which contribute to a body of knowledge. This new knowledge can help us tackle pressing problems and issues and help improve situations.

Purpose of this research: Through systematic literature review it was observed that software quality assurance processes in small businesses are often costly and difficult to implement. This research is being conducted to investigate if industrializing the software quality assurance (SQA) process will help increase the effectiveness of SQA implementation in small software development organizations.

Outline of the Procedures: Participation in the study involves completion of an interview which will last about one hour. The interview will be audio-recorded, and the interview will be transcribed into written transcripts later for the purpose of analysis. The interviews will occur in a setting that is mutually comfortable, such as a local coffee shop or done via an online platform. Secondary information, such as templates and process documentation may be requested from you during the interview.

Risks or Discomforts to the Participant: There are no risks or discomforts that are anticipated from your participation in the study.

Withdrawal from the Study without prejudice: Participation in this study is voluntary; and refusal to participate will involve no penalty. You are free to withdraw consent and discontinue participation in this project at any time without prejudice from this institution.

Benefits: You will receive a copy of this study to use as a reference for SQA processes.

Remuneration: You will not be compensated in any way for participation in this study.

Costs of the Study: You will not be expected to cover any costs towards the study.

Confidentiality: The information gathered during this study will remain confidential. Only the researcher and the Durban University of Technology will have access to the study data and information. There will not be any identifying names on the recordings and participant's names will not be available to any-one. The results of the research will be published in the form of a paper and may be published in a professional journal or presented at a conference. The information will help others better understand the industrialization of Software Quality Assurance processes.

Results: The data collected will be analyzed and results and conclusions documented in the final dissertation.

Storage of all electronic and hard copies including audio recordings : All data collected for this study will be stored for 5 years using a secured folder in a password protected cloud based storage. All hardcopies of data will be destroyed after the 5 year expiry and all electronic data will be deleted..

Persons to contact in the Event of Any Problems or Queries: Please contact the researcher Meena Patel (0828382773.), my supervisor Dr Oludolapo Olanrewaju _(DTech Industrial Engineering) on 031 373 6314 or the Institutional Research Ethics Administrator on 031 373 2375. Complaints can be reported to the Director: Research and Postgraduate Support Prof Keo Motaung on 031 373 2577 or researchdirector@dut.ac.za.

7.4 Appendix D – Gatekeeper permission (from Supervisor)

31 October 2022

To whom it may concern

Introducing Meena Patel – MEng Student (Durban University of Technology)

Dear Sir/Madam

This letter serves to confirm that Meena Patel (student no 22176189) is a duly registered MEng student in the Department of Industrial Engineering, at the Durban University of Technology. She is currently conducting research titled: *An Investigation into the effectiveness of industrialising Software Quality Assurance in small software businesses* under my supervision.

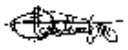
The outcome of the study is expected to improve software quality assurance processes in small software businesses and extend the theory in this field. As part of requirements for the degree she is expected to undertake research in an environment that is applicable to the study. The DUT ethical compliance regulations require her to provide proof (letter or email) that the relevant authority where the research is to be undertaken has given approval.

We appreciate your support and understanding to grant Meena Patel permission to carry out research in your organization.

If you require any further information, please do not hesitate to contact me oludolapoo@dut.ac.za or Meena Patel (22176189@dut4life.ac.za).

Thank you for your time and consideration in this matter.

Yours sincerely,



Dr O.A. Olanrewaju
Supervisor and HOD: Department of Industrial Engineering, Durban University of Technology

7.5 Appendix E – Gatekeeper permission (from Researcher)

31 October 2022

To whom it may concern

Request for Permission to Conduct Research in your Organization

Dear Sir/Madam

I am a registered Master of Engineering student in the Department of Industrial Engineering, at the Durban University of Technology, South Africa. The title of my research is: *An Investigation into the effectiveness of industrialising Software Quality Assurance in small software businesses.*

I have identified your organization as a potential role player in my study. I therefore humbly request permission to conduct the study in your organization. I intend to interview the software quality assurance lead and one software developer. The interviews will take about 60 minutes. Sharing of any process flow diagrams and procedure documents will be highly appreciated. I will sign a non-disclosure agreement with the organization should it be required. Participants will remain anonymous. All information received will be treated with the utmost confidentiality. There are no known risks, current or anticipated to any participant in this research.

I absolutely understand the time constraints that you may have and promise to co-operate with your organization in every possible way. I believe that the results of this study will not only be important to me but to your organization as well. New information that might be useful to your organization will emerge from this study.

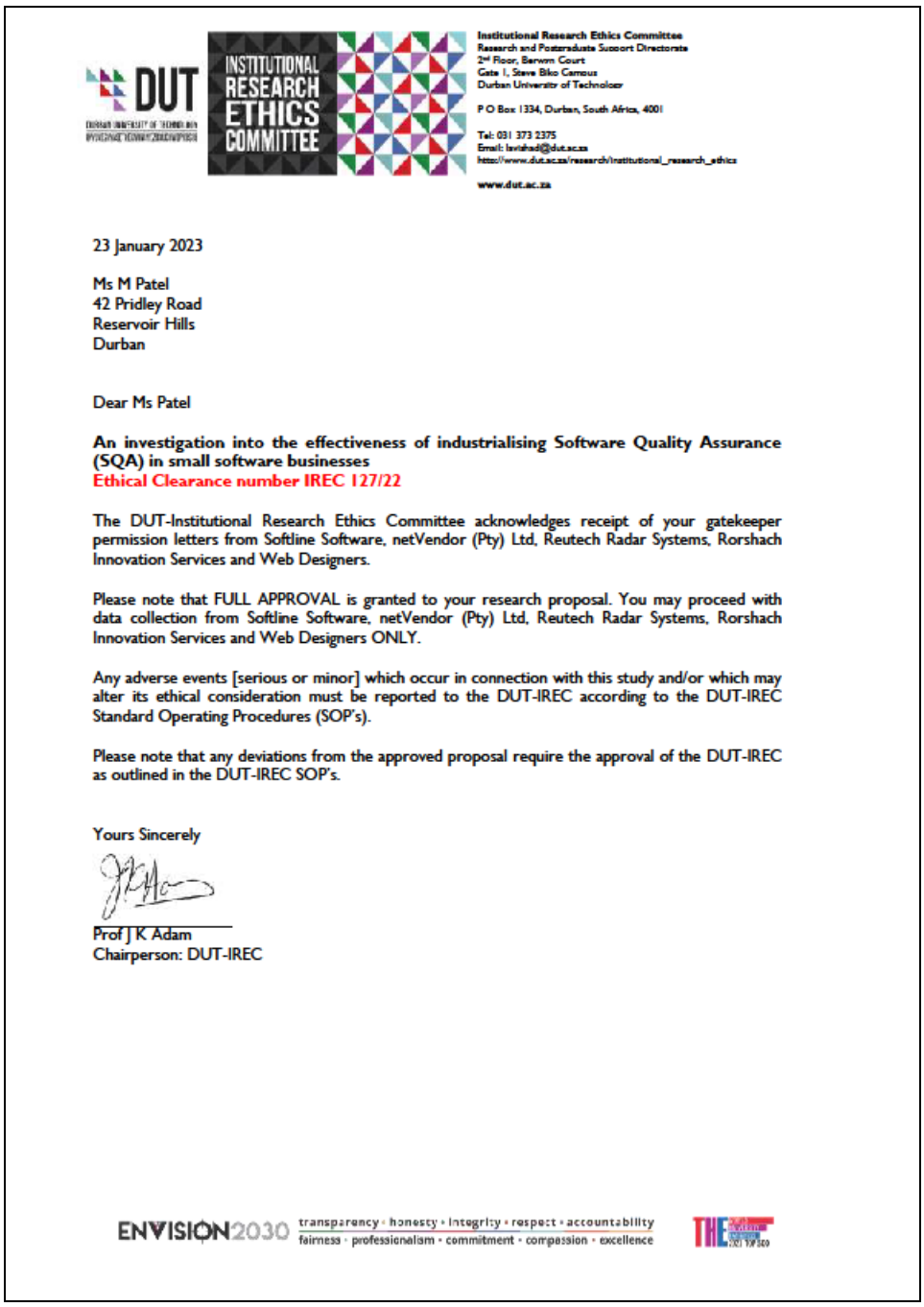
I hope my request will receive your favorable consideration.
My contact details are
Email: 22176189@dut4life.ac.za (primary) or meena@bpg.co.za (secondary)
Tel: 082 838 2773

Thank you for your time and consideration in this matter.

Yours sincerely,

Meena Patel

7.6 Appendix F – Ethics Clearance Letter



Institutional Research Ethics Committee
Research and Postgraduate Support Directorate
2nd Floor, Berwin Court
Gate 1, Steve Biko Campus
Durban University of Technology

P O Box 1334, Durban, South Africa, 4001

Tel: 031 373 2375
Email: lavishad@dut.ac.za
http://www.dut.ac.za/research/institutional_research_ethics

www.dut.ac.za

23 January 2023

Ms M Patel
42 Pridley Road
Reservoir Hills
Durban

Dear Ms Patel

An investigation into the effectiveness of industrialising Software Quality Assurance (SQA) in small software businesses
Ethical Clearance number IREC 127/22

The DUT-Institutional Research Ethics Committee acknowledges receipt of your gatekeeper permission letters from Softline Software, netVendor (Pty) Ltd, Reutech Radar Systems, Rorshach Innovation Services and Web Designers.

Please note that FULL APPROVAL is granted to your research proposal. You may proceed with data collection from Softline Software, netVendor (Pty) Ltd, Reutech Radar Systems, Rorshach Innovation Services and Web Designers ONLY.

Any adverse events [serious or minor] which occur in connection with this study and/or which may alter its ethical consideration must be reported to the DUT-IREC according to the DUT-IREC Standard Operating Procedures (SOP's).

Please note that any deviations from the approved proposal require the approval of the DUT-IREC as outlined in the DUT-IREC SOP's.

Yours Sincerely

Prof J K Adam
Chairperson: DUT-IREC

ENVISION2030 transparency • honesty • integrity • respect • accountability fairness • professionalism • commitment • compassion • excellence

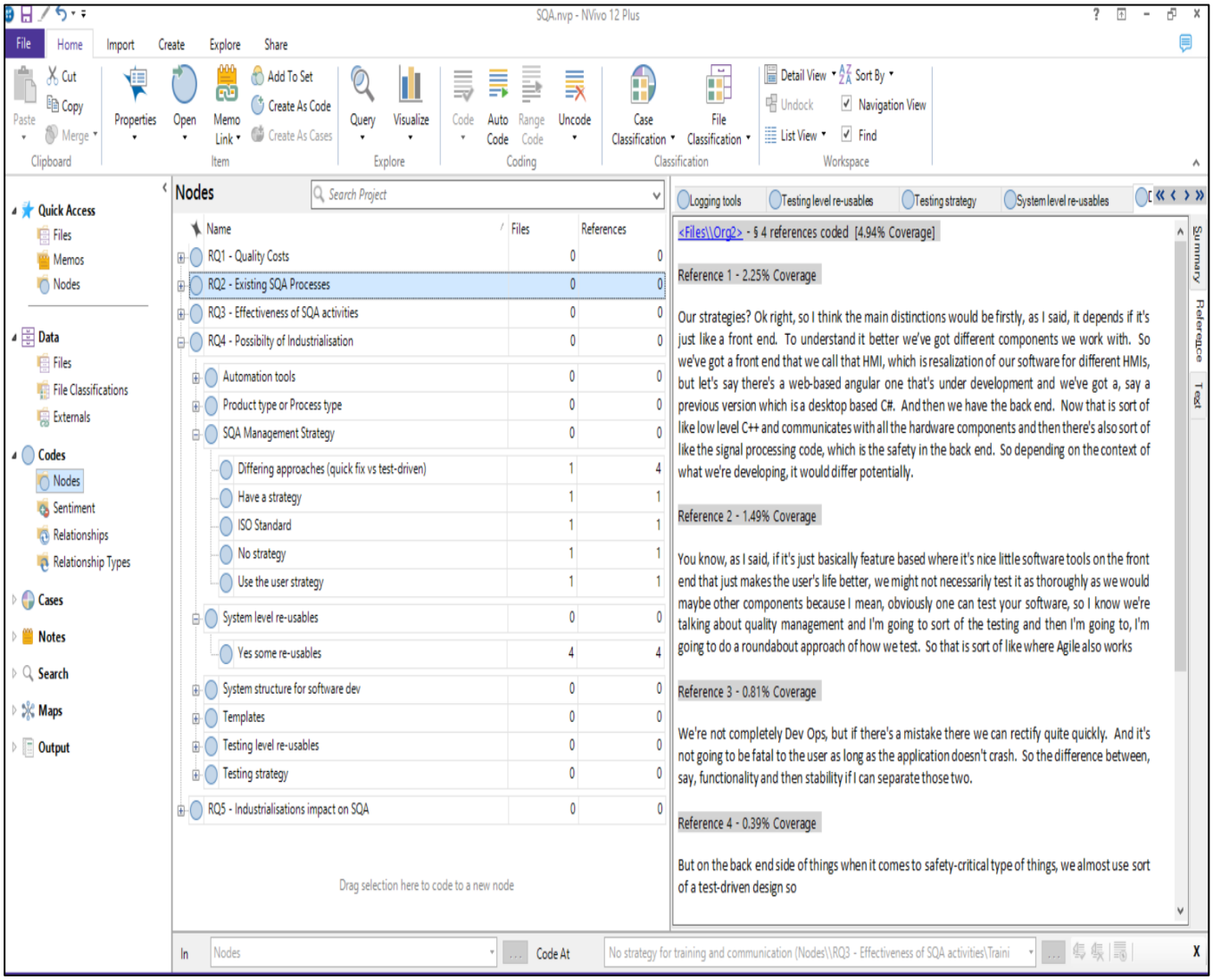
7.7 Appendix G – Codebook

Code Name
Budgets for risk and quality
No budgets
Not sure about budgets
Yes to budgets
Measuring of KPIs
Customer tells us about quality
KPIs fell off the wagon
We take meeting notes
We use statistics
Reference or International standards
Avoid re-invention of the wheel
Yes we do use standards
Benefits and constraints
Business value
Lack of documentation
Low pressure environment
Scalable and stable
Time constraints
Most effective strategies
Bug tracking system
Management System
Testing
Processes, procedures and instructions
Software development process
Some sort of a process
Project Management methods
Agile
No project management
Project plans
Requirements gathering
Ask the customer
Its a challenge
The correct quality
Dont know the correct level
Happy customer
Have to meet a target
Testing
Who does SQA
Developer and Tester
Developer tests
Tester tests
Least effective strategies
Holding teams to different standards
Look for improvements

Code Name
Simple stuff
Most effective strategies
Bug tracking
Good information from client
ISO Process
Literature Review
Re-architectured
Training and Communication
Circulate to staff
Good at communicating
No strategy for training and communication
Struggle with communication
Will put something in place for training
What is correct quality
Customer complaints
Happy customer
Meeting notes recorded
Meets requirements
Operational requirements
Testing
Automation tools
Backend tools
Development tools
Finance processing tools
Logging tools
Testing tools
Product type or Process type
Continuous development
Different process for diffferent product types
Same process
SQA Management Strategy
Differing approaches (quick fix vs test-driven)
Have a strategy
ISO Standard
No strategy
Use the user strategy
System level re-usables
Yes some re-usables
System structure for software dev
Full process - starts with product management
Templates
Maybe some templates
No templates
Testing level re-usables
Yes some re-usables
Testing strategy

Code Name
In-house tester
Its complicated
Test after code during sprint
Test driven
Correct quality (see RQ3)
Critical Success Factors
Ask the customer
Many success factors
No KPIs
Quality Metrics
Least effective strategies (see RQ 3)
Most effective strategies (see RQ3)
Training and communications (See RQ3)

7.8 Appendix H – Nvivo Nodes Screen Capture



7.9 Appendix I – List of Codes Mapped to Research Questions

Table 14 - Mapping codes to research questions

Research Question	Description	High level Category	Related Codes
RQ1 : What costs are incurred by the organisation to achieve the desired quality level?	The organisation's ability to measure costs of achieving quality The organisations ability to set and control targets related to quality and risk are explored	Measure, track and reduce costs to be effective	Cost allocation to quality costs but insufficient; No budgets or cost tracking for quality No known budgets but discussed with director Attempted KPI tracking but not sustainable Customers tell us about our quality No KPIs for defects but minutes of meetings kept Some automatic tools provide statistics Standardised design to avoid re-invention of the wheel Various standards are used
RQ2 : Do small software development businesses have a defined SQA process in place?	Some amount of structure, capability to repeat and reproduce products in an organisation will enable small software businesses to begin a journey of quality improvement and industrialisation	Processes provided repeatability and reproducibility to the outputs	Focus on time to market rather than documentation Insufficient negative testing Tools developed provide business value Well architectured system implies low maintenance No process but bug tracking helps with control Project Management methods only partially used Some sort of a process but not clearly defined Get requirements from the customer but not always documented Developer and Tester do SQA Have to meet a contractual target Quality is correct if the customer is happy

Research Question	Description	High level Category	Related Codes
RQ3: How effective are the SQA activities in these organisations	If these businesses are able to repeat and reproduce to a certain extent their current SQA activities will be effective	Existing SQA activities are effective but not measured	Different targets or standards for different employees Do small simple incremental stuff Look for improvements Good at communicating No strategy for training and communication A sound architecture for the system most effective Good requirements are effective Partial tracking and management systems are effective Bug tracking and CRM system provide control ISO 9001 Management System works well Rely on system testing
RQ4: How could the SQA process be industrialised in its application	Do small businesses have elements of their processes that they may industrialise	Some elements are able to be industrialised and will make small businesses more effective	Development and backend tools used Testing and Logging tools Different process for different product types Same process for product and process type Differing approaches (quick fix vs test-driven) Have a strategy for SQA ISO Standard No strategy for SQA Maybe some templates No templates Yes some re-usables Full process - starts with product management Its complicated Test after code during sprint The director is the best tester

Research Question	Description	High level Category	Related Codes
RQ5: Will this industrialisation improve the effectiveness of the SQA process	Industrialisation will effectiveness of SQA	Without measurement of existing processes, theoretically SQA will improve but not known for certain	Have to meet a contractual target Quality is correct if the customer is happy Ask the customer about CSF Perceived value, defects fix rate, user training Product meets feature and functional requirements Different targets or standards for different employees Do small simple incremental stuff Look for improvements Bug tracking and CRM system provide control ISO 9001 Management System works well Rely on system testing Good at communicating No strategy for training and communication

7.10 Appendix J – Emergent Themes

Table 15 - Emergent Themes - Thematic Structure of Study Results Aligned to Research Questions

RQ	Theme	Subtheme
RQ1 : What costs are incurred by the organisation to achieve the desired quality level?	Quality costs in small businesses are not measured	1a. Regulatory standards used for compliance but not governance 1b. Setting and measuring KPIs is difficult 1c. Small businesses have no budget or KPI for quality cost tracking
RQ2 : Do small software development businesses have a defined SQA process in place?	Small businesses have some elements of an SQA process	2a. Existing processes have some benefits and constraints 2b. Processes, procedures and instructions somewhat used but not much is documented 2c. Requirements are gathered by constant communication with customers 2d. SQA role played by the developer in most cases 2e. The correct quality mainly dependent on the customers satisfaction at handover
RQ3: How effective are the SQA activities in these organisations	SQA in small businesses work but effectiveness is not measured	3a. Effectiveness of strategies are not evaluated in small businesses 3b. No strategy for training and Communication 3c. Some strategies in small businesses are effective processes 3d. ISO 9001 Management System works well
RQ4: How could the SQA process be industrialised in its application	SQA Industrialisation is best applied to testing phase in small businesses	4a. Automation tools used in development, testing and logging 4b. Some differentiation between product type and process type 4c. SQA management strategies vary 4d. System level and testing re-usables 4e. Start with customer requirements 4f. Testing strategy involves developers and management
RQ5: Will this industrialisation improve the effectiveness of the SQA process	Industrialisation will improve effectiveness of SQA	5a. Critical Success Factors based on meeting customer requirements 5b. Hold all employees to the same standard will improve effectiveness 5c. Meeting customer requirements shows effective implementation of SQA 5d. Start industrialisation with bug tracking and a BMS 5e. Training and communication at every level from customer to management to development essential

7.11 Appendix K – Participant Secondary Data

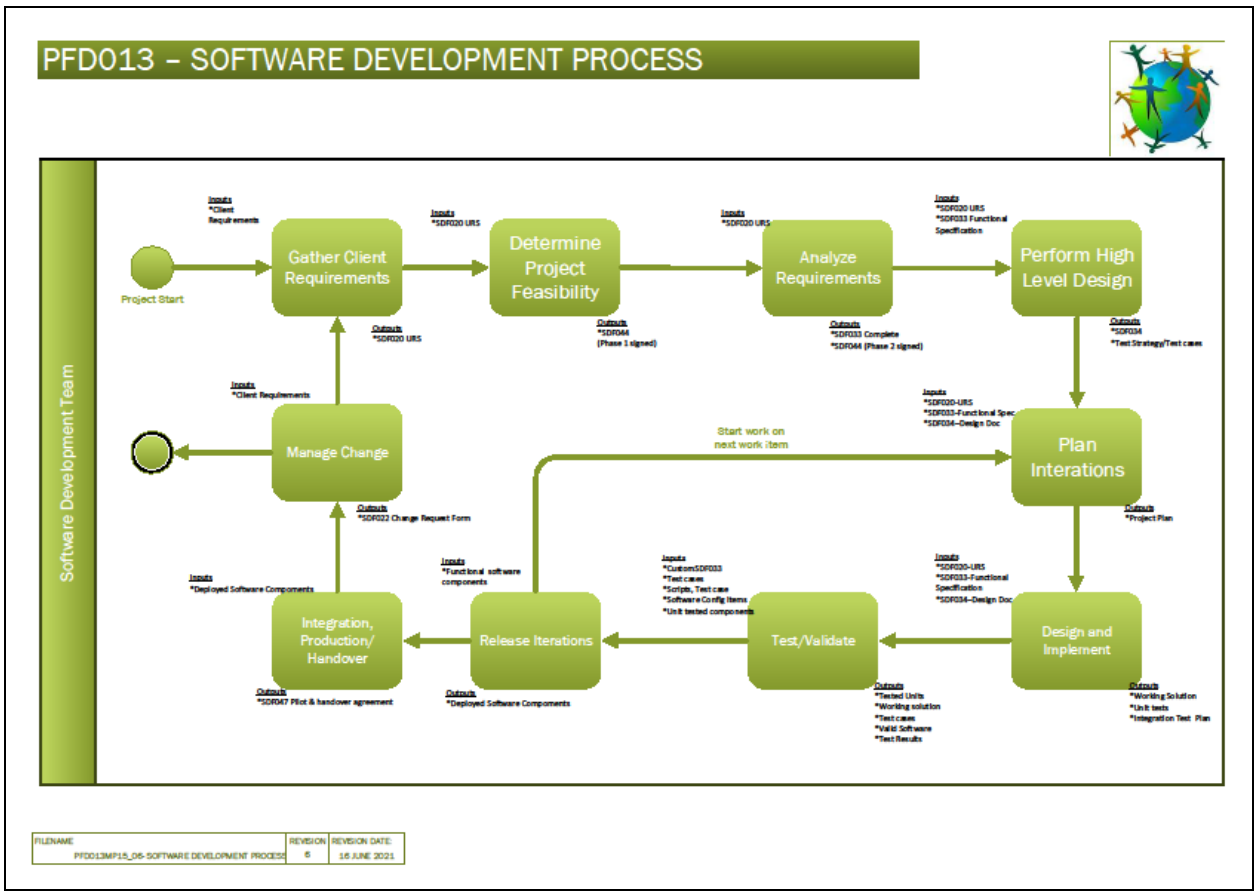


Figure 13 - Org1 secondary data

Source: Research participant

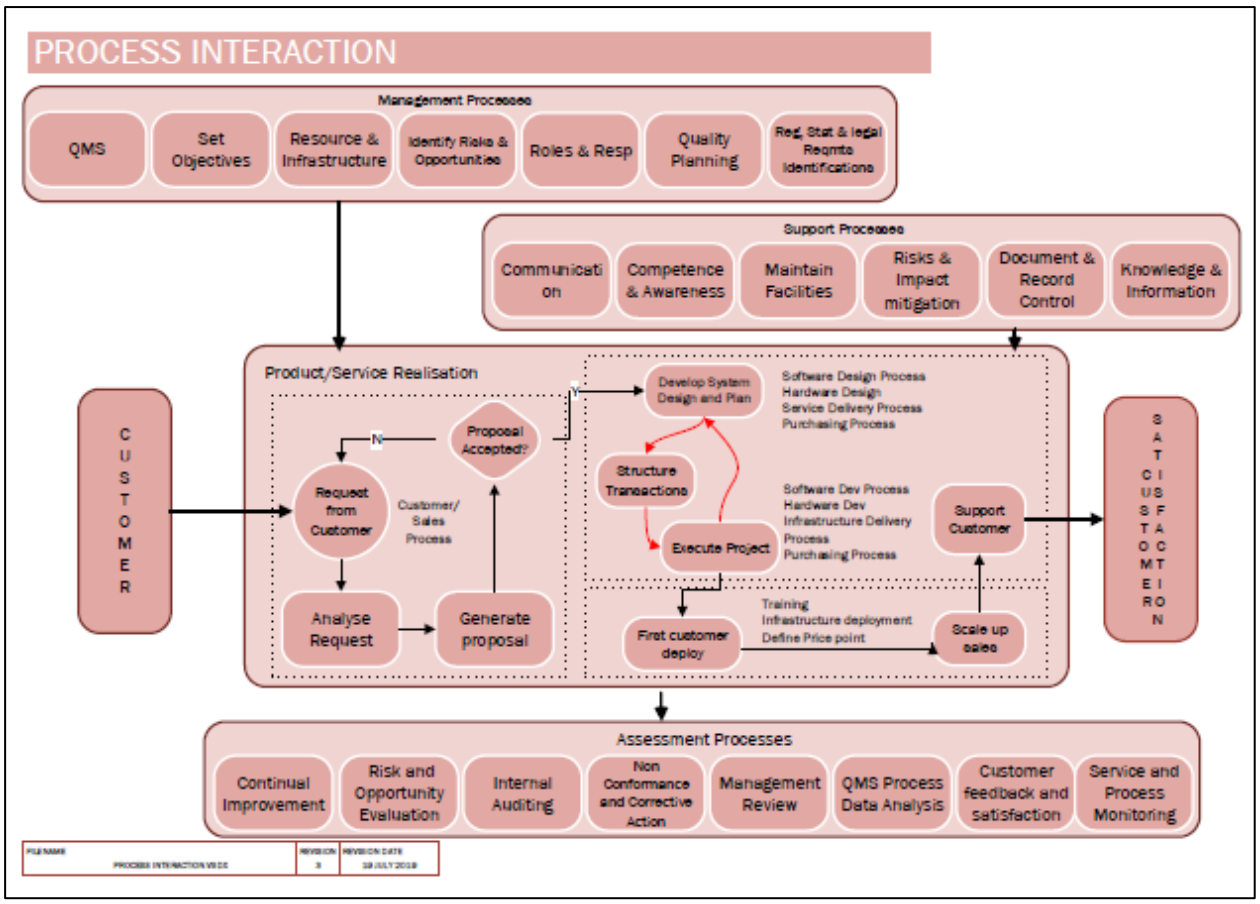


Figure 14 - Org3 secondary data

Source: Research participant

8 REFERENCES

Águila, I., Sagrado, J. and Cañadas, J. 2020. *Software Engineering Timeline: major areas of interest and multidisciplinary trends*.

9001:, I. 2015. *Quality Management Systems - Requirements*. Switzertland: ISO.

Adams, W. C. 2015. Conducting Semi-Structured Interviews. In: Kathryn E. Newcomer, H. P. H., Joseph S. Wholey ed. *Handbook of Practical Program Evaluation*. 4 th edn. Wiley Online Library, 492-505.

Agarwal, R., Nayak, P., Malarvizhi, M., Suresh, P. and Modi, N. 2007. Virtual Quality Assurance Facilitation Model. In: Proceedings of *Proceedings of the International Conference on Global Software Engineering*. 51-59.

Al-Sarayreh, K. T., Meridji, K. and Abran, A. 2021. Software engineering principles: A systematic mapping study and a quantitative literature review. *Engineering Science and Technology, an International Journal*, 24 (3): 768-781.

Alamri, S. and AbdulAziz, A. 2016. The effectiveness of implementing total quality management in software development. *International Advanced Research Journal in Science, Engineering and Technology, 3 (6)*: 206-211.

Ali, K. and Johl, S. K. 2022. Impact of Total Quality Management on SMEs Sustainable Performance in the Context of Industry 4.0. In: Al-Emran, M., Al-Sharafi, M. A., Al-Kabi, M. N. and Shaalan, K. eds. Proceedings of *Proceedings of International Conference on Emerging Technologies and Intelligent Systems*. Cham, 2022//. Springer International Publishing, 608-620.

Antwi, S. K. and Hamza, K. 2015. Qualitative and quantitative research paradigms in business research: A philosophical reflection. *European journal of business and management*, 7 (3): 217-225.

Aronson, J. 1995. A pragmatic view of thematic analysis. *The Qualitative Report*, 2 (1): 1-3.

Arvanitou, E.-M., Ampatzoglou, A., Bibi, S., Chatzigeorgiou, A. and Stamelos, I. 2019. Monitoring Technical Debt in an Industrial Setting. Paper presented at the *Proceedings of the Evaluation and Assessment on Software Engineering*. Copenhagen, Denmark, Association for Computing Machinery, 123–132. Available: https://doi.org/10.1145/3319008.3319019 (Accessed

Baptista, G. and Salles, J. 2012. A Proposal of a Software Development Model Based on Industrial Engineering Concepts. *Applied Mechanics and Materials*, 263-266: 1921-1924.

BITKOM. 2010. *Industrielle Softwareentwicklung Leitfaden und Orientierungshilfe*. Available: https://www.bitkom.org/Bitkom/Publikationen/Leitfaden-Industrielle-Softwareentwicklung.html (Accessed 31 October 2021).

Bogopa, M. E. and Marnewick, C. 2022. Critical success factors in software development projects. *South African Computer Journal*, 34: 1-34.

Booch, G. 2018. The History of Software Engineering. *IEEE Software*, 35 (5): 108-114.

Braun, V. and Clarke, V. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3 (2): 77-101.

Casula, M., Rangarajan, N. and Shields, P. 2021. The potential of working hypotheses for deductive exploratory research. *Quality & Quantity*, 55 (5): 1703-1725.

Chaudhary, M. and Chopra, A. 2017. CMMI Overview. In: *CMMI for Development : Implementation Guide*. Berkeley, CA: Apress, 1-7.

Creswell, J. 2014. *Research Design, Qualitative, Quantitative and Mixed Methods Approaches*. Fourth Edition ed. Sage Publishing.

Daniel, G. 2018. Software Quality Management Standards and Models. In: *Software Quality: Concepts and Practice*. IEEE, 585-616. Available: http://ieeexplore.ieee.org/document/8343644 (Accessed

Daughtery, T. 2013. Software Quality Costs. *Software Quality Professional* vol. 15, no. 2.

Del Águila, I. M., Sagrado, J. d. and Cañadas, J. 2020. Software Engineering Timeline: major areas of interest and multidisciplinary trends. *ArXiv*, abs/2002.10163: 1-12.

Dragos, P. 2021. Overview of the Agile Rational Unified Process (Rup) in the Context of Software Development Projects. *Journal of Business and Economics*: 681.

Drysdale, B. 2022. *Australia's software engineers face an industrialised call-to-action*. Available: https://itbrief.com.au/story/australia-s-software-engineers-face-an-industrialised-call-to-action (Accessed

Galin, D. 2004. *Software quality assurance: from theory to implementation*. Pearson education.

Georgiev, S. and Ohtaki, S. 2020. Critical success factors for TQM implementation among manufacturing SMEs: Evidence from Japan. *Benchmarking: An International Journal*, 27 (2): 473-498.

Gonen, B. and Sawant, D. 2020. Significance of agile software development and SQA powered by automation. In: Proceedings of *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 7-11.

Greenfield, J. and Short, K. 2003. Software factories: assembling applications with patterns, models, frameworks and tools. Paper presented at the *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. Anaheim, CA, USA, Association for Computing Machinery, 16–27. Available: https://doi.org/10.1145/949344.949348 (Accessed

Guest, G., Bunce, A. and Johnson, L. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods*, 18 (1): 59-82.

Hochstein, A., Ebert, N., Uebernickel, F. and Brenner, W. 2007. IT-Industrialisierung: Was ist das? *Computerwoche*,

In on Africa. 2018. *An Assessment of South Africa's SME Landscape*. Available: https://smesouthafrica.co.za/sme-resources/an-assessment-of-south-africas-sme-landscape-challenges-opportunities-risks-next-steps-report-2018-2019/ (Accessed 4 April 2023).

Institute of Electrial and Electronic Engineers (IEEE). 2017. *ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary*. IEEE.

Institute of Electrical and Electronic Engineers (IEEE). 2014. *IEEE Standard for Software Quality Assurance Processes IEEE Std 730-2014*. New York, USA: IEEE.

International Standards Organization (ISO). 2016. *Systems and Software Engineering - Lifecycle profiles for Very small Entities - ISO 29110*. ISO/IEC.

Isaacs, A. N. 2014. An overview of qualitative research methodology for public health researchers. *International Journal of Medicine and Public Health*, 4 (4)

Issac, G., Rajendran, C. and Anantharaman, R. N. 2004. A Conceptual Framework for Total Quality Management in Software Organizations. *Total Quality Management & Business Excellence*, 15 (3): 307-344.

Itkonen, J., Rautiainen, K. and Lassenius, C. 2005. Towards Understanding Quality Assurance in Agile Software Development. 8

Jacobson, I., Spence, I. and Seidewitz, E. 2016. Industrial-scale agile: from craft to engineering. *Communications of the ACM*, 59 (12): 63–71.

Jain, N. 2021. Survey versus interviews: Comparing data collection tools for exploratory research. *The Qualitative Report*, 26 (2): 541-554.

Janes, A., Lenarduzzi, V. and Stan, A. C. 2017. A Continuous Software Quality Monitoring Approach for Small and Medium Enterprises. Paper presented at the *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*. L'Aquila, Italy, Association for Computing Machinery, 97–100.

John, B., Kadadevaramath, R. and Immanuel, E. A. 2016. Recent Advances in Software Quality Management: A. *Merit Research Journal of Business and Management*, 4 (3)

Juran, J. M. and De Feo, J. A. 2010. *Juran's quality handbook: the complete guide to performance excellence*. McGraw-Hill Education.

Kalpande, S. D., Gupta, R. C. and Dandekar, M. D. 2012. Framework of TQM implementation in small businesses of an industrially backward region: A concept. *Performance Improvement*, 51 (7): 7-13.

Kambhampati, V. S. S. N. R. 2017. Principles of Industrial Engineering. *IIE Annual Conference. Proceedings*: 890-895.

Kapoulas, A. and Mitic, M. 2012. Understanding challenges of qualitative research: rhetorical issues and reality traps. *Qualitative Market Research: An International Journal*, 15 (4): 354-368.

Krasner, H. 2018. The cost of poor quality software in the US: A 2018 report. *Consortium for IT Software Quality, Tech. Rep*, 10

Krasner, H. 2021. *The Cost of Poor Software Quality in the US: A 2020 Report*. Available: https://www.it-cisq.org/ (Accessed

Krasner, H. 2022. *The Cost of Poor Software Quality in the US: A 2022 Report*. Available: www.it-cisq.org (Accessed

Lagstedt, A., Dirin, A. and Williams, P. 2021. Quality Practitioners' Differing Perspectives on Future Software Quality Assurance Practices. *International Journal of Interactive Mobile Technologies (iJIM)*, 15 (24): pp. 134-154.

Laporte, C. 2015. *The New ISO/IEC 29110 Systems Engineering International Standard for Very Small Entities*. Paris: First Systems Engineering Workshop of the EMEA Sector of INCOSE.

Laporte, C. and O'Connor, R. 2017. Software process improvement standards and guides for very small organization: An overview of eight implementations. *CrossTalk, The Journal of Defense Software Engineering*, 30 (3): 23-27.

Laporte, C. Y. and April, A. 2018. *Software quality assurance*. John Wiley & Sons.

Larrucea, X., Connor, R. V. O., Colomo-Palacios, R. and Laporte, C. Y. 2016. Software Process Improvement in Very Small Organizations. *IEEE Software*, 33 (2): 85-89.

Lárusdóttir, M. K., Cajander, Å. and Simader, M. 2014. Continuous improvement in agile development practice: The case of value and non-value adding activities. In: Proceedings of *Human-Centered Software Engineering: 5th IFIP WG 13.2 International Conference, HCSE 2014, Paderborn, Germany, September 16-18, 2014. Proceedings 5*. Springer, 57-72.

Magd, H. and Karyamsetty, H. 2021. Organizational Sustainability and TQM in SMEs: A Proposed Model. *European journal of business and management*, 13 (4): 88-96.

Magodi, A. Y., Daniyan, I. A. and Mpofu, K. 2022. An Investigation of the Effect of the ISO 9001 Quality Management System on Small and Medium Enterprises in Gauteng, South Africa. *South African Journal of Industrial Engineering*, 33 (1): 126-138.

Matkovic, P. and Tumbas, P. 2010. A Comparative Overview of the Evolution of Software Development Models. *Journal of Industrial Engineering and Management*, 1: 163-172.

Maxim, B. R. and Kessentini, M. 2016. Chapter 2 - An introduction to modern software quality assurance. In: Mistrik, I., Soley, R., Ali, N., Grundy, J. and Tekinerdogan, B. eds. *Software Quality Assurance*. Boston: Morgan Kaufmann, 19-46. Available: https://www.sciencedirect.com/science/article/pii/B9780128023013000028 (Accessed

Miguel, J. P., Mauricio, D. and Rodríguez, G. 2014. A review of software quality models for the evaluation of software products. *arXiv preprint arXiv:1412.2977*,

Miguel, P. A. C. 2005. Modularity in product development: a literature review towards a research agenda. *Product: Management and Development*, 3 (2): 165-174.

Mistrík, I., Soley, R. M., Ali, N., Grundy, J. and Tekinerdogan, B. 2015. *Software quality assurance: in large scale and complex software-intensive systems*. Morgan Kaufmann.

Mohammed, A. S. A., Tibek, S. R. H. and Endot, I. 2013. The Principles of Total Quality Management System in World Islamic Call Society. *Procedia - Social and Behavioral Sciences*, 102: 325-334.

Nenty, H. J. 2009. Writing a Quantitative Research Thesis. *International Journal of Educational Sciences*, 1 (1): 19-32.

News24. 2022. *The essential role of SMEs in the economy*. Available: https://www.news24.com/fin24/PartnerContent/the-essential-role-of-smes-in-the-economy-20220510 (Accessed

Ngulube, P. 2015. Qualitative data analysis and interpretation: systematic search for meaning. *Addressing research challenges: making headway for developing researchers*, 131: 156.

O'Connor, R. V. 2019. Software development process standards for very small companies. In: *Advanced Methodologies and Technologies in Digital Marketing and Entrepreneurship*. IGI Global, 681-694.

O'Leary, Z. 2021. *The essential guide to doing your research project*. Forth ed. London, UK: Sage.

O'Connor, R. and Laporte, C. 2014. An Innovative Approach to the Development of an International Software Process Lifecycle Standard for Very Small Entities. *International Journal of Information Technologies and Systems Approach*, 7: 1-22.

Okumoto, K., Mijumbi, R. and Asthana, A. 2018. Software Quality Assurance. In: *Telecommunication Networks-Trends and Developments*. IntechOpen.

Palinkas, L. A., Horwitz, S. M., Green, C. A., Wisdom, J. P., Duan, N. and Hoagwood, K. 2015. Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Administration and policy in mental health*, 42 (5): 533-544.

Pham, L. T. M. 2018. Qualitative approach to research a review of advantages and disadvantages of three paradigms: Positivism, interpretivism and critical inquiry. *University of Adelaide*,

Prenner, N., Unger-Windeler, C. and Schneider, K. 2021. Goals and challenges in hybrid software development approaches. *Journal of Software: Evolution and Process*, 33 (11): e2382.

Pusatli, O. T. and Misra, S. 2011. A discussion on assuring software quality in small and medium software enterprises: An empirical investigation. *Technical Gazette*, 18 (3): 447-452.

Rowlands, B. H. 2005. Grounded in practice: Using interpretive research to build theory. *Electronic Journal of Business Research Methods*, 3 (1): pp81-92-pp81-92.

Ruparelia, N. B. 2010. Software development lifecycle models. *SIGSOFT Softw. Eng. Notes*, 35 (3): 8–13.

Saha, P., Talapatra, S., Belal, H. M. and Jackson, V. 2022. Unleashing the Potential of the TQM and Industry 4.0 to Achieve Sustainability Performance in the Context of a Developing Country. *Global Journal of Flexible Systems Management*, 23 (4): 495-513.

Sahoo, S. and Yadav, S. 2018. Total Quality Management in Indian Manufacturing SMEs. *Procedia Manufacturing*, 21: 541-548.

Saunders, M., Lewis, P. and Thornhill, A. 2009. *Research methods for business students*. Fifth ed. Lebanon, Indiana, U.S.A: Prentice Hall.

Sfakianaki, E. and Kakouris, A. P. 2020. Obstacles to ISO 9001 certification in SMEs. *Total Quality Management & Business Excellence*, 31 (13-14): 1544-1564.

Shrestha, B. 2016. Best QMEs for measurement of Software quality for SMEs. Itä-Suomen yliopisto.

Singh, B. and Kannojia, S. P. 2013. A review on software quality models. In: Proceedings of *2013 International Conference on Communication Systems and Network Technologies*. IEEE, 801-806.

Sommerville, I. 2016. *Software engineering*. 10th ed. Harlow, England: Pearson.

South Africa, Department of Small Business Development,. 2019. *Revised Schedule 1 of the National Definition of Small Enterprise in South Africa*. Cape Town: Government Printer: Government Gazette 42304. Available: https://www.gov.za/sites/default/files/gcis_document/201903/423041gon399.pdf (Accessed 4 April 2023).

Sowunmi, O. Y., Misra, S., Fernandez-Sanz, L., Crawford, B. and Soto, R. 2016. An empirical evaluation of software quality assurance practices and challenges in a developing country: a comparison of Nigeria and Turkey. *SpringerPlus*, 5 (1): DOI: 10.1186/s40064-40016-43575-40065.

Sultana, N., Syeed, M. and Fatema, K. 2020. An empirical investigation on quality assurance practices in software industries: Bangladesh perspective. *International Journal of Software Engineering and Computer Systems*, 6 (2): 1-10.

Toke, L. K. and Kalpande, S. D. 2020. Total quality management in small and medium enterprises: An overview in Indian context. *Quality Management Journal*, 27 (3): 159-175.

Walkinshaw, N. 2017. *Software quality assurance*. 1 ed. Gewerbestrasse, Switzerland: Springer Cham.

Watson, G. H. and DeYong, C. F. 2010. Design for Six Sigma. *International Journal of Lean Six Sigma*, 1 (1): 66-84.

Weckenmann, A., Akkasoglu, G. and Werner, T. 2015. Quality management – history and trends. *The TQM Journal*, 27 (3): 281-293.

Wieczorek, M., Vos, D. and Bons, H. 2014. *Systems and Software Quality*. First ed. Berlin, Heidelberg: Springer.

Woiceshyn, J. and Daellenbach, U. 2018. Evaluating inductive vs deductive research in management studies. *Qualitative Research in Organizations and Management: An International Journal*, 13 (2): 183-195.

Wong, W. Y., Yu, S. W. and Too, C. W. 2018. A Systematic Approach to Software Quality Assurance: The Relationship of Project Activities within Project Life Cycle and System Development Life Cycle. In: Proceedings of *2018 IEEE Conference on Systems, Process and Control (ICSPC)*. 14-15 Dec. 2018. 123-128.

Xie, Z., Hall, J., McCarthy, I. P., Skitmore, M. and Shen, L. 2016. Standardization efforts: The relationship between knowledge dimensions, search processes and innovation outcomes. *Technovation*, 48-49: 69-78.

Yahaya, J., Tareen, A., Deraman, A. and Razak, A. 2017. Software Quality and Productivity Model for Small and Medium Enterprises. *International Journal of Advanced Computer Science and Applications*, 8 (5): 316-320.

Zhou, B. 2016. Lean principles, practices, and impacts: a study on small and medium-sized enterprises (SMEs). *Annals of Operations Research*, 241 (1): 457-474.

Yahaya, J., Tareen, A., Deraman, A. and Razak, A. 2017. Software Quality and Productivity Model for Small and Medium Enterprises. *International Journal of Advanced Computer Science and Applications*, 8