


**EVALUATING TIME, SCOPE AND COST OF HUMAN
LABOUR VERSUS DIGITAL PROCESS AUTOMATION
CAPABILITIES**

J MULLER

(Student number: 26058995)

 orcid.org/0000-0003-2709-5444



Dissertation accepted in fulfillment of the requirements for the degree

Master of Science

in Computer Science

in the School of Computer Science and Information Systems in the Faculty
of Natural and Agricultural Sciences
at the North-West University
(Vaal Campus)

Supervisors: Dr WH van Blerk
Dr JJ Greeff

Graduation: May 2022

To whom it may concern

Cecile van Zyl
Language editing and translation
Cell: 072 389 3450
Email: Cecile.vanZyl@nwu.ac.za

Dear Mr / Ms

Re: Language editing of dissertation (Evaluating time, scope and cost of human labour versus digital process automation capabilities)

I hereby declare that I language edited, reference-checked and formatted the above-mentioned dissertation by Ms Jacqui Muller (student number: 26058995).

Please feel free to contact me should you have any enquiries.

Kind regards

A handwritten signature in black ink, appearing to read 'Cecile van Zyl', with a large loop at the top.

Cecile van Zyl

Language practitioner

BA (PU for CHE); BA honours (NWU); MA (NWU)
SATI number: 1002391

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	2
1.1 Introduction.....	2
1.2 Background information.....	3
1.3 Problem statement	5
1.3.1 Scope	7
1.3.2 Time	7
1.3.3 Cost.....	8
1.4 Research questions.....	8
1.5 Research objectives	9
1.5.1 Primary objective	9
1.5.2 Secondary objectives.....	9
1.5.3 Theoretical objectives.....	10
1.5.4 Empirical objectives.....	11
1.6 Dissertation statement.....	12
1.7 Delimitation of field of study.....	12
1.8 Underlying assumptions	13
1.9 Significance of the study.....	13
1.10 Chapter layout	14
1.11 Study timeline.....	16
1.12 Conclusion.....	18
CHAPTER 2: LITERATURE REVIEW	19

2.1	Introduction.....	19
2.2	Automation	19
2.2.1	Advantages of automation	20
2.2.2	Disadvantages of automation	20
2.3	Automation evolution	21
2.4	Automation lifecycle.....	22
2.5	Process analysis.....	23
2.5.1	Feasibility and complexity analysis	24
2.5.2	Process mining and discovery	24
2.5.3	Process modelling	25
2.6	Automation process documentation.....	26
2.6.1	Process definition document (PDD).....	26
2.6.2	Solution design document (SDD).....	27
2.7	Triple constraint model	27
2.8	Conclusion.....	28
CHAPTER 3: RESEARCH METHODOLOGY		30
3.1	Introduction.....	30
3.2	Methodology	30
3.2.1	Philosophy: Pragmatism	31
3.2.2	Approach: Inductive	33
3.3	Strategy: Experiments within case study research.....	33
3.3.1	Case study environment.....	34

3.4	Choice: Design science research (DSR).....	39
3.4.1	The open group architecture framework (TOGAF).....	40
3.5	Data collection.....	41
3.5.1	Experiment 1: Human process execution data collection.....	44
3.5.2	Experiment 2: DPA solution process execution data collection.....	44
3.6	Participants.....	44
3.7	Research instruments.....	46
3.7.1	Task capture.....	46
3.7.2	Joint application development (JAD) workshops.....	46
3.7.3	GAP analysis.....	47
3.8	Data analysis.....	47
3.9	Validity and reliability.....	50
3.10	Limitations of study.....	50
3.11	Ethical considerations.....	51
3.12	Conclusion.....	52
CHAPTER 4: HUMAN EXECUTION DATA COLLECTION		54
4.1	Introduction.....	54
4.2	Data collection.....	55
4.2.1	Time.....	57
4.2.2	Cost.....	64
4.2.3	Scope.....	70
4.3	Conclusion.....	71

CHAPTER 5: INITIAL CONCEPTUAL DESIGN AND BUILD	73
5.1	Introduction..... 73
5.2	Overview 75
5.3	Business layer 76
5.3.1	Organisational overview 76
5.3.2	Strategy 77
5.3.3	Operating model 77
5.4	Information systems layer 81
5.4.1	Application architecture 81
5.4.2	Data architecture 85
5.5	Technology layer 89
5.5.1	High-level architecture 90
5.5.2	Master project runtime details 91
5.6	Report prototype 91
5.7	Joint application development (JAD) workshop..... 94
5.7.1	What went well? 94
5.7.2	What could have been done better? 94
5.7.3	What was not done correctly?..... 95
5.8	Conclusion..... 95
CHAPTER 6: PROPOSED CONCEPTUAL DESIGN AND BUILD	97
6.1	Introduction..... 97
6.2	Business layer 98

6.2.1	Operating model	98
6.3	Information systems layer	101
6.3.1	Application architecture	101
6.3.2	Data architecture	110
6.4	Technology layer	113
6.5	Report prototype	114
6.6	Joint application development (JAD) workshop.....	117
6.7	Data collection	117
6.7.1	Time	119
6.7.2	Cost.....	121
6.7.3	Scope	123
6.8	Conclusion.....	124
CHAPTER 7: DATA ANALYSIS		126
7.1	Introduction.....	126
7.2	Time analysis.....	127
7.3	Cost analysis	129
7.3.1	Additional costs	132
7.4	Scope analysis	142
7.5	Discussion on analysis	143
7.6	Conclusion.....	146
CHAPTER 8: CONCLUSION.....		148
8.1	Introduction.....	148

8.2	Summary of results.....	148
8.2.1	Time	150
8.2.2	Cost.....	150
8.2.3	Scope	151
8.3	Summary of study.....	152
8.3.1	Chapter 1: Introduction	153
8.3.2	Chapter 2: Literature review.....	154
8.3.3	Chapter 3: Research methodology	154
8.3.4	Chapter 4: Human execution data collection.....	154
8.3.5	Chapter 5: Intial conceptual design and build	155
8.3.6	Chapter 6: Proposed conceptual design and build	156
8.3.7	Chapter 7: Data analysis	156
8.3.8	Research objectives	157
8.3.9	Research questions.....	160
8.4	Limitations and future work.....	163
8.5	Conclusion.....	165
	REFERENCES.....	167
	ANNEXURE A: PROCESS ASSESSMENT	176
	ANNEXURE B: PROCESS DEFINITION DOCUMENT	177
	Introduction	177
	Objectives	177

Minimum prerequisites for the automation.....	177
AS IS process 178	
AS IS process overview.....	178
AS IS process applications used	179
High-level AS IS process flow.....	179
Detailed AS IS process steps	180
Input data description	183
TO BE process.....	184
Detailed TO BE process map	184
In-scope for automation.....	184
Out-of-scope for automation	185
Exception handling	185
Reporting	186
Process documentation	186
ANNEXURE C: SOLUTION DESIGN DOCUMENT	187
Introduction	187
Objectives	187
Development guide	187
Architectural structure	187
Detailed process definition	190
Data flow diagram	193
Database design	194

Project details 195

ANNEXURE D: PROCESS UPGRADE INFORMATION (PROCESS CONFIGURATION) ... 196

ANNEXURE E: PROCESS UPGRADE INFORMATION (SCRIPT LIBRARIES)..... 198

ANNEXURE F: PROCESS UPGRADE INFORMATION (PACKAGE VERSIONS)..... 202

ANNEXURE G: ETHICAL CLEARANCE..... 206

ANNEXURE H: CODE OF CONDUCT 208

LIST OF FIGURES

Figure 1-1:	Design science research within the context of this study	3
Figure 1-2:	Study overview.....	5
Figure 1-3:	Reasoning in the design science research cycle (Kuechler & Vaishnavi, 2008).....	14
Figure 1-4:	Chapter layout in terms of DSR.....	15
Figure 1-5:	High-level study timeline	17
Figure 1-6:	Build phase timeline	17
Figure 2-1:	The timeline of automation history.....	22
Figure 2-2:	UiPath automation lifecycle	23
Figure 2-3:	Chapter 2 DSR study layout contribution.....	29
Figure 3-1:	The research onion, adapted from Saunders <i>et al.</i> (2012).....	31
Figure 3-2:	Full scaled agile framework (Leffingwell, 2018)	35
Figure 3-3:	The end-to-end automation delivery model	36
Figure 3-4:	The open group architecture framework (TOGAF) (Josey, 2016)	41
Figure 3-5:	Experiments within this study	42
Figure 3-6:	High-level process followed by the human and DPA solution	43
Figure 3-7:	The determination of variables H, D and C in terms of c, t and s	48
Figure 3-8:	Chapter 3 DSR study layout contribution.....	53
Figure 4-1:	The process steps to be executed manually (AS IS process)	54
Figure 4-2:	The determination of variable, H, in terms of c, t and s for human execution	56
Figure 4-3:	The duration (in seconds) by run (in terms of phase)	61
Figure 4-4:	The duration (in seconds) by employee and run (in terms of phase)	62

Figure 4-5: The duration (in seconds) by employee (in terms of phase) 63

Figure 4-6: Total direct cost per run and rate per hour by employee..... 67

Figure 4-7: Total direct and indirect cost by employee (per run) 70

Figure 4-8: Chapter 4 DSR study layout contribution..... 72

Figure 5-1: The open group architecture framework (TOGAF) architecture development method (ADM) (Graves, 2013) 74

Figure 5-2: Process steps 75

Figure 5-3: Case study environment operating model 78

Figure 5-4: The end-to-end automation delivery model (lean study implementation) 79

Figure 5-5: Detailed AS IS process 82

Figure 5-6: Detailed TO BE process..... 83

Figure 5-7: Package information retrieval process steps 84

Figure 5-8: Latest package version retrieval process steps 84

Figure 5-9: Upgrade package DSS process steps..... 85

Figure 5-10: Database design 86

Figure 5-11: Data flow diagram 88

Figure 5-12: Technical landscape 89

Figure 5-13: High-level architecture 90

Figure 5-14: Auto-upgrade process summary tab: Power BI report 92

Figure 5-15: Auto-upgrade process upgrades tab: Power BI report..... 92

Figure 5-16: Auto-upgrade process upgrades interactive analytics 93

Figure 5-17: Chapter 5 DSR study layout contribution..... 96

Figure 6-1: Case study environment operating model 99

Figure 6-2:	Detailed AS IS process	102
Figure 6-3:	Detailed TO BE process.....	106
Figure 6-4:	Package information retrieval process steps	107
Figure 6-5:	Latest package version retrieval process steps	108
Figure 6-6:	Upgrade package DSS process steps.....	109
Figure 6-7:	Database design	110
Figure 6-8:	Data flow diagram	112
Figure 6-9:	Technical landscape	113
Figure 6-10:	Auto-upgrade process last run summary tab: Power BI report	115
Figure 6-11:	Auto-upgrade process upgrades tab: Power BI report.....	116
Figure 6-12:	The determination of variables D in terms of c, t and s for DPA execution .	118
Figure 6-13:	The duration (in seconds) of DPA execution by phase	121
Figure 6-14:	Chapter 6 DSR study layout contribution.....	125
Figure 7-1:	The determination of variables C in terms of c, t and s for human execution	126
Figure 7-2:	The duration (in seconds) by phase (per annum)	129
Figure 7-3:	The cost (in rands) by execution method (per annum)	131
Figure 7-4:	Requirements gathering cost and diminishing cost per annum.....	134
Figure 7-5:	Requirements gathering cost and diminishing cost per run	135
Figure 7-6:	Sunken cost and diminishing cost per annum	137
Figure 7-7:	Sunken cost and diminishing cost per run	138
Figure 7-8:	Development cost and diminishing cost per annum.....	140
Figure 7-9:	Development cost and diminishing cost per run	140

Figure 7-10:	The determination of variables A, B and C in terms of X, Y and Z.....	144
Figure 7-11:	Chapter 7 DSR study layout contribution.....	147
Figure 8-1:	Design science research within the context of this study.....	152
Figure 8-2:	Chapter layout in terms of DSR.....	153
Figure 8-3:	Chapter 8 DSR study layout contribution.....	166

LIST OF TABLES

Table 3-1:	Tabular representation of analytical calculations to be performed	49
Table 3-2:	Tabular representation of the validity and reliability	50
Table 4-1:	Tabular representation of analytical calculations to be performed while determining A through human execution	56
Table 4-2:	Tabular representation of human process execution time (in seconds)	60
Table 4-3:	Tabular representation of human process execution time	64
Table 4-4:	Tabular representation of human process execution direct cost	66
Table 4-5:	Tabular representation of human process execution indirect cost	69
Table 4-6:	Tabular representation of average human process execution scope	71
Table 5-1:	Tabular representation of the gap analysis matrix	80
Table 5-2:	Tabular representation of the gap analysis	81
Table 5-3:	Tabular representation of the DPA solution's technical aspects	91
Table 6-1:	Tabular representation of the changes to the initial conceptual design and build proposed	97
Table 6-2:	Tabular representation of the gap analysis matrix	100
Table 6-3:	Tabular representation of the detailed AS IS process steps	105
Table 6-4:	Tabular representation of the AS IS process input data description	105
Table 6-5:	Tabular representation of the known and unknown exceptions	109
Table 6-6:	Tabular representation of the JAD workshop feedback	117
Table 6-7:	Tabular representation of analytical calculations to be performed while determining D through DPA execution	118
Table 6-8:	Tabular representation of DPA solution process execution time	120

Table 6-9:	Tabular representation of the indirect cost of the DPA solution implementation.....	122
Table 6-10:	Tabular representation of the use cases depicted by the journey through the process flow	124
Table 7-1:	Tabular representation of analytical calculations to be performed while determining A through human execution	127
Table 7-2:	Tabular representation of average runtime per run (in seconds)	128
Table 7-3:	Tabular representation of average cost per run (in rands)	130
Table 7-4:	Tabular representation of additional cost analysis	141
Table 7-5:	Tabular representation of number of runs to break even	142
Table 7-6:	Tabular representation of average number of steps per run	143
Table 7-7:	Tabular representation of analytical calculations to be performed	145
Table 8-1:	Tabular representation of collected and analysed data	149
Table 8-2:	Tabular representation of the primary and secondary research questions..	163

DEFINITION OF TERMS AND CONCEPTS

The definitions of terms and concepts are detailed in the table below:

Term	Description
DPA	Digital Process Automation
RPA	Robotic Process Automation
4IR/5IR	The Fourth Industrial Revolution/ The Fifth Industrial Revolution
ICT	Information and Communication Technology
PDD	Process Definition Document
SDD	Solution Definition Document
SAFe	Scaled Agile Framework
IDE	Integrated Development Environment
DSR	Design Science Research
TOGAF	The Open Group Architecture Framework
ADM	Architecture Development Method
AI	Artificial Intelligence
QA	Quality Assurance
DSS	Decision Support System
IS	Information System
JAD	Joint Application Development
SME	Subject Matter Expert
CoE	Center of Excellence
GTM	Go to Market
BSS	Business Support Services
NuGet	A tool that allows developers to make use of reusable code in the form of libraries and packages.
Pip	A tool that allows Python developers to make use of reusable code in the form of libraries and packages.
AS IS	The current state of the process.
TO BE	The desired future state of a process.
DevOps	Development and operations: Refer to the practice of combining software development and IT operations to shorten the systems development lifecycle.
Library/ Package	Library and package are used interchangeably, within the context of this study. Both represent a reusable section of code that gets published to a code repository.
Rollback	The process of undoing steps that have been executed, resulting in an error.

CHAPTER 1: INTRODUCTION

1.1 Introduction

The Fourth Industrial Revolution (4IR) is upon us and with it comes automation, which is the art of making machines perform the highly repetitive and non-complex tasks that people are faced with daily. The aim of this research is to identify how the difference of executing the same process between human execution and the execution of a digital process automation (DPA) solution can be evaluated, specifically in terms of cost, time and scope. Time, cost and scope make up the triple constraint model. The theory behind the triple constraint model, as presented by Baratta (2006), suggests that only two of the constraints may be optimised at a time and that each of the constraints impact one another. The gathering of these constraints forms part of the process analysis phase, and the ability to accurately measure them once the automation is in place, allows the actual value of the automation to be realised. In order to measure the cost, time and scope, the DPA solution needs to be built according to the steps outlined in the project documentation, which will be introduced in Chapter 2 (sub-sections 2.5 and 2.6) with more specific reference to the appropriate annexures. Once the solution has been built, the following will be obtained:

- The cost, scope and time of the human executing a process
- The cost, scope and time of the DPA solution executing a process
- The result of comparison between the different execution methods (human execution and DPA solution execution), in terms of cost, scope and time

Each of the abovementioned aspects has multiple influencing factors that may vary between case studies, but could be grouped through the use of similar terms or parameters. Identifying these parameters and how they can be evaluated will be the focus of this study, applied within a case study as a proof of concept.

Iyawa *et al.* (2016) define design science research (DSR) as a methodology that focuses on creating new knowledge with the purpose of changing existing situations into preferred situations. The intention of using DSR, as suggested by Lukka (2003), is to solve real-life problems by making a contribution to the applied theory presented throughout the conducted research. Hevner (2007) strongly supports the use of DSR to introduce new knowledge through the invention and innovation of theories and artefacts, executed in three cycles, namely relevance, design and rigour. The model proposed by Hevner *et al.* (2004) has been amended, as seen in Figure 1-1, to represent the position and layout that this study will follow.

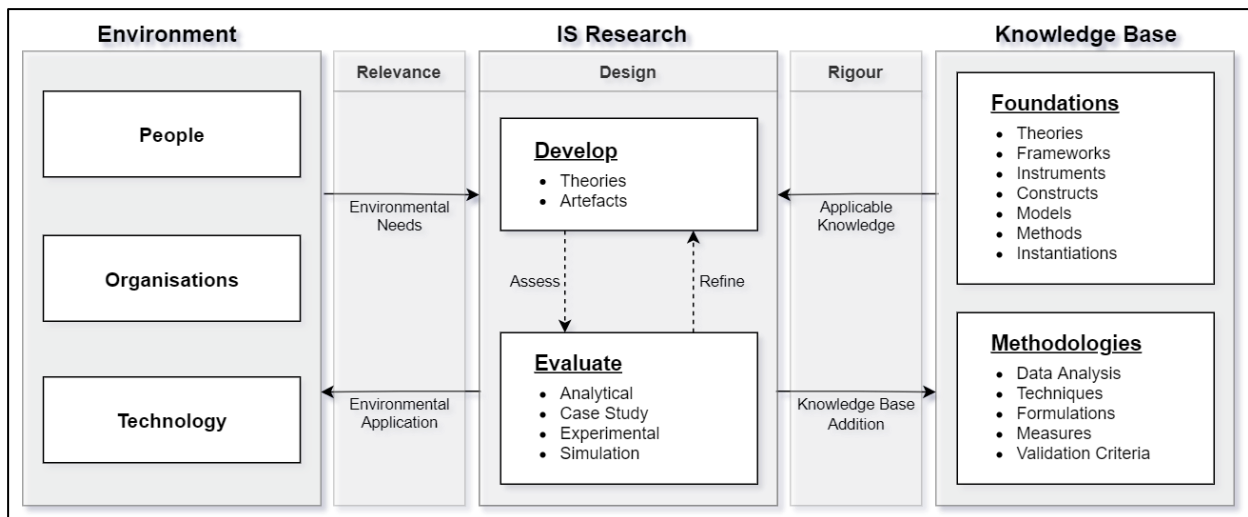


Figure 1-1: Design science research within the context of this study

Figure 1-1 illustrates how the development of theories and artefacts will take in all environmental needs and requirements as inputs, utilising any applicable knowledge from the available knowledge base. The iterative design cycles will be evaluated in each iteration, while outputting the environmental application and knowledge base additions to the environment and knowledge base, respectively. The relevance cycle will represent the flow of all environmental needs and environmental application between the environment and the information system (IS) research. The rigour cycle will represent the flow of knowledge between the knowledge base and the IS research.

Sub-section 1.2 will provide background information regarding this study. Sub-section 1.3 explains the problem statement; sub-section 1.4 addresses the research questions; sub-section 1.5 expands upon the research objectives considered while conducting this study; sub-section 1.6 discusses the problem statement of the dissertation; sub-section 1.7 elaborates on the limitations of this study; sub-section 1.8 explores the assumptions of the study; sub-section 1.9 elaborates on the significance of the study. sub-section 1.10 provides a chapter layout of the dissertation; and sub-section 1.11 outlines the study timeline.

1.2 Background information

The Fourth Industrial Revolution is an emerging chapter in human development that potentially represents a fundamental change in our lives (World Economic Forum, 2020). The World Economic Forum (2020) goes on to suggest that the Fourth Industrial Revolution could utilise the

opportunity to look beyond technology and find ways to give people the opportunity to positively impact society.

Acemoglu and Restrepo (2019) define automation as the use of technology to enable substituting capital for human labour, which is supported by Smith and Fressoli (2021), who expand on this by proposing automation as a supporting element of the 4IR. Van der Aalst *et al.* (2018) explain that there are different types of automation, one of which they identify as process automation, which contains other subsets of automation types. Digital process automation (DPA) may be considered as one of the subsets of process automation. DPA is a term used to describe a digital toolset that performs a process in order to complete a workflow or function (Koplowitz & Rymer, 2019). Through the use of DPA solutions, focus can be shifted from manually executing automatable tasks to enriching the lives of others.

The main goal of this study would be to understand how the capability of a DPA solution and human capability could be compared in terms of time, scope and cost, while evaluating the variables (or parameters) that influence each of the constraints. The identification of parameters, evaluation of the constraints and measurement of these factors will result from an automation opportunity provided through a case study.

The problem (or automation opportunity) presented by the case study involves manually identifying the libraries used within a solution (for multiple solutions) along with their latest available version. Once the libraries that are due for upgrade have been identified, the libraries need to upgrade, and the upgrades should be tested on testable solutions. The list of testable solutions is predefined through the filtering of certain requirements that make testing possible and appropriate. Conducting tests that do not provide value in determining the success of an upgrade or testing that results in negatively impacting the output of a solution would deem the solution infeasible for testing.

In a case where the upgrade has caused the solution to malfunction, the upgrade needs to be undone or 'rolled back' to ensure that the solution remains in working condition until the reason for the malfunction caused by the library can be investigated and fixed.

Rollbacks refer to the undoing (or reverting) of steps which resulted in an error, during a process run. It is important that all systems contain a good rollback or exception handling strategy that mitigate the risk of compromising a production environment or production data in cases where an entire process could not be executed from end-to-end due to errors or exceptions that occurred within the process run. The purpose of a rollback strategy, in this case environment, is aimed at maintaining the state of the environment to ensure that the automated process does not leave the

environment in a state that will result in errors and exceptions caused due to unstable environment parameters (like libraries that have deprecated functions, causing processes to fail).

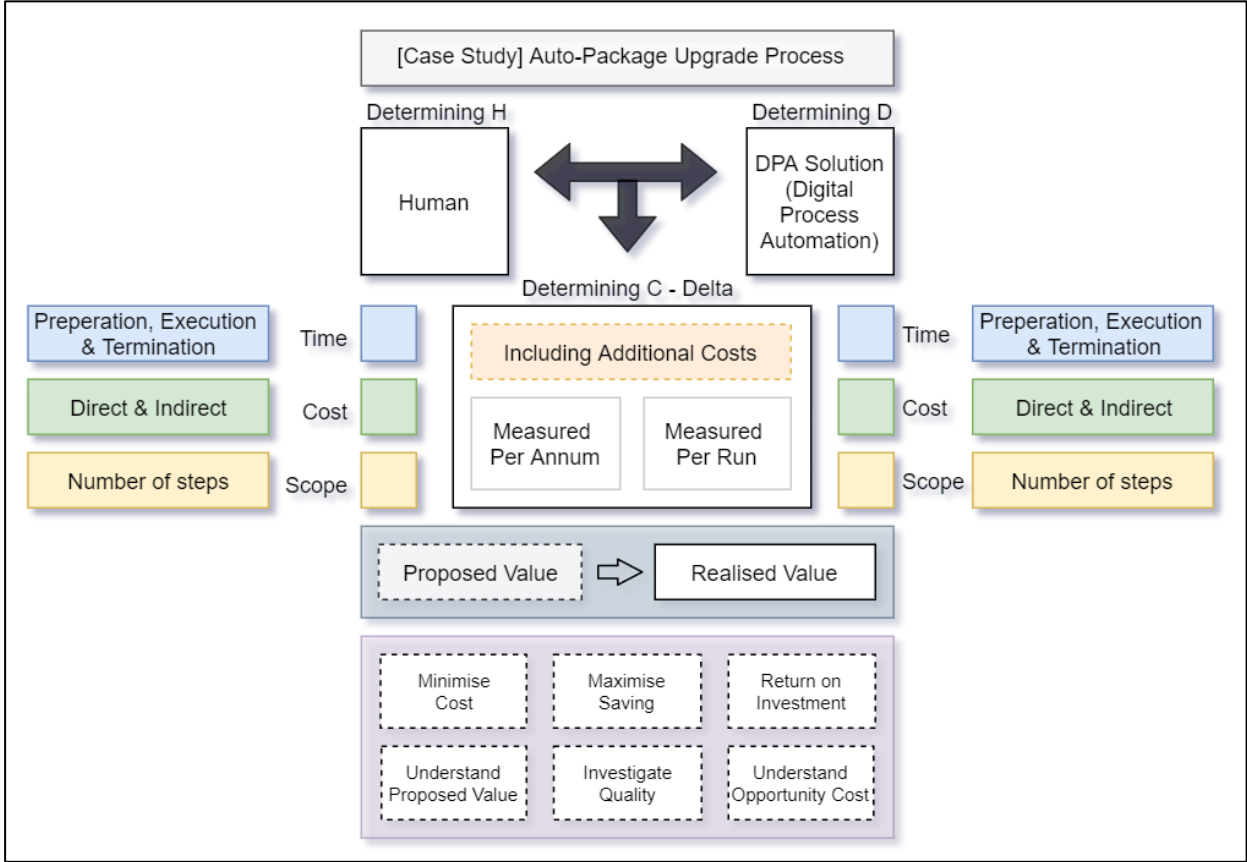


Figure 1-2: Study overview

Figure 1-2 illustrates the context of the study within the case study environment to determine the factors that influence each of the variables (time, cost and scope) measured as a result of the observations of each execution method (human execution and DPA solution execution). The items (white blocks with dotted outlines) presented in the purple block within Figure 1-2 indicate the potential for future work items as a result of this study. As a part of this study, the journey through identifying proposed value and transposing that into realised value is denoted by the grey block presented above the purple block in Figure 1-2.

1.3 Problem statement

A common problem within the automation space is that the wrong processes are being automated, in the wrong ways, without careful consideration of important aspects such as

governance and accountability. One of the contributing reasons is that certain factors of an AS IS (in-place) process are not evaluated appropriately before deciding to automate it.

Asatiani *et al.* (2019) present the following challenges and problems with the deployment of automation in the setting of their research:

- **Governance:** The decision of developing and implementing an automation as well as supporting the infrastructure can be decentralised and committed to within different business units, which makes ensuring governance tedious without defined standards and policies. To mitigate this, standards and governance should be clearly defined and controlled centrally.
- **False premise:** The implementation of automation is often driven by 'quick wins' that promise high returns for low investments. This can be a problem if the returns and investments have not been accurately surmised. This should be mitigated through properly understanding and analysing the factors that determine the measurable, as well as the predicted, returns and investments.

Pagallo (2017) expands on the point of governance by explaining the challenges of regulating accountability. The division of accountability into criminal and civil accountability mostly concerns the open-ended functionality of artificial intelligence (AI) models. Narrowing down the functionality (or scope) to specific, identifiable features reduces the concern that accountability poses.

In a blog posted by SolveXia (2015), understanding the factors that contribute to false premise may be broken down as follows:

- **Employee involvement:** Understanding the number of employees involved in the execution of the process.
- **Complexity:** Understanding how complex the process is in terms of the business rules included in each step.
- **Standardisation and stability:** Understanding how mature (or stable) the process is as well as how much the process has been standardised before attempting automation. Standardisation and stability play a large role in implementing central governance and best practices of automation. By centralising governance, the regulation of accountability also becomes centralised.
- **Volume:** Understanding the process run frequency (how often the process runs and how long it runs for).
- **Outsourcing:** Understanding whether or not the process is currently being executed by an external third party.

UiPath (2016) supports focusing on understanding the governance, accountability, standardisation, complexity and volume of a process before automating the process. UiPath (2016) also proposes focusing on identifying the differences and similarities between the AS IS process and the TO BE process (referring to scope). This is also supported by Willcocks, Hindle and Lacity (2018) who further delve into enablement as one of the three E's.

Evidentially, three common elements may be used to summarise and consolidate the highlighted points, namely: scope, time and cost.

1.3.1 Scope

The common element that can be identified by exploring process governance, accountability, standardisation and stability as well as the similarities and difference between the AS IS and TO BE states of a process, is scope. The scope of a process identifies which governance policies and best practice principles are applicable to the current process, resulting in the ability to specify accountability. The standardisation and stability of a process influence what the process steps should be (in turn, influencing scope). An important aspect of understanding the scope of a process would be understanding the gap between the AS IS and TO BE process as the gap may impact complexity and employee involvement.

1.3.2 Time

Time is an important element to consider when exploring volume, complexity and enablement. The volume (or execution frequency) of a process helps gather an understanding of how many times a process is executed in a given time period (for example, per week or per annum). An increase in process complexity could result in an increase in time required to execute the process. In a case where a process would still require human intervention, the employees would need to be trained and enabled to use the solution correctly. The time it would take for training and enablement would need to be taken into consideration when evaluating the time saving of automating a process.

1.3.3 Cost

Through understanding employee involvement, process complexity, outsourcing as well as the returns and investments that contribute to fake premise, a common influencing element may be identified, namely cost. Outsourcing the execution of a process has a direct cost implication on a business. The increase in complexity of a process results in an increase in the time required to execute a process, in turn, impacting the cost of executing the solution. The number of employees involved in a process impacts how much the process execution costs (which is also dependent on the amount of time it takes for a process to be executed). Investigating the measurable cost of investment and actual cost saving of automating a solution would decrease the risk of the 'quick win' fake premise.

The next challenge arises 18 months later when the automated process is re-evaluated and no longer has value or has significantly less value than when the process was initially implemented (King *et al.*, 2014). This may result in the automation being rebuilt using a different technology that provides more value or even depreciation. An important aspect of re-evaluating whether or not a process should be kept in production would be evaluating the benefit (and saving) of having the process run as a DPA solution instead of having a human execute the process manually. By understanding what criteria to use when evaluating a process, all processes in the same environment may be evaluated according to the same criteria, which may contribute to understanding the holistic benefits of automation within an environment. The problem statement is outlined as follows:

“How can human execution of a process and digital process automation capabilities be evaluated in terms of time, cost and scope?”

1.4 Research questions

The research questions may be split into two categories, namely: the primary questions, which focus on identifying the evaluation criteria of the study, and the secondary questions, which focus on quantitatively measuring the output of the primary questions in the case study environment.

The primary research questions that focus on the evaluation criteria of the study to address the three categories of the problem statement are:

- What parameters influence how long it takes a human to execute the process?
- What parameters influence how long it takes a DPA solution to execute the process?
- What parameters influence how much it costs having a human to execute the process?

- What parameters influence how much it costs having a DPA solution to execute the process?
- What parameters determine the scope of the human's execution of the process?
- What parameters determine the scope of the DPA solution's execution of the process?

The secondary research questions, which focus on quantitatively measuring the output of the primary questions in the case study environment, are as follows:

- How long (on average) does it take a human to execute the process?
- How long (on average) does it take the DPA solution to execute the process?
- Is there an actual realised benefit of time saving by automating the process?
- How much does it cost for a human to execute the process?
- How much does it cost for the DPA solution to execute the process?
- Is there an actual realised benefit of cost saving by automating the process?
- What is the scope of the human's execution of the process?
- What is the scope of the DPA solution's execution of the process?
- Is there an actual realised benefit of scope saving by automating the process?

1.5 Research objectives

The objectives of this study are outlined below.

1.5.1 Primary objective

The primary objective of this study is to identify the parameters that influence the scope, time and cost between human and DPA solution capability in terms of a predefined process.

1.5.2 Secondary objectives

The secondary objectives of this study will be an expansion of the primary objective by evaluating these parameters in a case study.

1.5.3 Theoretical objectives

The focus of the theoretical objectives is to gain understanding of and insight into critical variables identified for this study.

- **To identify the influencing parameters and measure the time it takes a person to execute the process:**
 - The parameters that influence the time it takes a person to execute the process need to be identified before they can be measured.
 - Measuring the time it takes a person to complete the process is the first variable needed to conduct a comparison between human and DPA capabilities. The measurement of this variable is achieved by evaluating the mean of the human execution times measured.
- **To identify the influencing parameters and measure the time it takes the DPA solution to execute the process:**
 - The parameters that influence the time it takes a DPA solution to execute the process need to be identified before they can be measured.
 - Measuring the time it takes the DPA solution to complete the process is the second variable needed to conduct the comparison between human and DPA capabilities. The measurement of this variable is achieved by evaluating the mean of the DPA execution times measured.
- **To identify the influencing parameters and measure the cost of a person execute the process:**
 - The parameters that influence the cost of a person to execute the process need to be identified before they can be measured.
 - Measuring the cost of a person executing the process is the third variable needed to conduct the comparison between human and DPA capabilities. The measurement of this variable is achieved by combining the fixed and variable cost of having a human executing this process.
- **To identify the influencing parameters and measure the cost of the DPA solution executing the process:**
 - The parameters that influence the cost of a DPA solution to executing the process need to be identified before they can be measured.
 - Measuring the cost of a DPA solution executing the process is the fourth variable needed to conduct the comparison between human and DPA capabilities. The measurement of this variable is achieved by combining the fixed and variable cost of having a DPA solution executing this process.

- **To identify the influencing parameters and measure the scope of a person executing the process:**
 - Identifying the scope of a person executing the process is the fifth variable needed to conduct the comparison between human and DPA capabilities.
 - The measurement of this variable poses a slight challenge due to the broad set of capabilities of a human; therefore, the scope will be evaluated based on the steps executed by a human.
- **To identify the influencing parameters and measure the scope of a DPA solution executing the process:**
 - Identifying the scope of a DPA solution executing the process is the sixth variable needed to conduct the comparison between human and DPA capabilities.
 - The measurement of this variable poses a slight challenge due to the broad set of capabilities of a human; therefore, the scope will be evaluated based on the steps executed by the DPA solution.

1.5.4 Empirical objectives

The focus for the empirical objectives is to generate the information that is needed in order to execute the study.

- **To acquire the quantitative data**
 - The quantitative data refers to the cost, time and scope data collected from observing the human execution and DPA solution execution of the process. This is only achievable if the primary objective has been met.
- **To compare the acquired data**
 - Once all data has been gathered, the comparison between scope, time and cost study can be done, resulting in more quantitative data to be analysed. This is only achievable if the primary objective has been met.
- **To identify the significant variables that have an effect on the study**
 - This is only achievable if the primary objective has been met. Data gathered while evaluating the comparison between execution methods should enable the identification of any significant variables that affect the process.
- **To apply business rules to the comparison in order to make informed decisions based on the comparative analysis**
 - This is achievable, provided that the business rules are outlined based on the outcomes of the comparison of the variables.

- **To analyse and communicate the results of the comparison between execution methods**
 - Analysing the comparison between the time, scope and cost of the two execution methods is achievable if all information needed has successfully been gathered. The analysis will enable the communication of the comparison results.

The following section will address the thesis statement of this study.

1.6 Dissertation statement

In combination with the problem statement and research objectives, the dissertation statement for this study is evaluating the time, scope and cost of human labour versus digital process automation capabilities. This encapsulates the identification of the parameters that impact time, cost and scope.

1.7 Delimitation of field of study

The case study takes place in a business unit within an ICT organisation. As such, the scope of this study is narrowed down and governed by the existing standards that have been put in place in the business unit.

The case study in question involves the following:

- A predefined process will be executed by the DPA solution as well as measured against human execution.
- The abovementioned process is a DevOps process that has a specific set of steps, which include going through all of the scripts stored in the central code repository (a set number of scripts will exist in the central code repository to be used by both execution methods. This will ensure that both execution methods are evaluated on the same number of executions), gathering libraries used by each solution with their current version and the latest available version. The latest and current versions are applied and used to identify whether or not an upgrade is available. In a case where an upgrade is available, the libraries are upgraded and the solution is tested to ensure the upgrade did not break anything. In a case where the

upgrade broke something within a solution, the package needs to be reverted to the previous version.

- The process detailed above will be measured in two ways, namely:
 - Through the process execution of a DPA solution.
 - Through the human execution of the process.
- The scope of human execution will be limited by observing three employees executing the process, ten times each. This results in 30 runs in total.

It is important to note that the approach taken to develop the DPA solution, instead of using pre-existing tools, is due to the instruction by the organisation as it takes into account the circumstances and support structures that are currently in place.

Rosenberger and Tick (2018) propose that scope, time and cost integrate and impact one another. This is presented by Raza and Shah (2012) as the triple constraint model. This study will identify and evaluate the parameters that influence scope, time and cost.

1.8 Underlying assumptions

The underlying assumptions of this study are based on the assumptions that the case study scope is clearly defined by the ICT organisation in which the case study will reside, the process is feasible for automation and the process has been documented appropriately. The feasibility and process analyses are supported by the process documentation as it is outlined in Chapter 2, sub-section 2-6.

1.9 Significance of the study

The case study for this study is significant because:

- Automation may be introduced into a workplace in a way that does not intrude on human employment. The focus is not to cut jobs but rather to remove the highly repetitive and non-complex tasks from a person's to-do list so that they can focus on more intuitive tasks.
- Automation can be used to allow time saving that can be used on more productive and challenging tasks.
- The scope of processes can be outlined and logically mapped to preceding and succeeding steps.

- The direct and indirect costs of building and/or executing the automated and manual processes at hand are analysed, broken down and evaluated.
- The execution time of a process is identifiable and potentially optimised.
- Any variables that influence time, cost and scope may be identified and quantitatively compared.

The significance of this study is highlighted as the proposition of the parameters that influence time, cost and scope of implementing automation in an environment to replace human execution processes. The use of a case study will provide evidence of what these parameters are, how they are measured and how they contribute to determining realised savings, if any exist. The work done in this study will provide more insight into the intricacies of time cost and scope that may be expanded in a future study to determine the return-on-investment per automation.

1.10 Chapter layout

Gerber *et al.* (2015) propose a research lens of using DSR as an approach. The purpose of the lens is to identify the motivation for use, the research design followed, the artefact produced and the user experience. The research lens is built on the foundation of the model (as seen in Figure 1-3) proposed by Kuechler and Vaishnavi (2008), which illustrates the different process steps within DSR (along with the outcomes for each step) as well as how each step contributes to the knowledge base.

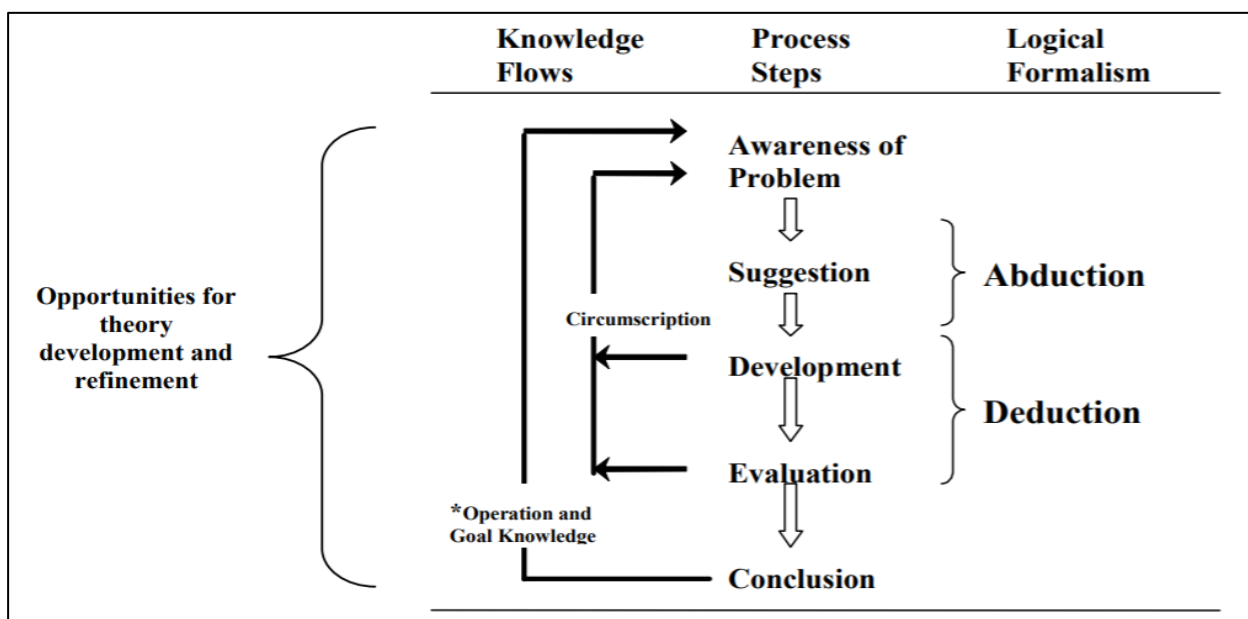


Figure 1-3: Reasoning in the design science research cycle (Kuechler & Vaishnavi, 2008)

The model illustrated in Figure 1-3 has been amended (as seen in Figure 1-4) to represent the contribution of each chapter to the DSR process and the knowledge base, respectively.

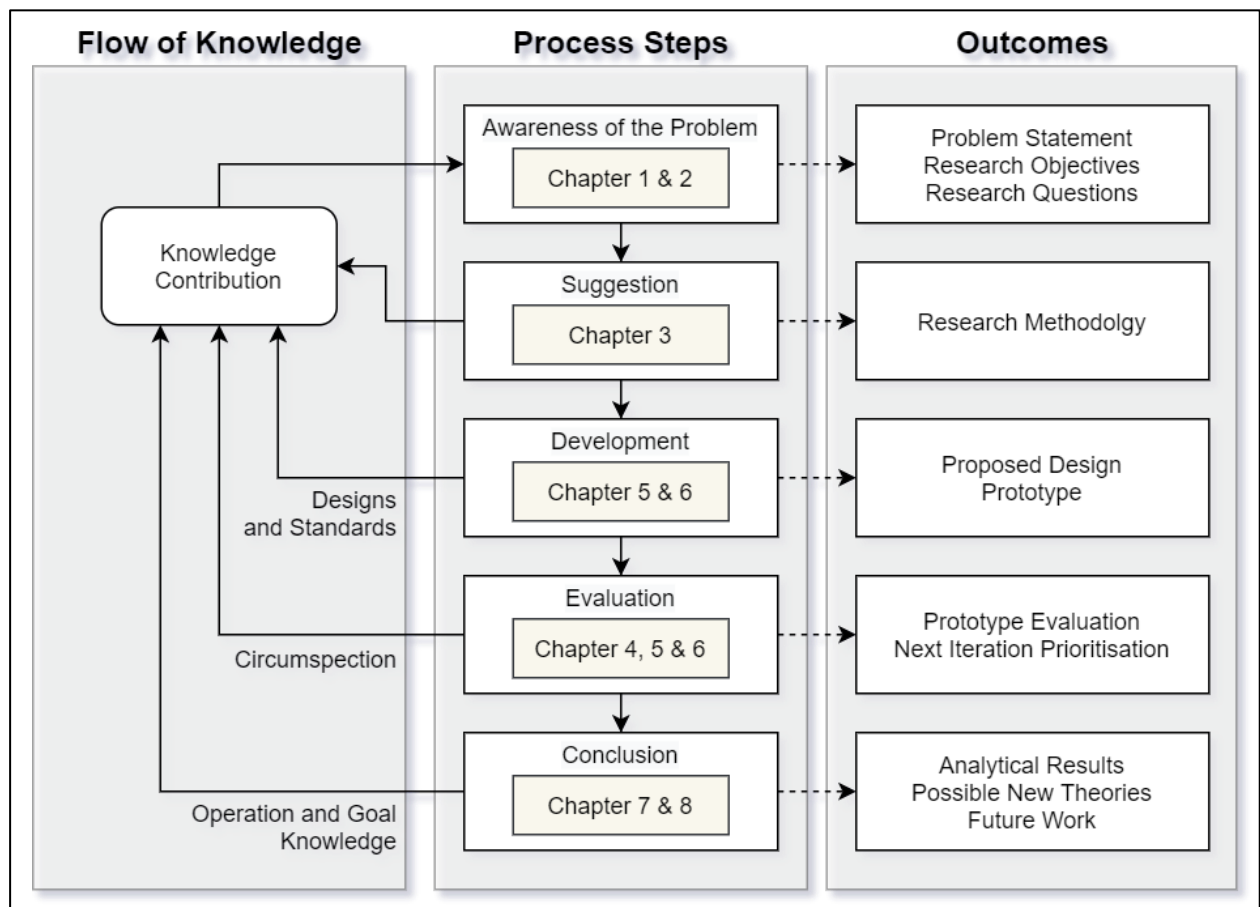


Figure 1-4: Chapter layout in terms of DSR

As seen in Figure 1-4, Chapters 1 and 2 will create the context and awareness of the problem with Chapter 3 suggesting a methodology. Chapter 4 will provide insight into the evaluation of the study. The development and evaluations commence, iteratively, in Chapters 5 and 6. Chapters 7 and 8 will conclude the research, presenting all results and findings. Each chapter is broken down into more detail as follows, in alignment with Figure 1-4:

- **Chapter 2: Literature review**
 - o The literature review will provide more information about concepts central to the study, providing more context and awareness of the problem that this study aims to address.
- **Chapter 3: Research methodology**
 - o The research methodology will be presented through each layer of the research onion, with a focus also placed on the data collection, participants, research instruments, data analysis, study timelines, validity and reliability, limitations of

the study as well as ethical considerations. The case study environment is also introduced in more detail in this chapter. The outcome of Chapter 3 is to provide a holistic view of the research methodology used throughout the study.

- **Chapter 4: Human execution data collection**
 - The data collected from the observation of manual execution of the process will be recorded in this chapter. The chapter details the environmental needs obtained from the environment and contributes to the observation of the environment to the knowledge base, laying the foundation for the scope prioritisation in Chapters 5 and 6.
- **Chapter 5: Initial conceptual design and build**
 - The conceptual design and output from the initial solution development will be recorded in this chapter with the outputs of the chapter being the conceptual design, the prototype and the evaluation of the prototype.
- **Chapter 6: Proposed conceptual design and build**
 - The conceptual design and build of the proposed solution will be recorded along with the data collected from the observation of DPA solution execution of the process. The outputs of the chapter will be the proposed design, the prototype and the evaluation of the prototype.
- **Chapter 7: Data analysis**
 - The analysis done to compare the two execution methods will be conducted and documented in this chapter, resulting in the analytical findings of the research.
- **Chapter 8: Conclusion**
 - The summary of findings, summary of contributions, future research and final conclusions will be provided as the outcomes of this chapter.

1.11 Study timeline

The research process in retrospect will include the conducting of the literature review, adhering to the research methodology, collecting data, analysing the data and producing the conclusion of the study. This will be overseen by consistent iterative review as illustrated in Figure 1-5.

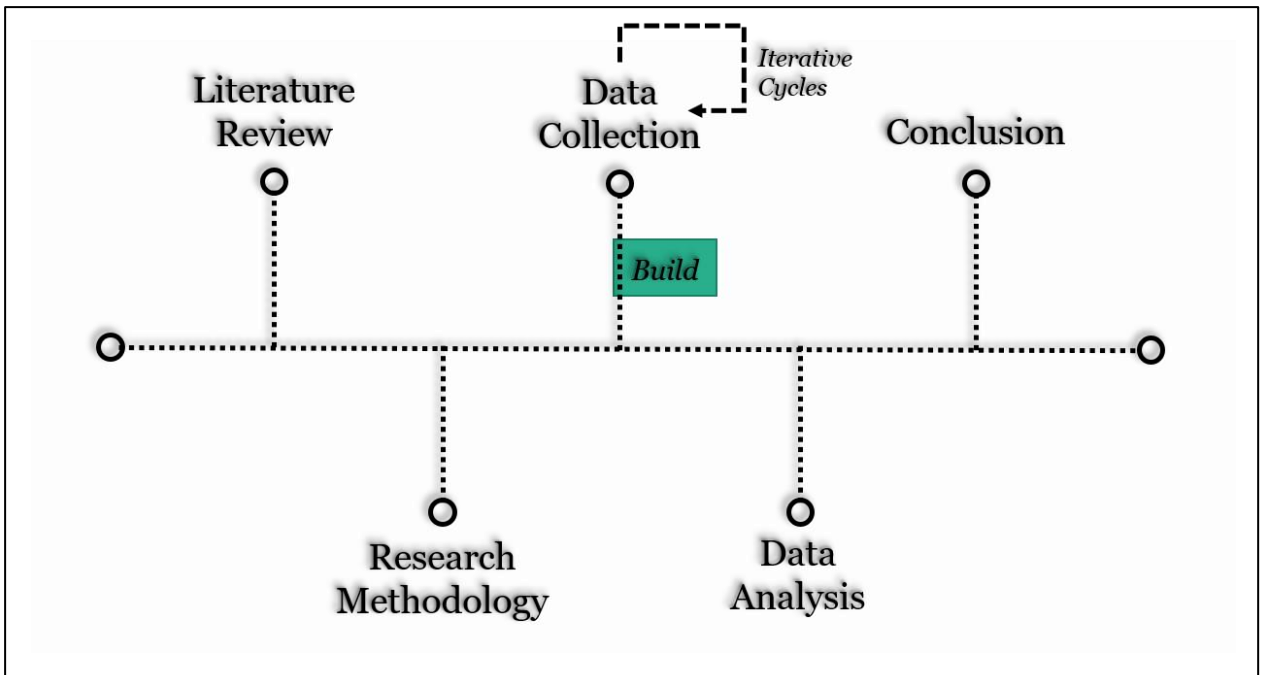


Figure 1-5: High-level study timeline

As mentioned previously, the artefact development forms part of the data collection phase (titled “Build” in Figure 1-5). As such, the build process is broken down in more detail in Figure 1-6.

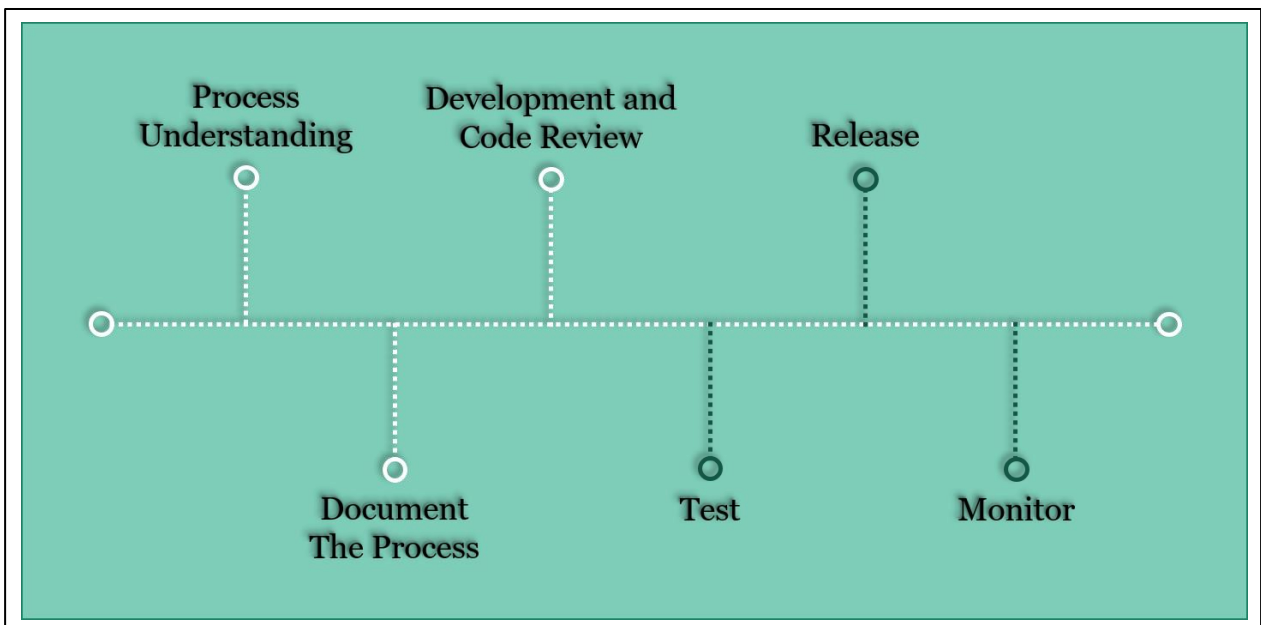


Figure 1-6: Build phase timeline

A deeper look into the building phase, which forms part of the data collection phase, includes the automation lifecycle informally adopted by the team working within the case study environment.

This lifecycle insists that there needs to be process understanding coupled with the correct documentation before development and code review may be done. Once the code review has been conducted, thorough testing is done and monitored before releasing the automation process. Once the process has been released, more monitoring is done to ensure that the process works as expected.

1.12 Conclusion

The aim of this study is to identify the parameters that influence the scope, time and cost between human execution and DPA solution execution in terms of a predefined process, in a case study environment. The study is subject to the research questions, objectives, limitations and underlying assumptions defined in this chapter, in order to create awareness of the problem which extends into Chapter 2.

Chapter 2 focuses on the literature review that highlights and supports elements to be considered throughout the study, with specific focus on the intricacies and components of automation.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter elaborates on the literature that has been found in relevance to this study. The literature studies referenced in this chapter incorporate elements of other studies conducted as well as personal industry experience.

Balic and Ebrahimi (2017) define automation as an entity that acts without external force. Digital process automation (DPA) is built on digital transformation principles, which take a process-centric approach to integrating digital processes and systems into pre-digital (or legacy) workflows as well as outdated operating models (Master of Code, 2020).

This chapter focuses on understanding the intricacies and components of automation. Sub-section 2.2 introduces automation with a focus on the advantages and disadvantages of automation. Sub-sections 2.3 and 2.4 expand on the automation evolution and automation lifecycle, respectively. Sub-section 2.5 focuses on process analysis, while sub-section 2.6 elaborates on automation process documentation. Sub-section 2.7 introduces the triple constraint model, and sub-section 2.8 concludes the chapter. The content of this chapter aims to provide sufficient context surrounding the problem that the study is addressing, in line with the 'awareness of the problem' DSR process step outlined in Figure 1-1.

2.2 Automation

Acemoglu and Restrepo (2019) define automation as the use of technology to enable substituting capital for human labour, which is supported by Smith and Fressoli (2021) who expand on this by proposing automation as a supporting element of the 4IR. Van der Aalst *et al.* (2018) explain that there are different types of automation, one of which they identify as process automation, which contains other subsets of automation types. DPA may be considered as one of the subsets of process automation. DPA is a term used to describe a digital toolset that performs a process in order to complete a workflow or function (Koplowitz & Rymer, 2019). Through the use of DPA solutions, focus can be shifted from manually executing automatable tasks to enriching the lives of others.

2.2.1 Advantages of automation

Automation poses an opportunity to relieve humans from repetitive, hazardous and unpleasant labour (Joshi & Shah, 2015). Future automation technologies also allow for the opportunity to provide a growing social and economic environment in which humans can enjoy a higher standard of living and a better way of life (Groover, 2019).

Groover (2019) proposes that some of the advantages of automation include:

- **Increased productivity** of employees who redirect their attention and focus on tasks that require more human intuition.
- **More efficient use of materials** (like infrastructure and licensing), which can be reused between automations.
- **Better quality** as the process is pre-defined and remains constant, unless there is a change in an underlying system or attribute.
- **Reduced lead times** as the automation has the ability to execute certain tasks faster and more efficiently than people.
- **Higher output**, in cases where an automation executes the process faster than a person, the process can be executed more times than when a person executes the process.
- **Increased process control** as a process should be standardised before it is automated, resulting in more control, stability and understanding of the process.

Master of Code (2020) published a post that largely support the points mentioned by Groover (2019), with an emphasis on improved support and management, improved client satisfaction, faster deployments, scalability and flexibility, cross-industry and cross-platform capability and reduced operation cost. These points are echoed by Joshi and Shah (2015).

An advantage discovered through industry experience is that process automation promotes critical evaluation of a process through which any process flaws can be discovered and addressed. Automation does come with a trade-off and, as such, there are also disadvantages to automation.

2.2.2 Disadvantages of automation

Automation is a very sensitive topic in focus areas where implementation may result in people losing their jobs. Job loss and environmental disruption have many knock-on effects. The high cost of adoption and the loss of jobs are some of the factors hindering the adoption of automation

(Garavaglia, 2009). In addition to this, Groover (2019) proposes the following disadvantages of automation:

- **Worker displacement:** Employees may suffer from stress (the fear of the unknown) and may even be displaced geographically due to a change in responsibility once the automation is implemented.
- **High capital expenditure investment:** Implementing automation can require high capital cost to set up the necessary infrastructure. Automation developers would need to be contracted in order to build the automations, which could also be costly.
- **A higher level of maintenance is needed:** The skills of automation developers would need to be retained when the automation requires enhancement or operational support. In cases where the automation needs to be scaled, these skills would also need to be contracted.
- **A lower degree of flexibility in the process:** Due to the rule-based process steps that automations are built off of, the delegated and 'unintelligent' behaviour of automations that do not have AI built into them, are only as flexible as they are coded to be. The human intuition aspect of a process is removed from the execution of a process when it is automated, resulting in a lower degree of flexibility.

Joshi and Shah (2015) provide an interesting perspective by mentioning that automation leads to draining money from a country, possibly in terms of income tax. In a case where there is mass job loss due to automation, there would also be a reduction in income tax paid to a country's revenue regulator (like the South African Revenue Service (SARS), for example), which directly impacts injections into the economy through reduced government spending (Mitchell, 2005). Garavaglia (2009) re-iterates that the automation adoption rate in South Africa is low, which supports that draining the economy is currently more of a risk than a disadvantage.

The list of advantages and disadvantages has been noticed and adapted since early automation developments, which have evolved over centuries.

2.3 Automation evolution

Groover (2019) highlights a few early and modern examples of automation throughout the history of automation – a few of that are presented in the form of a timeline in Figure 2-1.

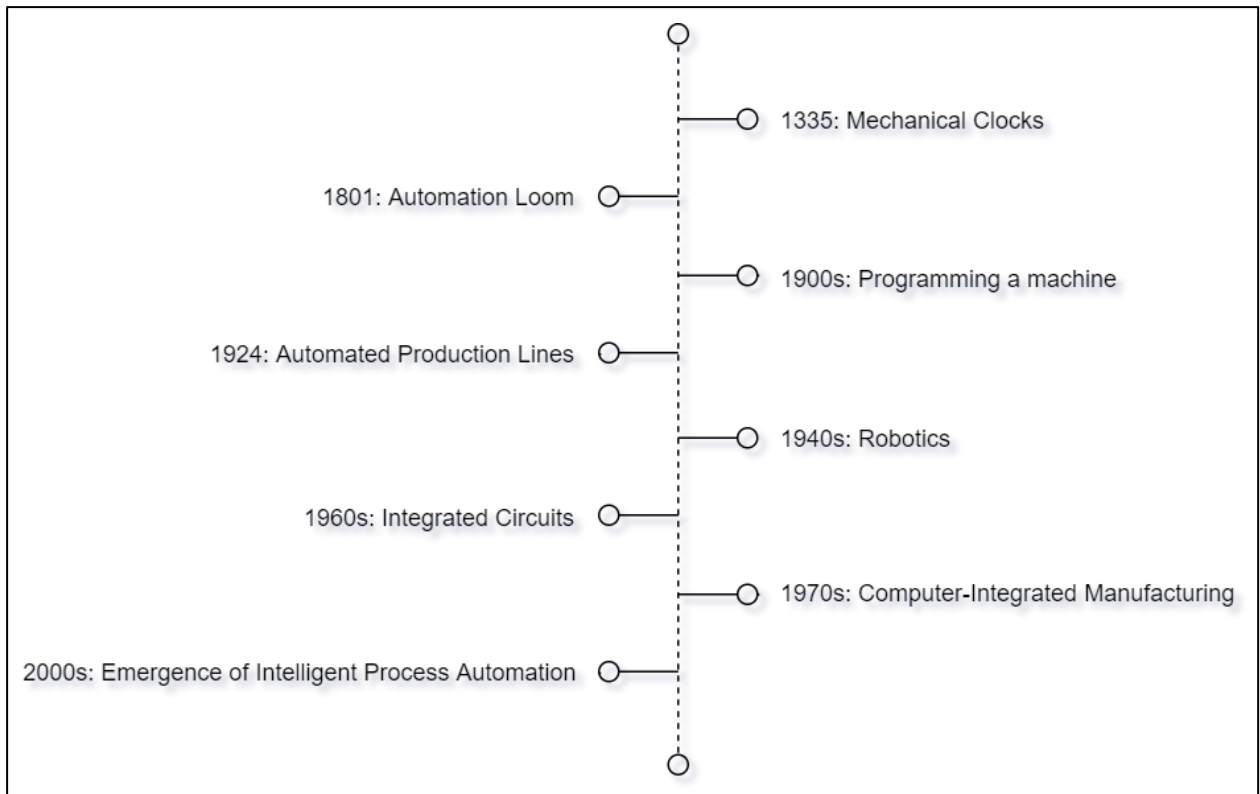


Figure 2-1: The timeline of automation history

The emergence of intelligent process automation has also seen multiple strands of process automation emerge (Van der Aalst *et al.*, 2018). These strands include robotic process automation (RPA) and digital process automation (DPA). Master of Master of Code (2020) refers to both types of RPA (attended and unattended) as systems (or processes) that are designed to mimic human actions, whereas DPA is a type of automated business process management that focuses on efficiently and optimally automating processes. This is supported by Van der Aalst *et al.* (2018) and Koplowitz and Rymer (2019).

Since the expansion on types of automation has commenced, it has become apparent in the industry that some sort of generalised lifecycle would need to be utilised in order to accommodate the implementation of more than one type (or subtype) of automation. Although teams and organisations may have information lifecycles that differ from one another, there are commonalities between them that can be used as a base for generalisation.

2.4 Automation lifecycle

Although there are very few lifecycles that are generalised enough to cater for all types of automation, UiPath (2020c) has released a lifecycle that makes sense for more than just RPA, as seen in Figure 2-2.

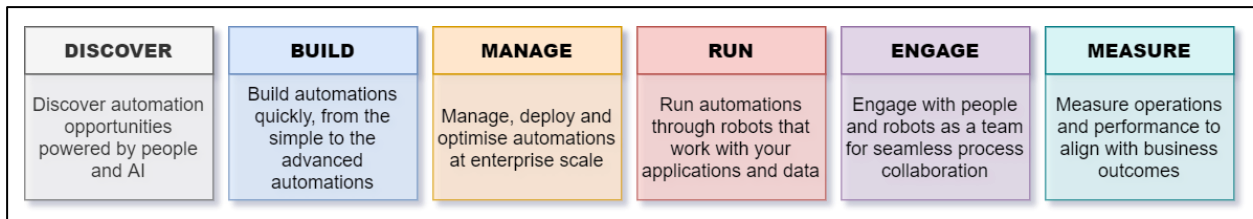


Figure 2-2: UiPath automation lifecycle

Jimenez-Ramirez *et al.* (2019) propose a method to improve the early stages of the automation lifecycle by leveraging the knowledge of the employees executing the process manually. The idea behind the proposal is to monitor the actions of the employee, in a non-invasive manner (using screen recording, for example), which is then transformed into a process model. This is supported by Auth *et al.* (2019). Through their exploration of challenges with an automation lifecycle, Ubis *et al.* (2008) insist that standards are important.

The phases outlined in Figure 2-2 have been amended and are currently in use within the case study environment (see section 3.3 for more detail) for all automation projects that are ongoing. Minor deviations have been tried and tested over the last few years and the current approach mostly aligns to the abovementioned phases. The biggest deviation from the UiPath model is the human element involved in communicating with business during the process analysis phase (which forms part of the *discover* phase represented in Figure 2-2), which is still highly manual. The *discover* phase (as seen in Figure 2-2) is made up of concepts such as process mining and task mining, which make use of tools that investigate log files to better understand the process.

2.5 Process analysis

Leshob *et al.* (2018) define a process as grouped or ordered steps that are executed to achieve a specific goal or outcome. Ziemons *et al.* (2019) present process analysis as the rational breakdown of the production process into different phases (or processes) along with the demonstration of how inputs are transformed into outputs, which is supported by Samost-Williams and Nanji (2020) and a post published by Business Jargons (2017).

System analysis is defined by Whitten *et al.* (2001) as a problem-solving technique that includes the decomposition of a system into components. In a number of studies, like the one conducted by Haruna *et al.* (2017), a process is treated as a subcomponent of a system and, therefore, many concepts that are used on a system (or system analysis) level may also apply to a process. Whitten *et al.* (2001), however, claim that a system *is* a process.

Behzadifard (2020) presents process analysis as the phase in which issues regarding the AS IS process are identified and prioritised according to potential, impact and estimated resolution effort. The Hobbs (2015) post presents the phases of process analysis (which is supported in more detail by Business Jargon (2017) and Oliveira (2015)) as: determining the process to analyse, analysing the process and using the process.

According to Whitten *et al.* (2001), analysis is an important aspect of understanding a process. Rozinat and Günther (2019) suggest that process mining significantly lowers the cost of understanding the current AS IS process.

2.5.1 Feasibility and complexity analysis

According to Vafadar *et al.* (2016), feasibility analysis evaluates the viability of a project. Feasibility analysis is not a single standard; however, it is an alternative to cost-benefit analysis (Masur & Posner, 2010). This is supported by Ramkumar and Sadashivaiah (2018), who suggest that the feasibility analysis should take into account the number of human interventions, the number of decisions to be made in the process and the technical challenges

Ramkumar and Sadashivaiah (2018) suggest that the feasibility analysis should also take the number of users, the number of interfaces, the types of interfaces, the volume of data to be processed, the presence of any complicated business logic as well as the need for standardising or optimising into account. UiPath (2020b) extends on this by proposing that the level of the data structure and the amount of expected change should also be considered.

The feasibility and complexity analysis can be done interchangeably; however, it is important to enforce strict pivots on analysing processes that are not feasible for automation (Ramkumar & Sadashivaiah, 2018). Once the complexity and feasibility of the process have been evaluated, the process mining and discovery phase commences. Process mining and discovery are discussed below.

2.5.2 Process mining and discovery

Van Der Aalst (2011) describes process mining techniques as techniques that aim at discovering, analysing and enhancing process models, which is supported by Behzadifard (2020). There are currently multiple process mining and process discovery tools and techniques that are in use; a few of which, are built as algorithms.

Heuristics Miner (Weijters *et al.*, 2006) is a well-known process discovery algorithm, used by practitioners and researchers. It has been used in benchmarking studies that illustrate its ability to discover high-quality models and processes (De Weerd *et al.*, 2012).

Van den Broucke and De Weerd (2017) present Fodina, which conducts process discovery with a strong focus on robustness and flexibility. Fodina references a Heuristic Miner approach, which works well in scenarios where task and event logs are available. Within the context of this study, there are no event or task logs available, which excludes the more digital approach (like using the Heuristics Miner approach) to understand the process, and instead requires the manual mapping of the process. In a Frost and Sullivan (2019) white paper, the Kryon process discovery approach is outlined to include five steps, namely analyse the process, map the process, recommend the process, visualise the process and automate the process.

The assumption that lays the basis of many process discovery algorithms is the existence of log (or event) files. In a case where human execution of a process is conducted over multiple systems, or even discovering a process that does not yet exist, log files may not be the most reliable basis of process discovery. The alternative to this would be mapping out process steps manually. Once process discovery has concluded, process modelling may commence (Behzadifard, 2020).

2.5.3 Process modelling

Process modelling is defined by Whitten *et al.* (2001) as a technique used to organise and document and structure the flow of data through a system's processes.

Mendling *et al.* (2010) suggest the following guidelines when conducting process modelling:

- Use the least possible number of elements in the model
- Minimise the routing paths of each element in the model
- Make use of one start and one end event
- Make use of verb-object activity labels
- A model that consists of more than 50 elements should be decomposed further

Whitten *et al.* (2001) expand on the guidelines that Mendling *et al.* (2010) suggest by proposing the following steps for conducting process modelling:

1. Construct a context data flow diagram
2. Construct a functional decomposition diagram
3. Construct an event list

4. Update the decomposition diagram
5. For each event, construct an event diagram
6. Combine the event diagrams into one (or more) system diagram
7. For each event diagram, either:
 - a. Describe the elementary process using structured English or
 - b. Explode it into a primitive data flow diagram

The process modelling component of process analysis forms a crucial part of any documentation done in terms of process analysis. The documentation involved is discussed in the subsection below.

2.6 Automation process documentation

An important part of any automation is the process discovery phase in which the discoveries need to be thoroughly documented. The process discovery phase, in terms of RPA, focuses on the business analyst producing a process definition document (PDD). The information present in the PDD is used to determine whether or not a process is feasible for automation. In a case where the process is feasible for automation, the solution architect then produces a solution design document (SDD).

2.6.1 Process definition document (PDD)

A PDD is a document that details the information about a process in its current (AS IS) state as well as in its future (TO BE) state (UiPath, 2020a). The PDD typically introduces the process and the surrounding environment before providing more detail around the AS IS and TO BE process steps, descriptions and input data. Exception and error handling is also an important part of the PDD as it helps the development team understand what business exceptions to expect and cater for.

As the PDD forms part of requirements gathering, the business analyst is responsible for producing such documentation (Shah (2017)). Industry experience suggests that a PDD being compiled by a business analyst who consults with the solution architect on the feasibility of the proposed TO BE process is good practice. In an agile environment, time is always a constraint and the lack of time often impacts the quality of the output or its documentation.

Smith *et al.* (2005) emphasise the importance of mapping out stakeholder roles and responsibility by using RACI. RACI is broken into responsible stakeholders, accountable stakeholders, stakeholders who should be consult when something changes and stakeholders who should be informed when something changes. The stakeholders of a PDD are determined by everyone who forms part of the RACI classification for the specified project. The PDD is signed off by all stakeholders listed in the PDD and is to be obtained by the business analyst (Shah, 2017), once all the necessary information has been documented and typically needs to be signed off before the SDD can be compiled.

2.6.2 Solution design document (SDD)

Ilyin *et al.* (2017) define a solution design document (SDD) as a document that details the information about the TO BE design of a process. Izzo *et al.* (2002) and Ilyin *et al.* (2017) agree in their suggestions that a solution (or software) design document should at least contain an introduction of scope, goals (or aim), intended audience and people involved as well as detailed designs (including module design, database design and user interface design).

The case study environment extends their development standards to the SDD and explain what development standards will be used when developing the solution. These standards extend to naming conventions of the components and variables used within the solution as well as within the database. The case study environment also enforces governance around the secure storage of any information or credentials. The environment to be used to store credentials and sensitive information should also be presented in the SDD to bridge the gap between the solution architect and the developer.

2.7 Triple constraint model

In their evaluation of the workplace and its impact on project management, Raza and Shah (2012) introduce the triple constraint as time, scope and cost. It can be seen in the work presented by Rosenberger and Tick (2018) that the Project Management Book of Knowledge (PMBOK) presents cost management, time management and scope management as three of the project management knowledge areas. This is also evident in a study presented by Khalilzadeh and Alikhani (2020).

The triple constraint model is not without its critics, like Baratta (2006), who proposes that the three factors cannot be predictably defined and related, meaning that cost is not necessarily a

function of time and scope (and *vice versa*). Baratta (2006), however, goes on to investigate whether one constraint can be calculated, given a limit on the other two constraints. For this study, the premise of the triple constraint model is accepted.

As presented by PMBOK, each of the knowledge areas integrates and impacts one another, which means that limiting constraints may not necessarily determine a third constraint, but it would have an impact that may be assessed. The assessment of these constraints has been expanded to include quality and is known as the project diamond, as noted by Bekemeier *et al.* (2020).

2.8 Conclusion

With automation and the triple constraint model central to this study, concepts like the automation evolution and lifecycle introduce the history and process of automation implementation, respectively. By understanding the process analysis and types of process documentation (along with their uses), the value of analysing a process properly becomes evident. Many shortfalls of implementing automation can be avoided when process analysis is done thoroughly, bedding down the scope in the appropriate phase of the lifecycle. By clearly defining the scope, the variance of time and cost become more measurable and manageable. The context surrounding how scope, time and cost fit into the study forms the necessary awareness of the problem, which enables the next step in the DSR model, presented in Figure 2-3, the suggested research methodology.

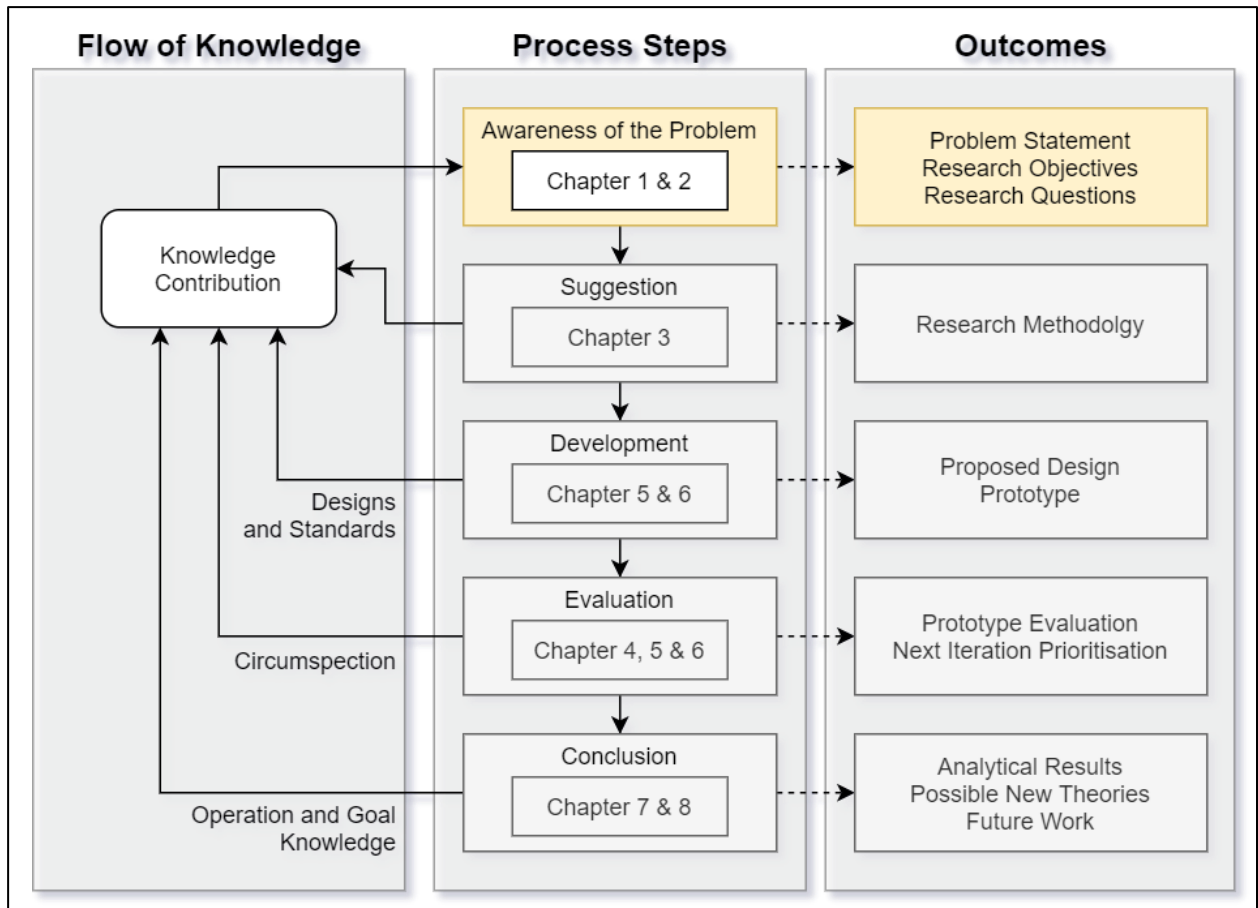


Figure 2-3: Chapter 2 DSR study layout contribution

Chapter 3 introduces the research methodology to be used throughout the study. The case study environment will also be introduced in greater detail so that it provides a clear foundation for the participant selection, data collection and data analysis approaches identified.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

Mohajan (2018) defines a research methodology as the logical development of the process used to generate the theory that is used as the procedural framework within which the research is conducted. Snyder (2019) extends on this through the proposition that a good research method lays the foundation for facilitating the development of knowledge.

A pragmatic approach will be used to conduct the research presented through this study with the encapsulation of design science research (DSR). The data collection portion of the study relies heavily on the development of an artefact for which DSR will be used. Since the development of the artefact is needed for data to be collected, the artefact is considered to be a research instrument, rather than the focus of the study – which is why DSR is not the main research methodology of the study.

Sub-section 3.2 of this chapter will focus on detailing the methodology to be used throughout this study with specific reference to the ontology, epistemology and axiology of the study. Sub-section 3.3 introduces case study research with more detail about the case study environment in which the research will be conducted. Sub-section 3.4 provides insight into the study timeline, while sub-sections 3.5 and 3.6 discuss data collection and participants of the study, respectively. Sub-section 3.7 explains how data analysis will be conducted. Sub-section 3.8 focuses on the validity and reliability of the study; sub-sections 3.9 and 3.10 discuss the limitations of the study and the ethical considerations, respectively, with sub-section 3.11 concluding the chapter.

3.2 Methodology

According to Saunders *et al.* (2012), a research philosophy organises the development of knowledge in a particular field of study. The research onion presented by Saunders *et al.* (2012) proposes that there are mainly four different research philosophy approaches, namely positivism, realism, interpretivism and pragmatism, which have been amended in Figure 3-1 to represent the position of this study. The sections that follow further elaborate on the research onion elements as well as how they are to be applied to this study.

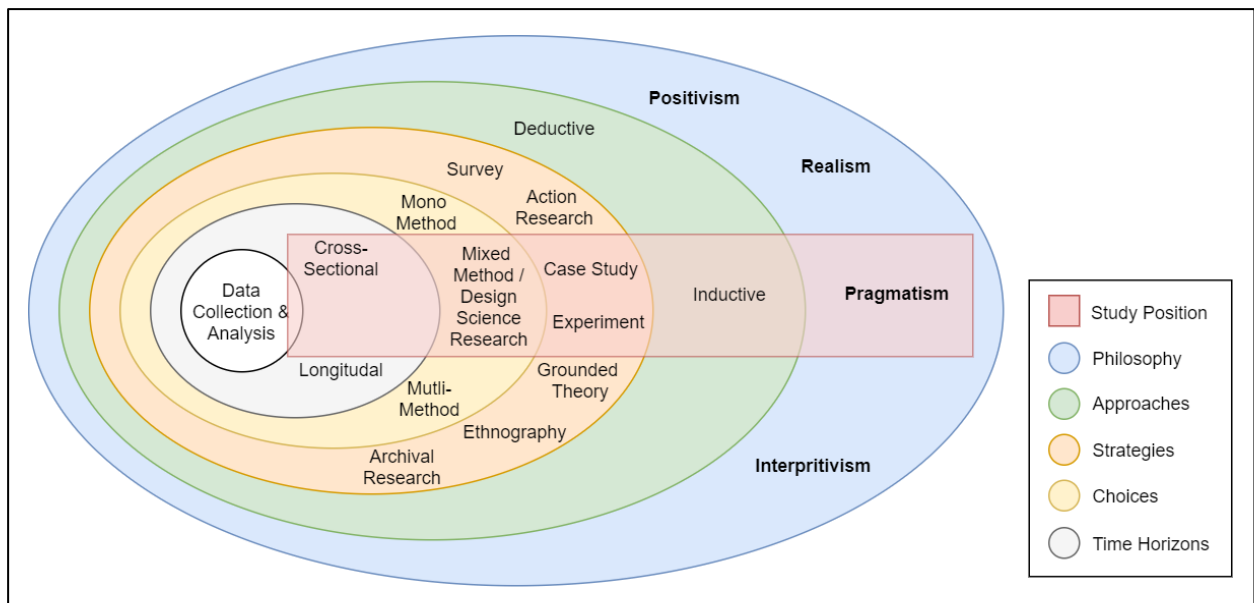


Figure 3-1: The research onion, adapted from Saunders *et al.* (2012)

3.2.1 Philosophy: Pragmatism

The outer-most layer of the research onion presented in Figure 3-1 represents the research philosophies that a study could adopt. The philosophy that best fits this study is pragmatism. The pragmatism philosophy revolves around adopting a somewhat unrealistic approach in practice and avoids the debates on reality or truth with the focus rather placed on what the researcher perceives as valuable (Gunesh, 2016). The importance of this approach is the practical consequence of the study. Gunesh (2016) suggests that pragmatists recognise that there are many different ways of interpreting the world and undertaking research. There might not be only a single point of view but rather the perception is that there may be multiple realities with different outcomes (Saunders *et al.*, 2012). This is reiterated by Creswell and Clark (2017). Tashakkori *et al.* (1998) state that “as a research paradigm, pragmatism is based on the proposition that researchers should use the philosophical and/or methodological approach that works best for the particular research problem that is being investigated”.

3.2.1.1 Ontology

Saunders *et al.* (2012) as well as Gunesh (2016) propose that ontology refers to the researcher's view of the nature of reality. This is supported by Kaushik and Walsh (2019), who propose that the view of the researcher is that the research at hand may either be conducted as objective or

subjective research. Objective research is conducted based on facts and the premise that objective assertions are proven to be true or false, resulting in facts or statements. Subjective research is conducted on the deviation from facts with a more opinion-based guide. This dissertation is to be written from the perspective of being as objective as possible since the variables that are to be determined through a list of steps and rules (assertions) are not subject to belief-based elements.

3.2.1.2 Epistemology

Saunders *et al.* (2012) as well as Gunesh (2016) propose that epistemology is the researcher's view regarding what constitutes acceptable knowledge, while Kaushik and Walsh (2019) suggest that epistemology refers to the assumptions about how the world is known. The assumptions surrounding this research include the following:

- Due to the impact that the research topic may have on lives, it is important to understand as many aspects of the research as possible.
- Better understanding of impact allows researchers to find and propose new ways of enriching lives and curbing the effects of inevitable changes.

This research will be conducted from the background perspective of the above understanding along with measurable elements that assist in gaining further knowledge about the impact of automation.

3.2.1.3 Axiology

Kaushik and Walsh (2019) suggest that axiology refers to the beliefs about the role of values and morals in research. This is supported by Saunders *et al.* (2012) as well as Gunesh (2016). Since this research is predominantly focused on automation, there are certain levels of sensitivity that need to be taken into consideration around the impact (both positive and negative) that automation could have on people's lives. With that in mind, the focus of this study is to maximise as much positive impact as possible and minimise the negative impact as far as possible.

3.2.2 Approach: Inductive

The green (or second-most-outer) layer of the research onion presented in Figure 3-1 represents the approaches that a study could adopt. Research can either follow an inductive or deductive approach (Saunders *et al.*, 2012). The determination of a study being inductive versus deductive is based on the direction of flow between generalised principles and a specific conclusion. Inductive reasoning moves from specific instances towards a generalised conclusion (Thomas, 2003). Deductive reasoning moves from generalised principles to a specific conclusion (Perdicoúlis & Glasson, 2006). Since this dissertation is placed within a case study environment with specific standards, objectives and a brief from which a theory should be developed, this research will follow an inductive approach as the reasoning will converge from a specific instance to a generalised theory.

3.3 Strategy: Experiments within case study research

A research strategy is deemed to be an overall plan for conducting a research study. It guides a researcher with planning, executing and monitoring a study (Johannesson & Perjons, 2014). As seen in Figure 3-1, this study is positioned as experiments within a case study, which design science research has close ties to, within the research strategies ring of the research onion.

Croson *et al.* (2007) propose experiments as the collection of primary data through observing individual decision-making or interactive tasks, which is supported by the way that Hibbard *et al.* (2007) take the changes within an environment into account as they conduct experiments to identify similarities. Sub-section 3-5 provides more detail on the experiments to be conducted within this study.

According to Harrison *et al.* (2017), case study research is used to study a range of purposes and topics that contain the necessary variables and problems that the study is focused on understanding or solving.

Harrison *et al.* (2017) propose the fundamental elements of case study research as follows:

- **The case:** The objective of the case study is identified as an entity of interest.
- **A bounded system:** The binding of relevant connections by time, space and activity to manage contextual variables.
- **Studied in context:** In order to understand the case, the natural environment should be observed to obtain the necessary contextual variables.

- **In-depth study:** Intensive analysis of the identified issue(s) should be conducted as an intrinsic enquiry into the environment.
- **Selecting the case:** The purpose and condition of the study should guide the selection of the case and the decisions that should be made within the environment.

The elements above include approaches suggested by Stake (2006) as well as Merriam and Tisdell (2015). These elements are further elaborated on in sub-section 3.3.1.

3.3.1 Case study environment

Since the case study project is based in a specific business unit within an ICT organisation context, it is subject to the specific business unit's environment, governance and standards. The organisation adopted the scaled agile framework (SAFe) as a project management methodology in 2018. Since then, the way that projects have been managed has transitioned slowly from waterfall-like methods to a more agile approach.

3.3.1.1 Project management methodology: SAFe

Ageling (2019) presents a high-level introduction of SAFe, as positioned by Leffingwell (2018), as a template to scale agile to suit the environment of large organisations. It has two to four levels:

- **Team level:** The team level consists of agile teams that work together (using Scrum or even Kanban) to deliver progress on projects every two weeks. The team backlog remains the responsibility of the product owner (PO). Iteration planning, stand-up and retrospective are similar to the standard Scrum ceremonies. Scrum is an agile framework that provides a guiding methodology to designing, developing and delivering solutions (Gonçalves, 2018). Iteration planning refers to the ceremony in which the workload is planned for a two-week period, also referred to as a sprint (Henreaux *et al.*, 2021). Stand-ups are used to determine what work has been done since the last stand-up, what work will be done before the next stand-up, and what blockers exist (Stray *et al.*, 2018). A retrospective happens at the end of an iteration and aims to identify what went well, what did not go well and what could be improved (Gaikwad *et al.*, 2019).
- **Programme level:** The programme level is proposed as the level with exactly the same elements as team level, except that the focus shifts to scaled practices (scrum of scrums) where teams work in parallel (and in some instances with one another) to deliver value

incrementally. That, in turn, means that multiple teams work to deliver something together. This setting can be scaled to up to 150 people. The teams work to create a programme increment (PI), which needs to be delivered after five sprints or iterations (ten weeks).

- Large solution level (optional):** The large solution level manages large solutions that cannot be handled by a single programme or agile release train (ART). The solution train coordinates the larger solutions. The solution train consists of a solution manager, solution architect and a solution train engineer. The solution train aligns and coordinates multiple ARTs. ARTs encapsulate multiple teams working together to deliver value and solutions incrementally.
- Portfolio level (optional):** Portfolio management directs the trains through budget allocations. This level includes a backlog of portfolio-level epics, carried down to the train's product managers. Epics are large pieces of work that are broken down into features (which are further broken down into user stories). Overhead is limited on this level through the use of Kanban (Ahmad *et al.*, 2018).

The scaled agile framework is explained by means of Figure 3-2.

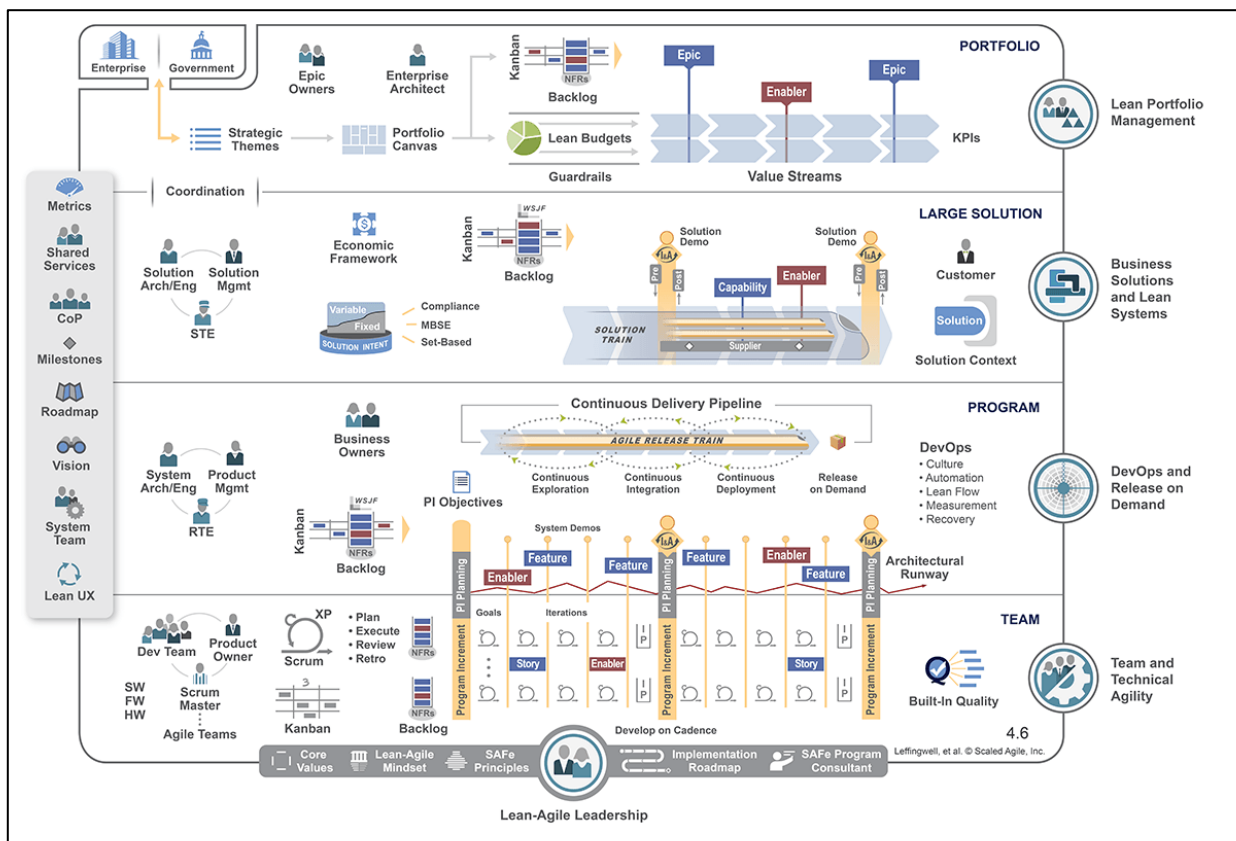


Figure 3-2: Full scaled agile framework (Leffingwell, 2018)

3.3.1.2 Four flavours of SAFe

As presented by Leffingwell (2018) and Ageling (2019), SAFe has four different flavours, namely:

- **Essential SAFe:** consists of team level and programme level
- **Large solution SAFe:** consists of team level, programme level and large solution level
- **Portfolio SAFe:** consists of team level, programme level and portfolio level
- **Full SAFe:** consists of all four levels.

The case study environment consolidates all four levels of SAFe, making it a full SAFe environment. Furthermore, the automation team within case study environment has amended the UiPath automation lifecycle to cater for the end-to-end model that guides the delivery of all automation projects. This model is presented in Figure 3-3.

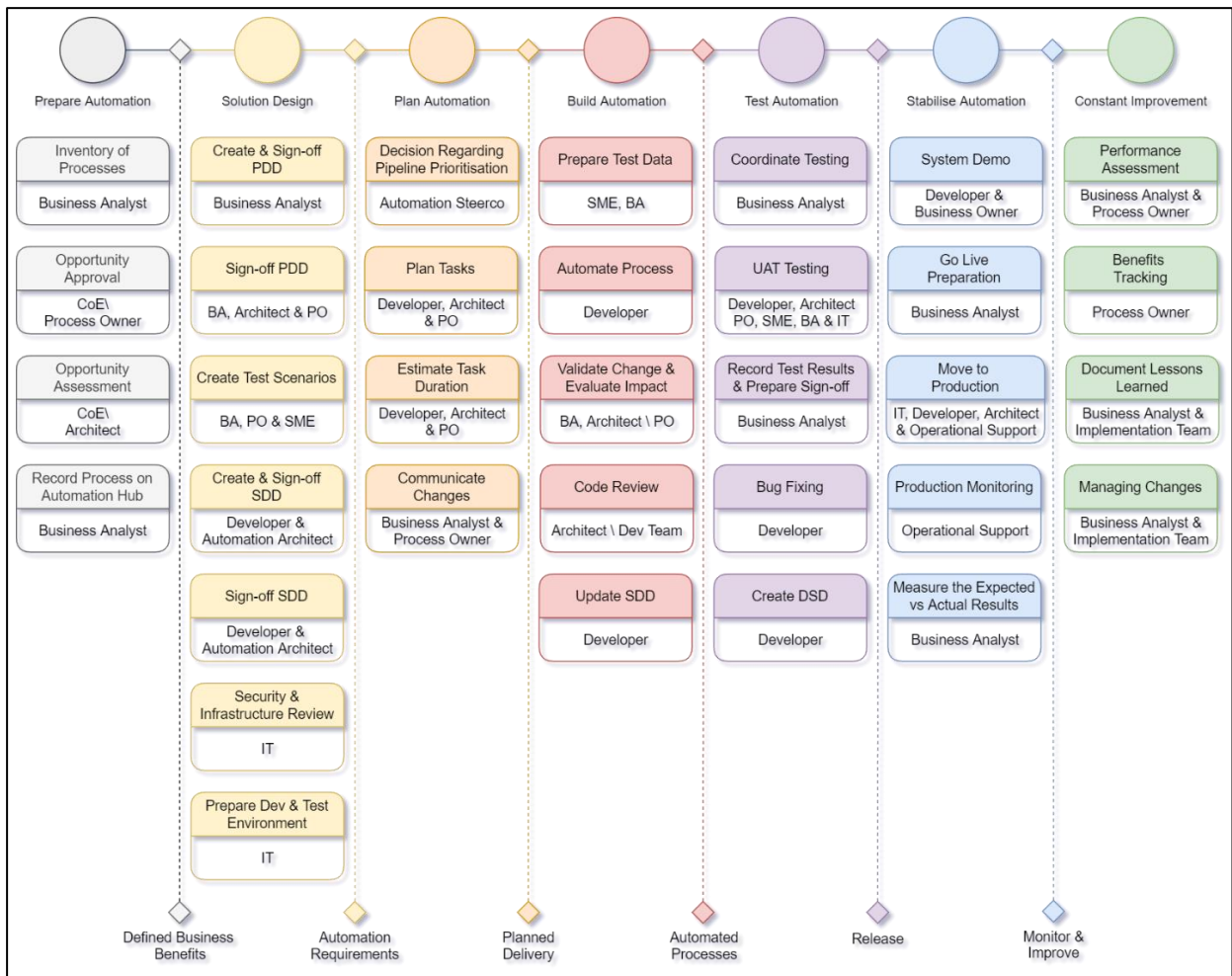


Figure 3-3: The end-to-end automation delivery model

There are multiple stakeholders involved in the end-to-end automation lifecycle. Each of these stakeholders performs different tasks and has different responsibilities. These tasks and responsibilities are outlined per stakeholder as follows:

- **Centre of excellence (CoE):** The CoE is responsible and accountable for controlling the governance, standards and policies of the automations that are developed. In the case study environment, the development team forms part of the automation CoE
- **Business owner (BO):** The project sponsor from a business perspective who determines the business value of a solution.
- **Process owner (PO):** A member of the greater team who defines the user stories with acceptance criteria and provides the prioritisation of stories. The product owner is responsible for maintaining the conceptual and technical integrity of features.
- **Scrum master:** A member of the greater team who leads the team through a project using agile project management techniques. The scrum master is responsible for managing the agile process.
- **Business analyst (BA):** The BA is responsible for bridging the gap between the project team and the developers, in terms of the requirements and expected outputs of the solution.
- **Architect:** The architect can be a solution architect or an automation architect or both. The terms are used interchangeably to describe the person responsible for compiling the technical design of the solution.
- **Subject matter expert (SME):** The person within business who has experience in aspects or components of a solution. In automation, the SME would be the person (or people) who executes the process manually.
- **Developer:** The developer or developers collectively make up the project team, responsible for developing and delivering the working features of the solution, according to the user stories and acceptance criteria that is committed to.
- **IT:** The team concerned with setting up the infrastructure needed for the solution to be executed.
- **Operational support:** The team that is responsible for maintaining and supporting the solution once it is in production.
- **Implementation team:** The team of developers, operational support, the architect and IT who are responsible for deploying the solution in the production environment and conducting the necessary hand-overs of the solution knowledge.
- **Automation steering committee (steerco):** The automation steerco is made up of all business owners who are responsible for projects that are included in the PI. The objective

of a steering committee is to ensure that there is consensus of project priority based on the business value each project receives.

- **Project team:** The team of developers who develop the functionality of the solution according to the user stories and acceptance criteria that were committed to.

It can be seen in Figure 3-3 that the business unit iterates through the following high-level phases:

- **Prepare automation:** The preparation of the automation phase includes the pre-work and process information gathering needed to evaluate whether or not the automation would be feasible as well as how complex it would be. This information is recorded in the organisation's instance of the UiPath automation hub, which caters for any automation solutions. The UiPath automation hub is a platform that allows users to capture the details of automation opportunity along with information pertaining to the potential cost saving, associated documentation and stakeholders. The UiPath automation hub provides the cost-benefit and feasibility analysis of automating the opportunity that has been identified.
- **Solution design:** Once the automation has been assessed and approved on the approval hub by the CoE, the documentation is prepared by the BA and the solution architect. Once the PDD and SDD have been signed off, the automation project is put forward for PI planning.
- **Plan automation:** During PI planning, the tasks that are needed to deliver the projects that have been put forward are broken down and workload is estimated. The tasks (along with their estimations) are placed on team's board, along with velocity (taking into account leave and team members off days, like public holidays or training) and load per sprint in the order that they should be completed in. This leads to prioritisation of the projects based on certain factors, one of which is business value determined and assigned by the automation steering committee.
- **Build automation:** The building of the automation project happens during the sprints, which is followed by a code review and the amendment of the documentation, if necessary. Code reviews are performed iteratively when new features are added or when issues that were raised in previous code reviews are complete.
- **Test automation:** The testing phase includes the preparation of the environment as well as the actual testing and bug fixes, which are also recorded on the Azure DevOps board for visibility. Azure DevOps is a Microsoft platform that allows users to create and manage user stories, tasks, code repositories, pipelines (as well as releases) and test plans (among other features). Azure DevOps is used to track project progress and burn down. Once all tests have passed, the solution may be demonstrated at the system demo.

- **Stabilise automation:** Once the system demo has been conducted, the feedback is noted and turned into action points, where necessary. The system demo is followed by the deployment of the solution to production as well as the measurement of actual versus expected results and benefits.
- **Constant improvement:** The constant improvement phase includes the constant evaluation of performance and measurement of benefit. In a case where enhancements are requested, the process is reiterated as the 'prepare automation' phase is kicked off again.

The above model and phases will be adhered to throughout the development of the artefact to be used for the data gathering action of this study.

3.3.1.3 Environment composition

There are six people working within the specific case study environment, of whom three employees can successfully execute the process chosen for automation as part of their role. The environment consists of employees on different skill levels, with different specialties and years of experience. The team working within the case study environment is made up of managers, architects, IT operation support engineers and developers.

3.4 Choice: Design science research (DSR)

Iyawa *et al.* (2016) define DSR as a methodology that focuses on creating new knowledge with the purpose of changing existing situations into preferred situations. Simon (2019) (supported by Hevner (2007)) proposes that DSR introduces new and innovative artefacts that motivate the desire to improve the environment. This is supported by Lukka (2003), who suggests that DSR is used to solve real-life problems by making a contribution to the applied theory presented throughout the conducted research.

Hevner (2007) strongly supports the use of DSR to introduce new knowledge through the invention and innovation of theories and artefacts, executed in three cycles, namely relevance, design and rigour. The development of theories and artefacts will take in all environmental needs and requirements as inputs, utilising any applicable knowledge from the available knowledge base. The iterative design cycles will be evaluated in each iteration, while outputting the environmental application and knowledge base additions to the environment and knowledge base, respectively. The relevance cycle will represent the flow of all environmental needs and

environmental application between the environment and the information system (IS) research. The rigour cycle will represent the flow of knowledge between the knowledge base and the IS research.

Gerber *et al.* (2015) propose a research lens of using DSR as an approach. The purpose of the lens is to identify the motivation for use, the research design followed, the artefact produced and the user experience. The research lens is built on the foundation of the model proposed by Kuechler and Vaishnavi (2008), which illustrates the different process steps within DSR (along with the outcomes for each step) as well as how each step contributes to the knowledge base.

The model presented in Figure 1-4 illustrates how each chapter will contribute to the knowledge base and implementation of DSR throughout the study. Chapters 1 and 2 will create the context and awareness of the problem with Chapter 3 suggesting a methodology. Chapter 4 will provide insight into the evaluation of the study. The development and evaluations commence, iteratively, in Chapters 5 and 6. Chapters 7 and 8 will conclude the research, presenting all results and findings.

The artefact should be developed and evaluated iteratively, which would require a well-defined framework that works well with iterative cycles. The data collection portion of this study is dependent on the development of an artefact so that the appropriate variables may be measured and used for comparison

3.4.1 The open group architecture framework (TOGAF)

Due to the iterative nature of this research, there is a need for a framework, like the open group architecture framework (TOGAF), which provides an approach to application architectural design, planning, implementation and governance (Josey, 2016). Josey (2016) proposes TOGAF as an enterprise application framework that is based on an iterative process model that assists in the acceptance, production, use and maintenance of architectures, according to best practices and re-usable architectural assets. Blevins *et al.* (2004) suggest TOGAF as an iterative cycle of phases within the framework. The iterative cycles outlined in Chapters 4 and 5 will be focused on in the business layer, the information systems layer and the technology layer. The business layer will be used to highlight the organisational structure and architecture of the case study environment with emphasis placed on the operating model. The information systems layer will focus on outlining the data architecture and the application architecture. The technology layer will expand upon the existing technology architecture that will form the base of the proposed automation solution. These phases are outlined Figure 3-4.

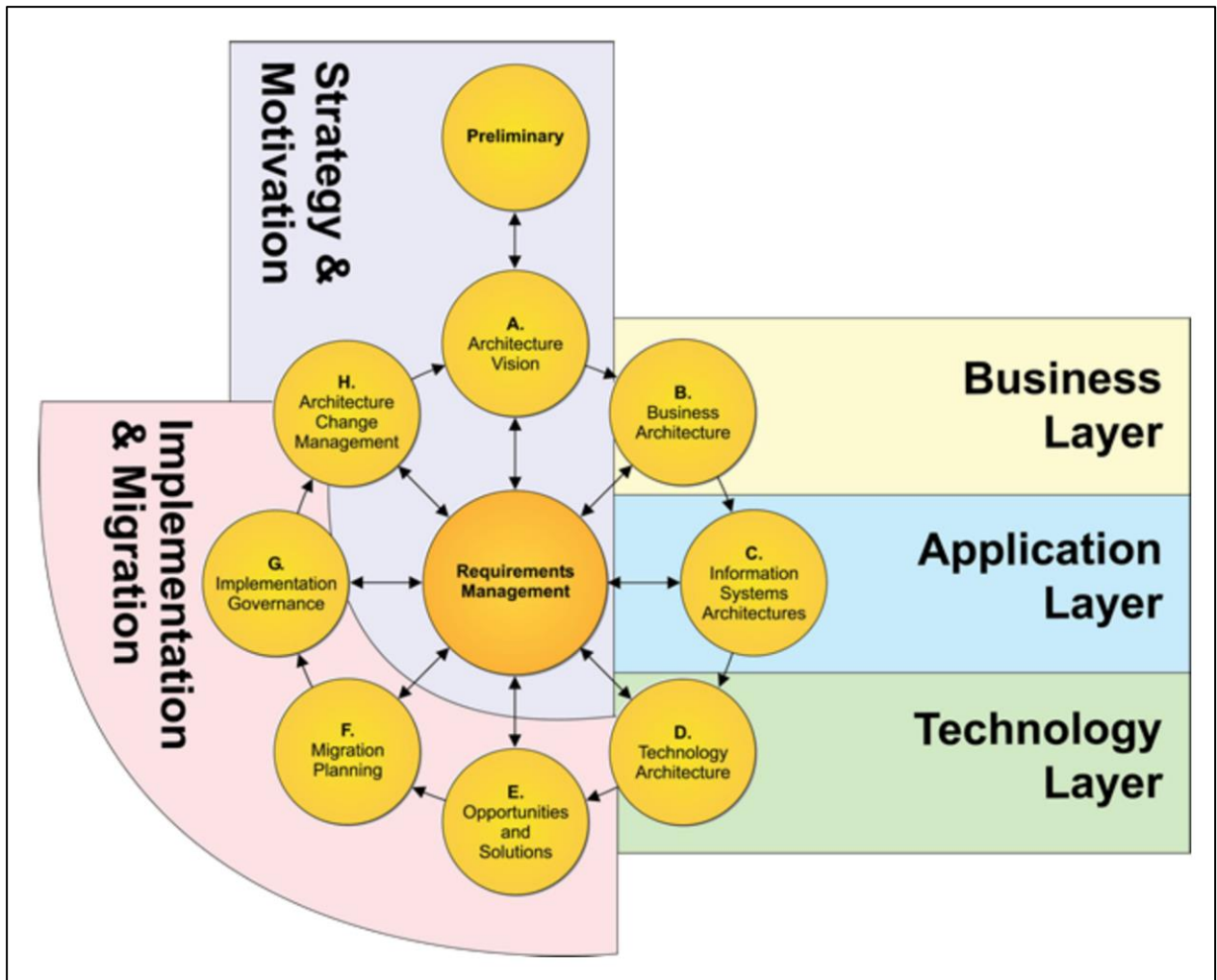


Figure 3-4: The open group architecture framework (TOGAF) (Josey, 2016)

Since the case study environment makes use of SAFe, the development of the artefact will be executed in accordance with SAFe (as explained in section 3.3.1.1) and TOGAF.

3.5 Data collection

Boterman (2021) proposes that quantitative research is concerned with measuring the quantification of a study, meaning that observations and experience should be measured in quantities. Skinner (2020) expands on this by stating that the objective of quantitative research may be determined prior to data collection, but the relationship between variables may (in some cases) only present themselves during analysis. The relationship between variables may be open-ended or exploratory (Skinner, 2020).

In their study, Anguera *et al.* (2018) use observation as a quantitative data collection tool to measure qualitative elements. Tootian (2018) justifies the use of exploratory data collection, as

part of quantitative data collection, as it enables the elicitation of understanding the impact that certain approaches have on a study.

The data that needs to be collected for this dissertation can be obtained by conducting two exploratory experiments, through observation, which is a quantitative data collection method. The data needed in this study can be split into six variables, namely:

- 1. Cost of human execution of the process
- 2. Time of human execution of the process
- 3. Scope of human execution of the process
- 4. Cost of DPA solution execution of the process
- 5. Time of DPA solution execution of the process
- 6. Scope of DPA solution execution of the process

The above can be split into two parts, namely collecting data from human execution of the process and collecting data from the DPA solution executing the process, as seen in Figure 3-5.

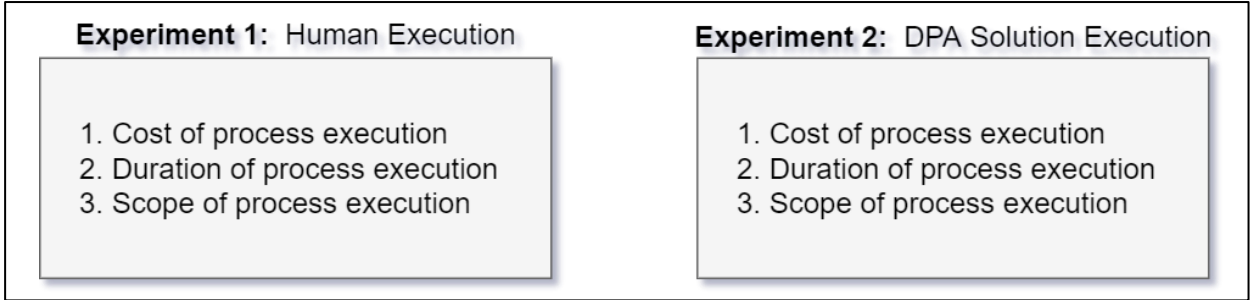


Figure 3-5: Experiments within this study

Both methods of execution (as identified in Figure 3-5) will be measured using the same process, which is illustrated in Figure 3-6. The process in question was one of many that were subjected to a selection and prioritisation process which took the feasibility analysis and complexity analysis (further discussed in Chapter 2) into account before business value was assigned to each automation opportunity. The automation opportunity with the highest business value assigned to it was selected as the process to be automated as part of this study. The selection of another process may render different results however, the contributing factors could still be generalised across processes/automation opportunities.

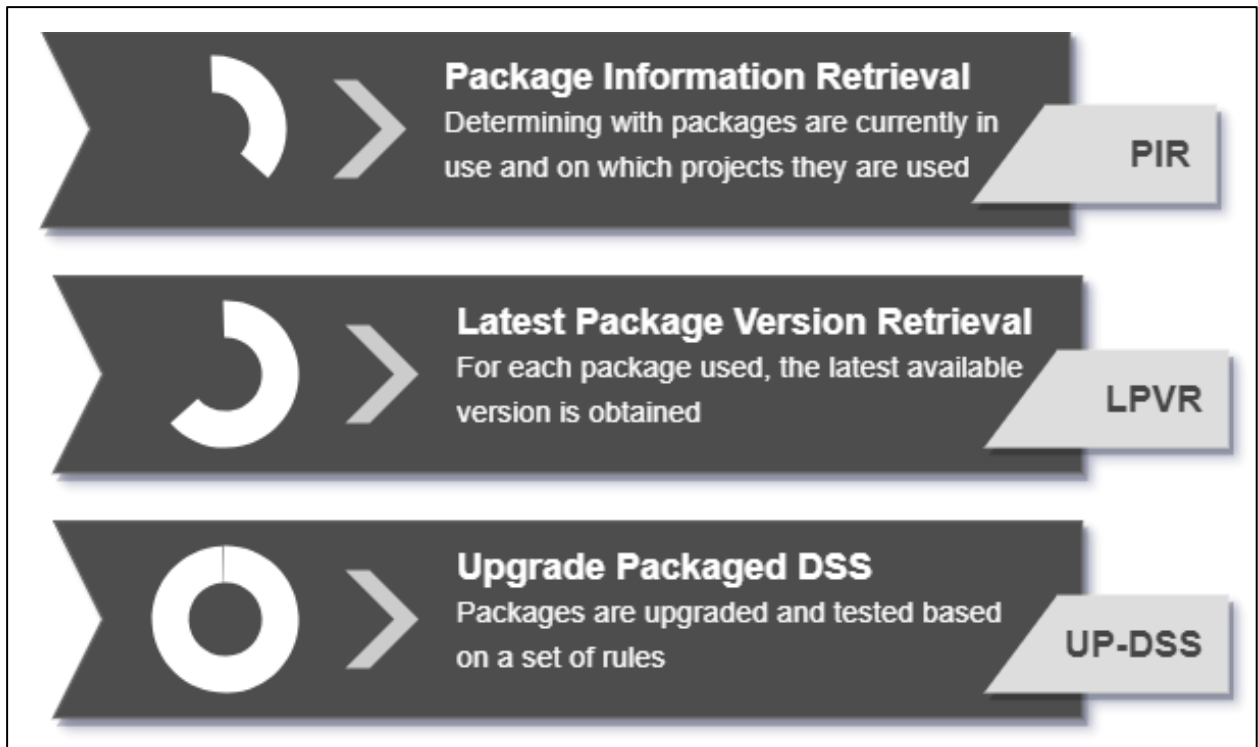


Figure 3-6: High-level process followed by the human and DPA solution

Figure 3-6 explains the rolled-up summary of the steps to be executed using both execution methods. The order of the steps that will be observed are as follows:

1. Retrieve a list of processes.
2. Retrieve a list of packages (or libraries) used within each process.
3. Retrieve the latest version for each package used.
4. If the latest version of a package is not the same as the current version, evaluate each process that uses the package according to the following business rules:
 - a. Is the process business critical?
 - b. Does the process have a QA environment?
 - c. Is the current library version less than one major version behind the latest version?
5. For each process and corresponding package that meets the above criteria, the package that has an upgrade available is upgraded and tested.
6. If the process ran successfully during the test, apply the upgrade to production.
7. If the process was unsuccessful during the test, apply a rollback.

3.5.1 Experiment 1: Human process execution data collection

In order to gather the human process execution data, SMEs are required to execute the process manually in order to record the cost, time and scope. The human execution of the process will be monitored between three people over the course of five days, over two iterations, resulting in ten days of human observation (one run per day), to ensure that the consistency of variables extracted from this method is also evaluated. An average of each variable will be obtained from the 30 execution runs, by the three humans.

3.5.2 Experiment 2: DPA solution process execution data collection

In order to gather the DPA solution data, the DPA solution needs to be built and executed in order to obtain the results suitable for measurement. The execution of the DPA solution will be monitored over the course of 30 days (one execution per day) to ensure that the consistency of variables extracted from this method is also evaluated. An average of each variable will be obtained from the 30 execution runs.

Once the data has been collected and validated, the data will be analysed as explained in the subsection below.

3.6 Participants

Swanepoel *et al.* (2021) define a population as a complete set of elements from which information may be obtained. They also define a sample as a subset of the population that is supported by Ackerman *et al.* (2021) and Asiamah *et al.* (2017). Asiamah *et al.* (2017) define the target population as the group of elements to which the study findings are generalised. They also define the accessible population as the portion of the population to which the researcher has reasonable access.

Morse (2000) states that the more data that is collected from participants, the fewer participants are needed as the scope of the study, the nature of the topic, the quality of the data and the design of study should all be taken into account when determining the appropriate sample size. This is supported by Malterud *et al.* (2016), who expand on this by introducing the concept of information power, which suggests that the more information power the sample provides, the smaller the sample size needs to be and vice versa. Francis *et al.* (2010) recommend that

cumulative frequency figures be presented by researchers to support their judgement that sampling more data would not lead to more information related to the research questions.

At the time of investigation into the case study environment, it was recorded that the population size would be six since there were six people within the environment who worked within the specific case study context. Although there were six employees, only three of the employees from the organisation were dedicated to executing the process; therefore, the target and accessible population is three people. The participants in the accessible population represent the diverse nature of the environment, in terms of skill level, seniority and experience in executing the process. This means that each participant has a different level of skill and experience (in terms of the organisational environment and the process). The selected participants will demonstrate the length and scope of the process being executed by a human as distinct data sources for the purpose of this study. In accordance with the approach suggested by Morse (2000), more data will be collected from each participant by increasing the number of process runs that each employee should execute to justify that the accessible population contains enough participants to gather meaningful data from.

Each employee will be observed executing the process ten times. Thereafter, the cost of having each participant (or employee) execute the process will be calculated. Since each employee will execute the process ten times, the resulting dataset will include 30 data points for each variable that would need to be analysed. In addition to this, to support the notion presented by Francis *et al.* (2010), cumulative graphs will be included during the data collection phase to justify that saturation has been achieved. All personally identifiable information of the participant will be excluded from the data collection process with specific care and attention placed on the collection of sensitive information, like an employee's rate per hour. The rate per hour will be collected on a historical basis (as data collection took place during a previous financial year), taking into account the level and role of the employee and the average rate per hour associated with that level and role, respectively. This removes the risk of sensitive information being tied to a specific employee, which would breach the ethical considerations of this study, further explained in sub-section 3.11.

The employees did not undergo any additional training in order to execute the process as part of this study. The selected process is a process that they currently perform as part of their work items. All employees included as participants of this study were aware that their execution runs were being timed. Although their level of experience differed, their backgrounds (relative to the study) were on par. The difference between each employee's skillset is attributed to their years of experience.

3.7 Research instruments

The data collection phase of this study will use a number of instruments to gather the data in the appropriate format as well as enhance the artefact as necessary. Hofstee (2015) describes a research instrument as any element that is used to conduct data gathering. The research instruments to be used through this study include task capture, joint application development (JAD) workshops and gap analysis.

3.7.1 Task capture

Jimenez-Ramirez *et al.* (2019) propose a method to improve the early stages of the automation lifecycle by leveraging the knowledge of the employees executing process manually. The idea behind the proposal is to monitor the actions of the employee, in a non-invasive manner (using screen recording, for example), which is then transformed into a process model. This is supported by Auth *et al.* (2019). Through their exploration of challenges with an automation lifecycle, Ubis *et al.* (2008) insist that standards are important. UiPath (2021) has developed the tool proposed by Auth *et al.* (2019), which is known as UiPath task capture.

The process discovery tool, known as UiPath task capture, records the user's interactions with each system and records the process steps, the process flow as well as detailed user system-interaction data (like number of windows used, number of keys pressed, number of mouse clicks, number of actions executed, number of text entries made, number of hotkeys used, number of tasks, number of processes and duration of process execution). This information will be used to understand the time and scope elements of each employee executing the process. Each employee will need to start the task capture recording before every process run in order to capture the information accurately. The data needed for the analysis of time and scope will be extracted from each of the 30 task capture documents (one per run) and presented in Chapter 4, as part of data collection. The costs associated with executing the process will be obtained directly from stakeholders within the case study environment.

3.7.2 Joint application development (JAD) workshops

Whitten *et al.* (2001) propose that joint application development (JAD) is a methodology used to encourage user involvement and participation. Carmel *et al.* (1993) expand on this by suggesting that the inclusion of users during the development process accelerates the design of systems as well as promote comprehensive and high-quality results. There are four building blocks of a JAD meeting (or workshop), namely:

- **Facilitation:** A facilitator will lead the meeting and ensure that each point on the agenda is discussed and discussion points are documented.
- **Agenda setting:** There should be a structured layout for what the meeting will address.
- **Documentation:** The outcomes of each workshop (or meeting) should be documented in detail.
- **Group dynamics:** Techniques are used to motivate and encourage creativity.

JAD workshops are held at different points throughout the development lifecycle (Carmel *et al.*, 1993). These workshops will be held in the form of system demonstrations and development retrospectives after each design and development iteration, within the case study environment. The stakeholders included in the system demos would be the BO, the PO, the scrum master and the project team. Retrospectives would include the product owner, the scrum master and the project team.

3.7.3 GAP analysis

Brown and Swartz (1989) suggest that gap analysis be used as a method to identify inconsistencies. The ability to identify gaps allows the team to formulate strategies and tactics to address (and mitigate) the gaps where possible. This notion is supported by Mineraud *et al.* (2016), who perform gap analysis on internet-of-things (IOT) platforms. Once these gaps are identified, they may be addressed between design and development iterations.

3.8 Data analysis

The analysis of the quantitative data gathered during this research entails the following:

- Comparing the average time duration of a person completing the process and completing the process through the use of a DPA solution.
- Comparing the estimated cost of a person completing the process and completing the process through the use of a DPA solution.
- Comparing the scope dynamics of a person completing the process and completing the process through the use of a DPA solution.

Based on the above and Figure 3-7, it can be summarised that there are three stages that need to be executed in order to collect and analyse the data appropriately, namely determine the human execution factors, determine the DPA execution factors, and determine the difference

between the execution methods. Figure 3-7 shows the breakdown of how to collect and analyse data appropriately.

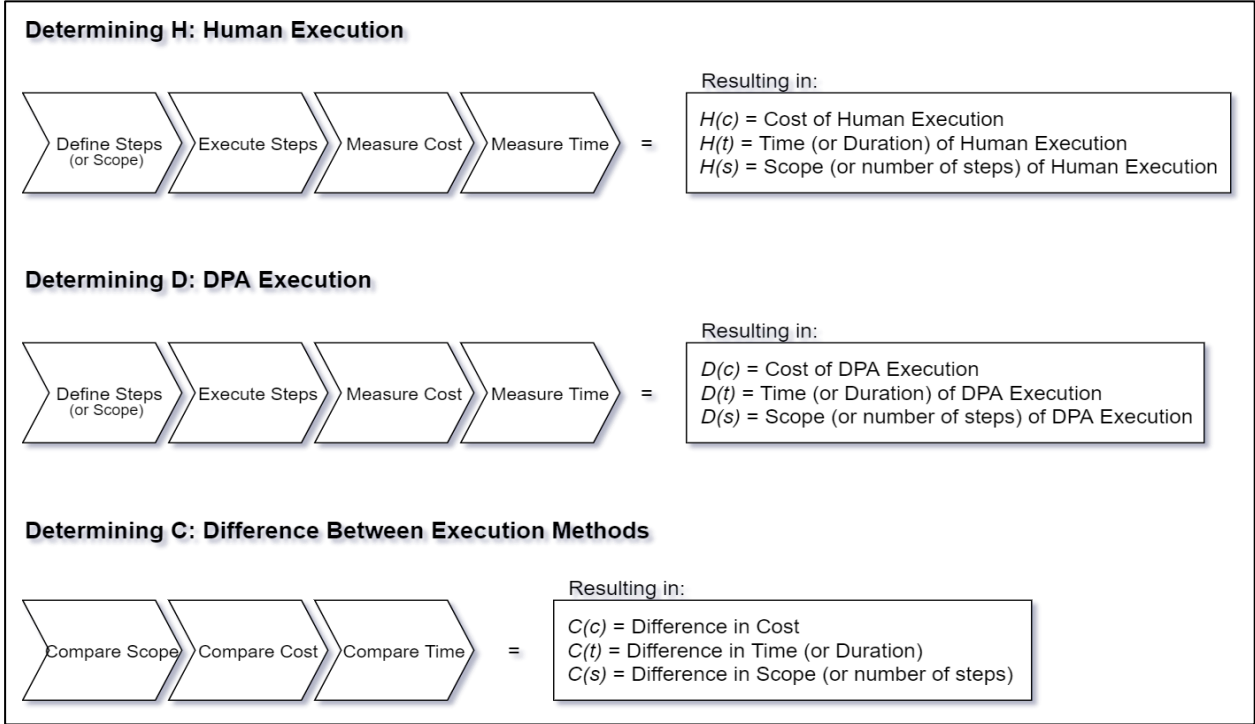


Figure 3-7: The determination of variables H, D and C in terms of c, t and s

As seen in Figure 3-7, collecting and determining each of these data points can be formalised into calculations. These calculations are summarised in Table 3-1 with regard to the following variable keys:

- H = Human execution of the process
- D = DPA solution execution of the process
- C = Difference between the execution methods
- c = Cost ($c_1 + c_2$)
 - c_1 = Direct Cost
 - c_2 = Indirect cost
- t = Time ($t_1 + t_2 + t_3$)
 - t_1 = Process preparation time
 - t_2 = Process execution time
 - t_3 = Process termination time
- s = Scope (in terms of the number of steps)
- P = Proposed value
- R = Realised value

Measures	Cost (c)	Time (t)	Scope (s)
Human execution (H)	$H(c)$	$H(t)$	$H(s)$
DPA solution execution (D)	$D(c)$	$D(t)$	$D(s)$
Delta (C)	$C(c) = H(c) - D(c)$	$C(t) = H(t) - D(t)$	$C(s) = H(s) - D(s)$
Realised value (R)	<p><i>Let $P(c) = C(c) > 0$</i></p> <p><i>$P(c) \rightarrow R(c)$, where $R(c)$ represents a realised cost benefit of automation. The value of $R(c)$ should either be 1 or 0 (True of False)</i></p>	<p><i>Let $P(t) = C(t) > 0$</i></p> <p><i>$P(t) \rightarrow R(t)$, where $R(t)$ represents a realised time benefit of automation. The value of $R(t)$ should either be 1 or 0 (True of False)</i></p>	<p><i>Let $P(s) = C(s) \geq 0$</i></p> <p><i>$P(s) \rightarrow R(s)$, where $R(s)$ represents a realised scope benefit of automation. The value of $R(s)$ should either be 1 or 0 (true of False)</i></p> <p>A difference in scope does not contribute to realised value for the process used within the case study. The scope is therefore excluded from the determination of realised value.</p>

Table 3-1: Tabular representation of analytical calculations to be performed

In a case where $R(c)$ or $R(t)$ is 0, the full benefit of automation was not realised and fell short in terms of value gain. The impact of the occurrence of a 0 for $R(s)$ is not as severe as it would be if $R(c)$ or $R(t)$ were 0 as it pertains to subjective benefits of possibly having more steps in a process (possibly because of additional quality assurance steps, for example) and not an increase in time or cost (which would impact the feasibility of implementing the DPA solution).

In a case where scope is deemed to be part of the realised value of a solution, the value of R would be calculated as $(R(c) \wedge R(t)) \wedge (R(s) \vee \neg R(s))$. Although a difference in scope can be measured through comparing the number of steps in each execution method, the difference does not necessarily contribute to the realised value within the context of the used case. The value of R , therefore, is determined by $(R(c) \wedge R(t))$. If $R = 1$, then there is an actual realised benefit of automating the solution. If $R = 0$, then there has not been a realised benefit of automating the solution.

3.9 Validity and reliability

Validity is carefully considered when researching research methodologies as it is used by the researcher to explain how they obtain and provide information. The researcher should acknowledge internal and external validity (Kolb, 2012). Internal validity is focused on data accuracy, which can be validated through checks and triangulation processes. Therefore, the researcher only considers data that is relevant to the study. Table 3-2 outlines the validity checks.

Criteria	Description
Credibility	Corbin and Strauss (2014) propose this as establishing trustworthiness and linking findings with observations of the case study
Confirmability	Confirmability is linked to the concern of objectivity. Data should be traceable from the source to the target and vice versa (Shenton, 2004).
Dependability	The enhancement of dependability is done through an audit report (Bashir <i>et al.</i> , 2008).
Transferability	Corbin and Strauss (2014) suggest that transferability takes care of the external validity by providing background data and context for credible comparisons to be done as a part of the research.

Table 3-2: Tabular representation of the validity and reliability

According to Kimberlin and Winterstein (2008), reliability is the characteristic of any score that is obtained by an instrument used for measurement or collection of data, also known as the observed score, which is composed of both the “true” score, and “error” in the measurement process. In this study, the observed scores will be retrieved from the logs and visualised as successes and errors.

3.10 Limitations of study

Since this is a case study on an ICT organisation environment, the scope of this study is specific to governing standards and parameters put in place by the specific business unit within the ICT organisation.

The case study in question is subjected to the following limitations:

- A predefined process will be evaluated by the DPA solution as well as measured against human execution.
- The abovementioned process is a DevOps process that has a clear set of steps.
- The approach taken to develop the DPA solution, instead of using pre-existing tools, is due to the brief of the case study provided by the ICT organisation as it takes into account the

circumstances and support structures that are currently in place. Sub-section 3.2 elaborates more on this.

- The elements to be evaluated are limited to cost, scope and time.

3.11 Ethical considerations

Arifin (2018) proposes that ethical considerations be used as principles that researchers acknowledge and take responsibility for while conducting their research. These principles are aimed at maintaining the validity, objectivity, best practices and participant protection throughout a study (Zyphur & Pierides, 2017). Myers and Venable (2014) investigated the ethical principles used for design science research and propose the use of the principles suggested by Mason (1986), which are:

- Understanding, protecting and safeguarding all privacy.
- Gathering accurate data and validating the accuracy.
- Understanding the ownership and consequences of intellectual property.
- Understanding, defining, maintaining and auditing the access to the collected data.

According to Driscoll and Brizee (2012), ethical considerations include:

- Ensuring that the opinions of the researcher are not biased and do not affect the study.
- Ensuring that the research is accurate.
- Informing all of the participants about their anonymity during the study.
- Ensuring that the participants do not have emotional or physical harm inflicted upon them through the study.
- Ensuring that permission is obtained from all participants as well as any additional stakeholders who may be affected (like managers).

In addition to the ethical considerations above, the considerations that should also be respected, from a North-West University (NWU) perspective, include acquiring ethical clearance from NWU and acquiring approval from the NWU Faculty of Natural and Agricultural Sciences Research Ethics Committee (FNASREC) to collect data from participants.

A letter of permission has been issued by the ICT organisation, allowing the researcher access to the voluntary and anonymous participation of its employees. No information presented in this study may be linked to an employee of the organisation as per the agreement signed between the organisation and the researcher. The necessary precautions are to be taken by the researcher to anonymise the participation of the employees by removing any personally identifiable

information during the data collection phase of the study. The letter of permission has been provided to the North-West University ethics committee who has issued ethical clearance for this study to proceed (see Annexure G and H).

3.12 Conclusion

The dissection of the research onion throughout the chapter positions this study as having a pragmatic philosophy, an inductive approach, with experiments conducted within a case study focused strategy, while making use of DSR. The use of frameworks like SAFe and TOGAF will cater for the iterative development of the artefact needed for data collection, in line with the purpose of the DSR design cycle. The research instruments defined should enable the collection of data, in the different forms that it may be presented in. The observation of human execution will make use of task capture while the data collected from the DPA solution execution (including development and evaluation) would use GAP analysis and JAD workshops. Once the data has been collected, it will be analysed in order to provide the realised saving (if any). Ethical considerations as well as validity and reliability are to be used and evaluated throughout this study. The output of this chapter is aligned to the 'suggested methodology' proposed by the DSR study layout, illustrated in Figure 3-8.

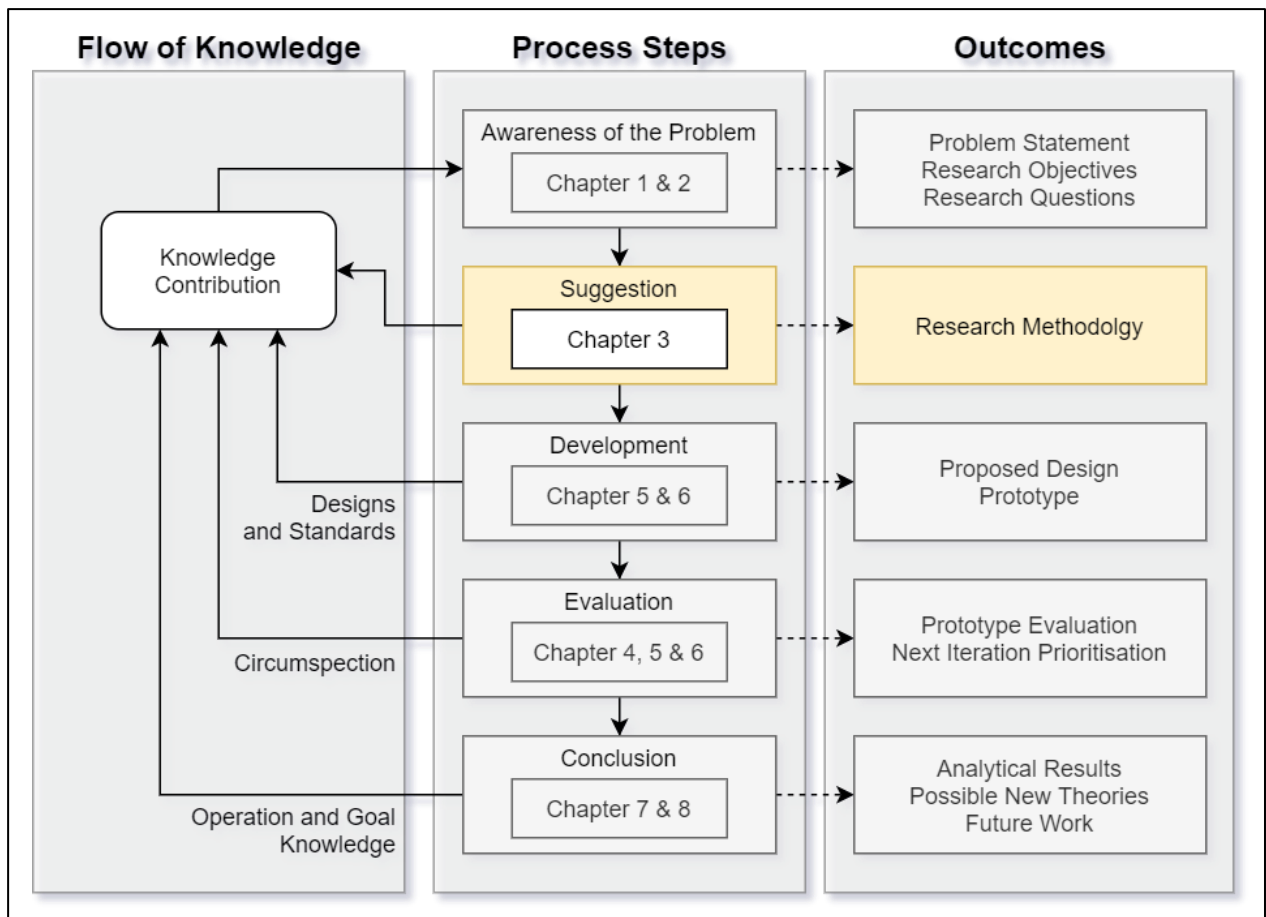


Figure 3-8: Chapter 3 DSR study layout contribution

Chapter 4 introduces the human execution data collection, which will detail how data pertaining to cost, time and scope will be collected as part of the 'evaluation' step in the DSR study layout.

CHAPTER 4: HUMAN EXECUTION DATA COLLECTION

4.1 Introduction

The process at hand is executed manually by individuals working within the case study environment. The study is focused on collecting three variables from observing human execution of the process, namely time, scope and cost. This chapter is focused on the collection of the abovementioned variables in preparation for comparison with the variables gathered while observing the execution of the DPA solution. Three employees that work within the case study environment were used as participants of the study in order to collect the relevant data.

After requirements had been gathered, a change in scope was introduced. The process steps remained the same; however, there was a change in technology that resulted in the need for a second development iteration. Both development iterations are detailed in the chapters that follow. The process observed in the latest version of the process is post scope change. The process in question focuses on updating Python packages automatically, based on predetermined business rules. The process that is executed manually as part of the data collection portion of the study is outlined in Figure 4-1. 33, consisting of 57 libraries to be evaluated as part of this process (as seen in Annexure D).

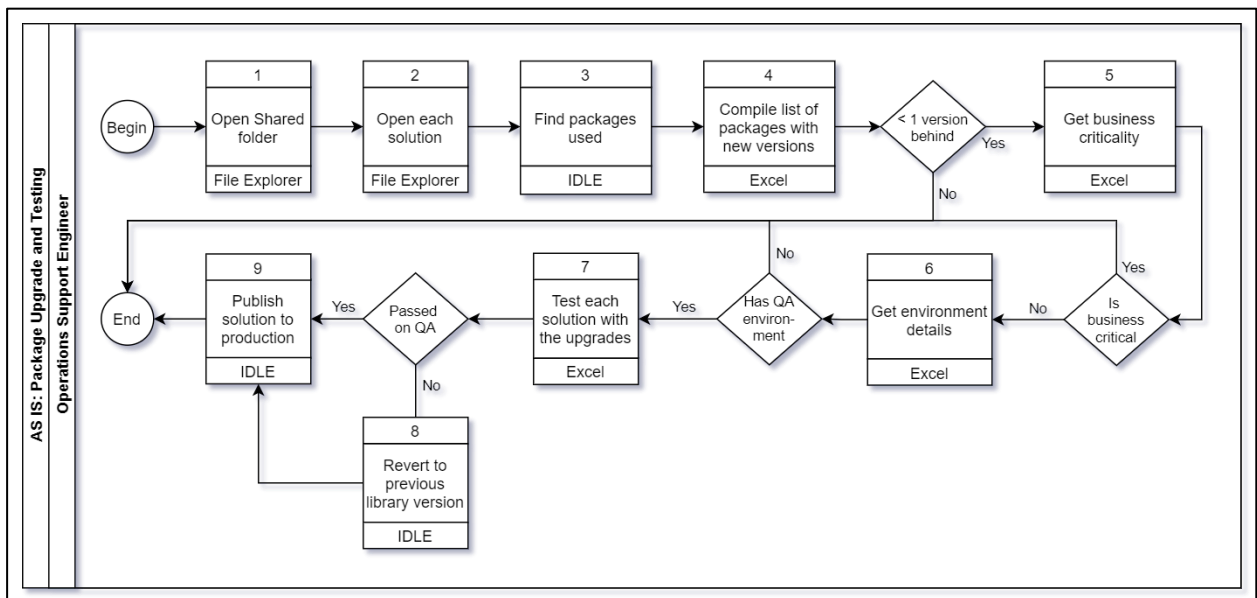


Figure 4-1: The process steps to be executed manually (AS IS process)

Annexure D contains the Excel workbook that is used by each employee to gather and store the information necessary to identify whether or not a package upgrade is applicable. The packages and projects listed in the Annexure D are the same packages and projects to be used during the

DPA solution execution testing. The use of the workbook ensures that the employees execute the exact same process steps (as listed in Figure 4-1) while still allowing for task variations between employees (for example, one employee minimising applications using mouse clicks and another using hotkeys or keyboard shortcuts).

Sub-section 4.2 expands on the data collection of time, cost and scope with 4.3 concluding the chapter. This chapter aims to evaluate the factors that contribute to time, cost and scope of employees executing a process, in line with the DSR layout.

4.2 Data collection

The observation of the process being manually executed is used to collect the cost, time and scope variables of human execution (Determining H: Human Execution, as seen in Figure 4-2 adapted from Figure 3-7). Figure 4-2 explains that human execution will be observed throughout the process of defining the steps (as done in Figure 4-1), executing the steps, measuring the cost of each employee executing the process and measuring the time it takes each employee to execute the process. UiPath task capture will be used to record each employee executing the process for every process run, outputting a document from which the data will be extracted and presented throughout this chapter. The white shapes in Figure 4-2 indicate which variables still need to be determined, while the yellow shapes indicate which variables will be determined in this chapter.

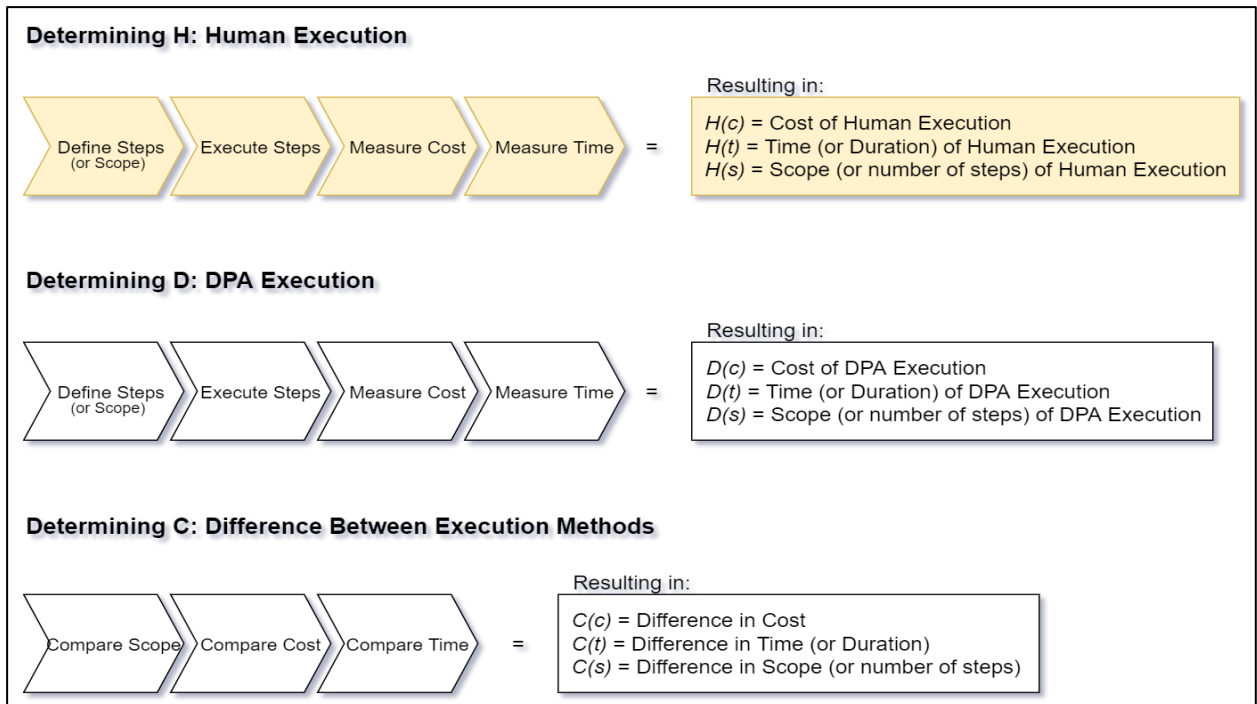


Figure 4-2: The determination of variable, H, in terms of c, t and s for human execution

As seen in Figure 4-2, collecting and determining each of these data points can be formalised into calculations. The variable keys relevant to Determining H: Human Execution are expanded below:

- H = Human execution of the process
- $H(c)$ = Cost ($c_1 + c_2$)
 - c_1 = Indirect cost
 - c_2 = Direct cost
- $H(t)$ = Time ($t_1 + t_2 + t_3$)
 - t_1 = Process preparation time
 - t_2 = Process execution time
 - t_3 = Process termination time
- $H(s)$ = Scope (in terms of the number of steps)

	Cost (c)	Time (t)	Scope (s)
Human execution (H)	$H(c)$	$H(t)$	$H(s)$

Table 4-1: Tabular representation of analytical calculations to be performed while determining A through human execution

Three employees within the case study environment were participants of the study as they demonstrated the manual human execution of the process. All identifiable information pertaining to the employees has been removed to adhere to proper data governance and as such, the sum of the data gathered (pertaining to each employee) is subject to the following:

$$\sum_{n=1}^k (e_n)$$

Where:

- k refers to the upper limit of study participants, which in this study is three.
- n refers to the data captured by observing an employee.
- e refers to an employee as an entity/variable in and of itself.

The abovementioned formula aims to represent data points specific to an employee so that the collection of the time, cost and scope variables remains applicable and relative to the same persistent context. Since execution time is also used in the measurement of direct cost, it is important that the time and rate of each employee are calculated separately as failure to do so would lead to an inaccurate representation of the data collected.

4.2.1 Time

The execution time of the process may include multiple factors, which may vary between use cases and may also be adapted to exclude wait time (the amount of time spent waiting for an action to be executed before continuing with the process. For example, waiting for the results from the pip command). For the sake of this study, only execution time will be considered (excluding any wait time). Within the context of this study, wait time would refer to the time spent waiting for an action to be executed (like loading data into a report, for example) in which an employee would execute on other actions (like scrolling through Facebook or taking a coffee break, for example). Time needs to be taken into consideration per employee executing the process; therefore, a variable for employee (e_n) is introduced. The time calculation is broken up as follows:

Time = $t_1 + t_2 + t_3$, in terms of employee (e_n)

Therefore,

$$te_n = t_1 + t_2 + t_3$$

$$H(t) = \frac{\sum_{n=1}^k (te_n)}{k}$$

Where:

- e_n = The employee executing the process
 - This study contains three participants (or employees), which means that n would be evaluated as e_1, e_2 or e_3 as n refers to a specific employee.
- k = the upper limit of employees participating in the study, which in this study is three
- t_1 = Process preparation time
 - Includes tasks like:
 - Opening the shared folder
 - Opening each solution
 - Opening lists
- t_2 = Process execution time
 - Includes tasks like:
 - Evaluating business rules
 - Testing the upgrades
 - Initiating rollbacks
 - Applying changes to production
- t_3 = Process termination time
 - Includes tasks like:
 - Cleaning the environment after the rollback
 - Reporting on progress
- $H(t)$ = Average duration
 - The duration (or time) for human execution will be evaluated as an average across the number of participants, provided that there are no outliers that skew the data. In a case where outliers occur, the outliers would need to be excluded or collected again.

Preparation, execution and termination may be split up to contain different tasks depending on the scope of the process. Preparation could typically include getting the Excel workbook ready for use and comparison before the upgrade is executed. Execution would include the steps that work towards executing the upgrade. Termination would refer to closing all open applications and

ensuring that the Excel workbook is saved and that the environment is returned to the appropriate state for the next process run.

Each employee executed the process ten times, with two runs focused on a set number of rollbacks and is broken up as follows:

- Runs 1 and 2 have 4 rollbacks
- Runs 3 and 4 have 3 rollbacks
- Runs 5 and 6 have 2 rollbacks
- Runs 7 and 8 have 1 rollback
- Runs 9 and 10 have 0 rollbacks

The goal of splitting runs by the number of rollbacks is to identify what the impact would be of changing the scope (in this case, number of rollbacks). Table 4-2 summarises the results from the human execution data collection of the time (in seconds) that it takes each employee to execute the process.

			Runtime per run (in seconds)			
Employee	Run	Rollbacks	$t_1 =$ Preparation	$t_2 =$ Execution	$t_3 =$ Termination	Time (t) = $t_1 + t_2 + t_3$
Employee 1 (e_1)	Run 01	4	1 030	159	482	1 671
	Run 02	4	1 052	152	476	1 680
	Run 03	3	1 046	149	483	1 678
	Run 04	3	1 060	149	476	1 685
	Run 05	2	1 021	139	478	1 638
	Run 06	2	1 035	131	481	1 647
	Run 07	1	1 023	118	472	1 613
	Run 08	1	1 042	123	483	1 648
	Run 09	0	1 026	72	439	1 537
	Run 10	0	1 058	69	450	1 577
Employee 2 (e_2)	Run 01	4	710	145	377	1 232
	Run 02	4	742	147	383	1 272
	Run 03	3	695	141	369	1 205
	Run 04	3	699	138	374	1 211
	Run 05	2	718	129	375	1 222
	Run 06	2	732	133	382	1 247
	Run 07	1	692	116	371	1 179
	Run 08	1	701	121	377	1 199
	Run 09	0	733	72	341	1 146
	Run 10	0	718	65	325	1 108
Employee 3 (e_3)	Run 01	4	925	280	294	1 499
	Run 02	4	930	272	276	1 478
	Run 03	3	927	249	282	1 458
	Run 04	3	929	256	281	1 466
	Run 05	2	932	237	279	1 448
	Run 06	2	925	241	292	1 458
	Run 07	1	918	224	285	1 427
	Run 08	1	920	220	290	1 430
	Run 09	0	912	193	281	1 386
	Run 10	0	907	195	276	1 378

Table 4-2: Tabular representation of human process execution time (in seconds)

Figure 4-3 aims to demonstrate the consistency in time between runs (in terms of the number of rollbacks). There is a noticeable trend that occurs as the number of runs increases. The average preparation time between employees per run is consistent with little change. The average execution time increases very slightly with an increase in rollbacks. The most noticeable difference in execution time occurs when there are no rollbacks, which is expected since fewer process steps are executed when there are no rollbacks. The same consistency for termination time is observed – the termination time is less when there are no rollbacks because there are

fewer actions to revert before terminating the process. Although there is a slight difference in the average duration when there are no rollbacks, the difference between no rollbacks and four rollbacks is approximately 116 seconds, which is only about 8% of the average duration of the process.

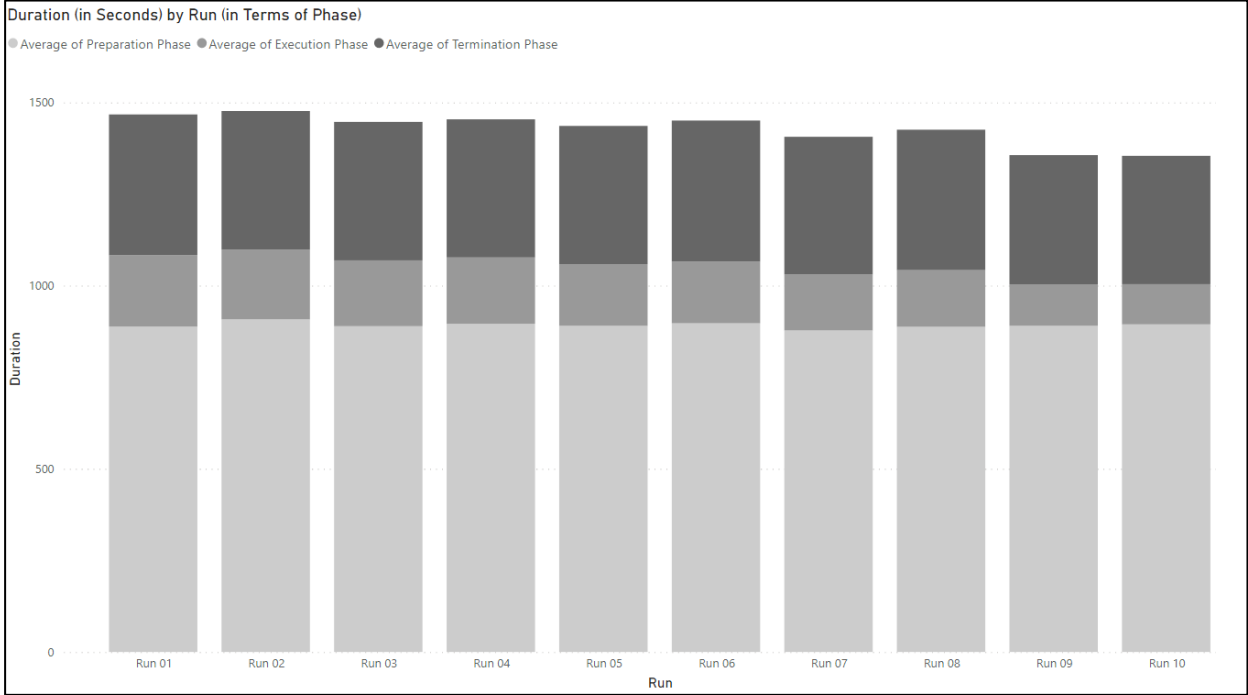


Figure 4-3: The duration (in seconds) by run (in terms of phase)

Figure 4-4 shows the duration of the preparation, execution and termination phase for each run by each employee. It can be seen that there are no significant outliers that could skew the averages deduced from the collected data.

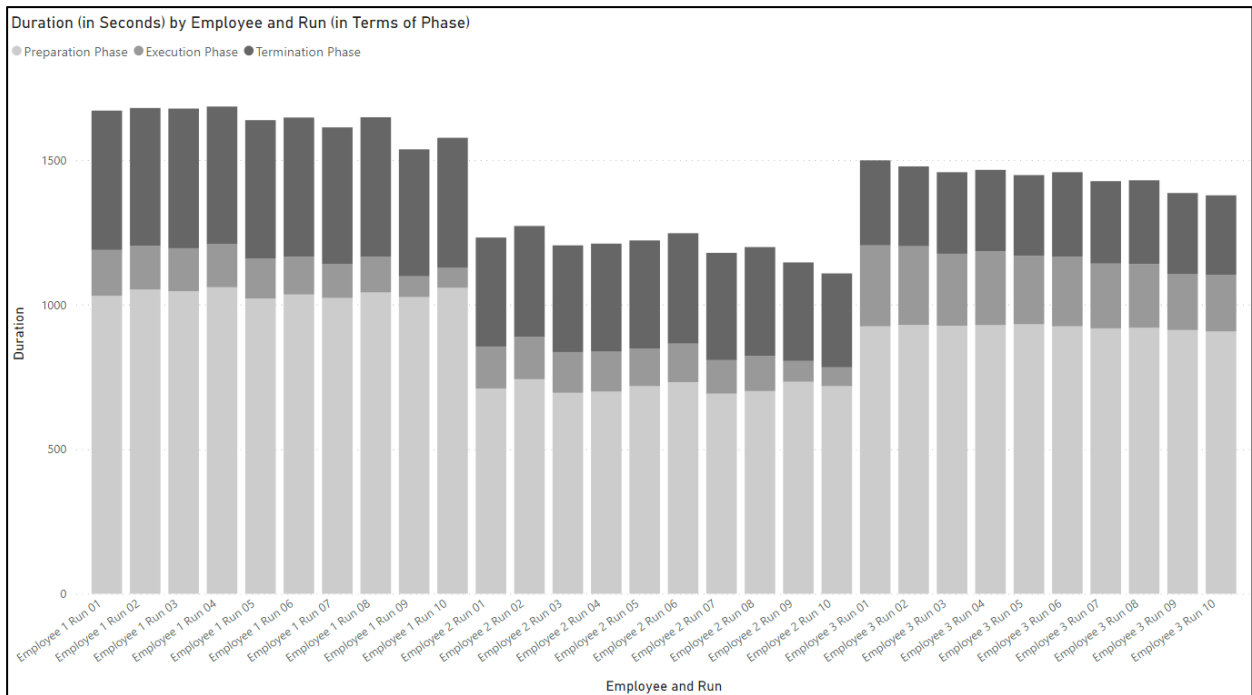


Figure 4-4: The duration (in seconds) by employee and run (in terms of phase)

From Figure 4-4, there is a noticeable difference in duration between employee 2 and the other employees. Figure 4-5 also shows that employee 2 executes the process faster than the other two employees by approximately six minutes (340 seconds). Employee 1 executes the process the slowest, trailing employee 2’s average duration by approximately three minutes (190 seconds). This could be due to each employee’s level of seniority or perhaps even their experience executing the process (including the way that each employee interacts with the different applications). It is interesting that employee 3 has a higher average duration for the execution phase and a lower average duration for the termination phase than the other two employees – this is most likely due to the way that the employee interacts with the applications in use.

There are three different noticeable trends that occur in Figure 4-4 and each trend correlates with an employee. The three trends may be justified by the different levels of seniority of each employee (supported by the rate per hour included in the direct cost calculation, as seen in Figure 4-6) as well as duration of each employee executing the process.

Beyond the case study environment, the difference in time may be measured between one employee and the DPA solution. The ideal comparison would be to compare the execution time of the DPA solution with the employee who executes the process the fastest, or possibly the cheapest, depending on the variable that should be optimised. The fastest option may not necessarily be the cheapest option.

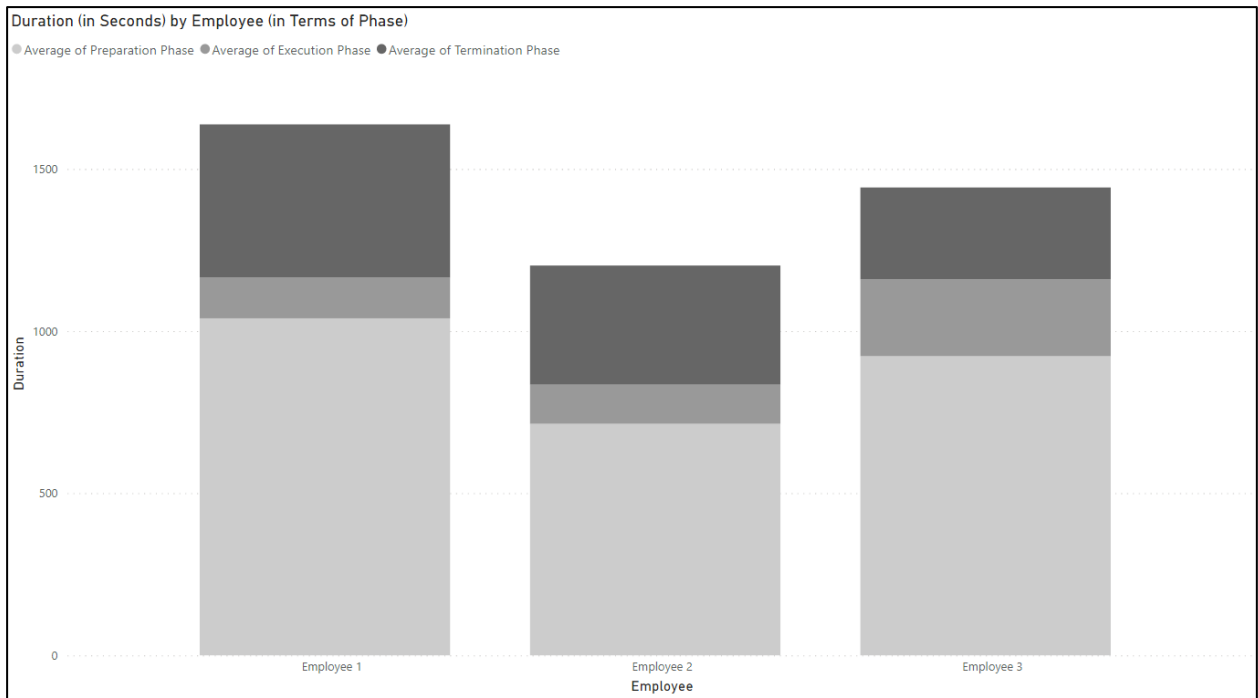


Figure 4-5: The duration (in seconds) by employee (in terms of phase)

In order to appropriately evaluate the difference in time and cost between execution methods, an average across runs needs to be obtained per employee. The duration of process execution per employee is taken into consideration when calculating direct cost. Table 4-3 represents the average duration split up between the preparation, execution and terminate phases, per employee.

Phase	Runtime per run		
	Employee 1 (e_1)	Employee 2 (e_2)	Employee 3 (e_3)
t_1 = Preparation	17 minutes 19 seconds	11 minutes 54 seconds	15 minutes 22 seconds
t_2 = Execution	2 minutes 6 seconds	2 minutes 1 second	3 minutes 57 seconds
t_3 = Termination	7 minutes 52 seconds	6 minutes 7 seconds	4 minutes 44 seconds
Time: $te_n = t_1 + t_2 + t_3$ In terms of employee (e_n)	27 minutes 17 seconds	20 minutes 2 seconds	24 minutes 3 seconds
Average time: $H(t) = \frac{\sum_{n=1}^3(te_n)}{3}$ In terms of employee (e_n)	23 minutes 47 seconds		

Table 4-3: Tabular representation of human process execution time

It can be seen from Table 4-3 that, on average, it takes approximately 23 minutes and 47 seconds to execute the process manually (as human execution) per run. Since this process is run once a week, every week (52 runs per year), the total execution time of this process in a year results to approximately 20 hours 37 minutes.

4.2.2 Cost

The cost of execution can be broken up into direct and indirect costs, which may be adapted as necessary for different use cases. Some use cases may, additionally, consider the cost of employee overtime, employee absence, consumption of office facilities, fringe benefits, etc.

For the sake of this study, cost is calculated as follows:

$$H(c) = \text{Average Cost } (c_1 + c_2)$$

- c_1 = Average direct cost

$$c_1 = \frac{\sum_{n=1}^k (te_n \cdot re_n)}{k}$$

- Where:
 - k = the upper limit of employees participating in the study, which in this study is three
 - te_n = execution time per employee
 - re_n = direct labour (rate per hour by number of hours spent on execution)

- c_2 = Average Indirect Cost

$$c_2 = \frac{\sum_{n=1}^k \frac{(i_1 + i_2)}{h} te_n}{k}$$

- Where:
 - k = the upper limit of employees participating in the study, which in this study is three
 - h = the number of working hours per annum
 - Within the context of the case study environment, the employee contracts state that a minimum of 160 hours per month should be worked. Therefore, h is calculated by multiplying 160 by 12, resulting in 1 920 working hours per annum.
 - Infrastructure costs ($i_1 + i_2$)
 - (i_1) = Hardware expenses (per annum)
 - (i_2) = Licensing expenses (per annum)
 - te_n = execution time per employee

The direct cost of the human execution of the process represents the cost of direct labour, which refers to the cost of having a human execute the process. The total direct cost of human execution is calculated by multiplying the number of hours it takes each employee to execute the process manually by their rate per hour, outlined in Table 4-4. The rate per hour was collected on a historical basis (as the data collection of cost took place during a different financial year as analysis), taking into account the level and role of the employee and the average rate per hour associated to that level and role, respectively.

	Direct cost per employee (in ZAR) per run		
Direct cost	Employee 1 (e_1)	Employee 2 (e_2)	Employee 3 (e_3)
Rate per hour, per employee (re_n)	R161.40	R207.70	R173.08
Execution time, per employee (te_n)	0.45 hours	0.33 hours	0.41 hours
Total direct cost: $c_1 = te_n \times re_n$	R73.41	R69.35	R69.37
Average direct cost: $c_1 = \frac{\sum_{n=1}^3(te_n \cdot re_n)}{3}$	R70.71		

Table 4-4: Tabular representation of human process execution direct cost

Total direct cost, based on the average of all three employees, equates to R70.71 per run. In a case where the process is run once a week (52 weeks per year), the total direct cost per annum is approximately R3 677.00. Figure 4-6 shows the difference in total direct cost and rate per hour for each employee. From Figure 4-6, it can be seen that employee 2 earns a higher rate per hour than the other two employees, whereas employee 1 earns a lower rate per hour than the other two employees. The total direct cost for all three employees remains constant with employee 1 having the highest direct cost and employee 2 having the lowest direct cost. For this to occur, it would mean that employee 2 executes the process the fastest and employee 1 executes the process the slowest, on average, which is supported by the time data collection. It can be seen from Table 4-4 that employee 2 executes the process in 0.41 hours (on average) and employee 1 executes the process in 0.45 hours (on average).

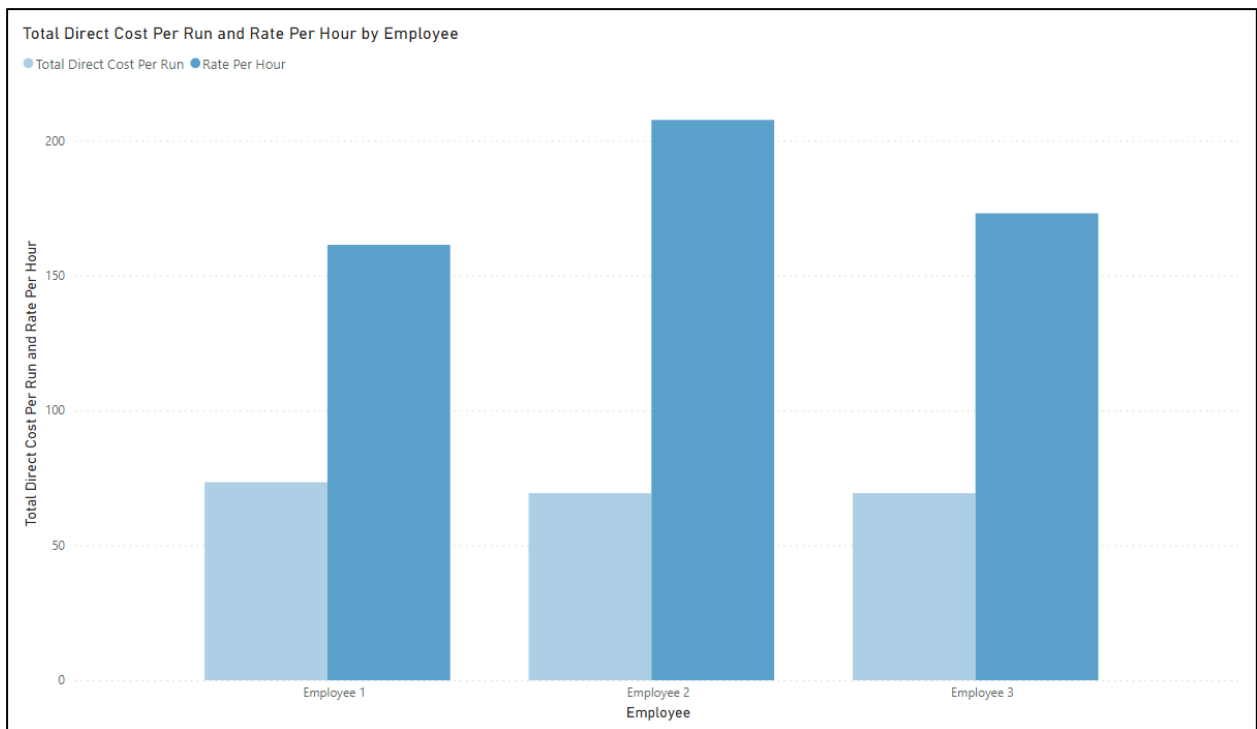


Figure 4-6: Total direct cost per run and rate per hour by employee

The indirect cost of human execution represents the operational cost of having people execute the process. The costs include hardware and licensing expenses for each employee.

Although employees have licenses that they may not necessarily use when executing this process, all licensing costs still need to be taken into account when calculating the return on investment. Licensing costs for employees within the case study environment include:

- **Microsoft E5:** The Microsoft E5 license allows a user access to Office, Microsoft Teams, OneDrive for Business, On-Premise Active Directory sync for single sign-on (SSO), a Power BI Pro subscription and many more. The team predominantly uses the abovementioned features to conduct day-to-day tasks and the E5 license is assigned to all employees within the organisation by default.
- **UiPath user license:** The developers within the case study environment develop automations in UiPath for which they require a developer/user license.
- **ServiceNow standard user license:** Each developer that performs DevOps activities requires access to ServiceNow to monitor any tickets that have been opened and should be updated with progress on the issues. Any user who wants to log tickets or work on tickets needs a ServiceNow license. ServiceNow is the ITSM platform of choice within the ICT organisation.

- **Jira Align user license:** Jira Align is used on organisational level to maintain visibility on PI progress among teams, centrally. Each user who works on a PI project would require a Jira Align license. Jira Align is used alongside Azure DevOps. Azure DevOps is used on a more granular level, bringing together the artefacts of a solution (code repositories, test plans, pipelines and releases).
- **Visual Studio Enterprise subscription:** The Visual Studio subscription allows a user access to Visual Studios, Azure DevOps and monthly Azure credits. Employees in the case study environment utilise all three aspects of the subscription across projects.
- **Workday user license:** A user requires a Workday license in order to login and interact with their profile and human resource-related matters, like booking leave, retrieving payslips and more. Workday is the platform selected by the ICT organisation to enable employees to centrally engage with their information.

The license costs listed above are publicly available either through the request of a quote or on their respective websites. These costs could also be subject to enterprise agreements between an organisation and each vendor, depending on the case study environment. No enterprise agreement discounts were taken into account for this study.

Licenses are often purchased on a yearly basis as they work out cheaper than month-to-month subscriptions. In a case where licenses are purchased on a month-to-month basis, the cost would need to be converted to annual costs in order to calculate the cost per run on the same scale. The cost of a license not being used (because of a process execution that does not utilise that license), still needs to be used when calculating the price per run as it can be considered as sunken cost. The alternative would be to categorise sunken cost separately and calculate all sunken cost across all processes, per annum, separately.

	Indirect cost per employee (in ZAR)		
Indirect cost	Employee 1 (e_1)	Employee 2 (e_2)	Employee 3 (e_3)
Infrastructure cost	R140 692 per year	R140 692 per year	R140 692 per year
Hardware (i_1)	R32 000 per year	R32 000 per year	R32 000 per year
Licensing (i_2)	R108 692 per year	R108 692 per year	R108 692 per year
Total indirect cost (per year)	R140 692 per year	R140 692 per year	R140 692 per year
Total indirect cost (per run): $c_2 = \frac{(i_1 + i_2)}{1\,920} \times te_n$	R32.97 per run	R24.18 per run	R29.31 per run
Average indirect cost (per run): $c_2 = \frac{\sum_{n=1}^3 \frac{(i_1 + i_2)}{1\,920} \times te_n}{3}$	R28.82 per run		

Table 4-5: Tabular representation of human process execution indirect cost

Total indirect cost (as seen in Table 4-5), on average, equates to approximately R28.82 per run and approximately R1 499.00 per annum (based on 52 runs per year).

The average cost of this solution per run is, therefore, calculated as follows:

$$c = (c_1 + c_2)$$

$$c = R70.71 + R28.82$$

$$c = R99.53$$

The auto-package upgrade process is to be executed once a week, every week throughout the year. This results in 52 process runs in a year, meaning that this process costs approximately R5 176.00 per annum.

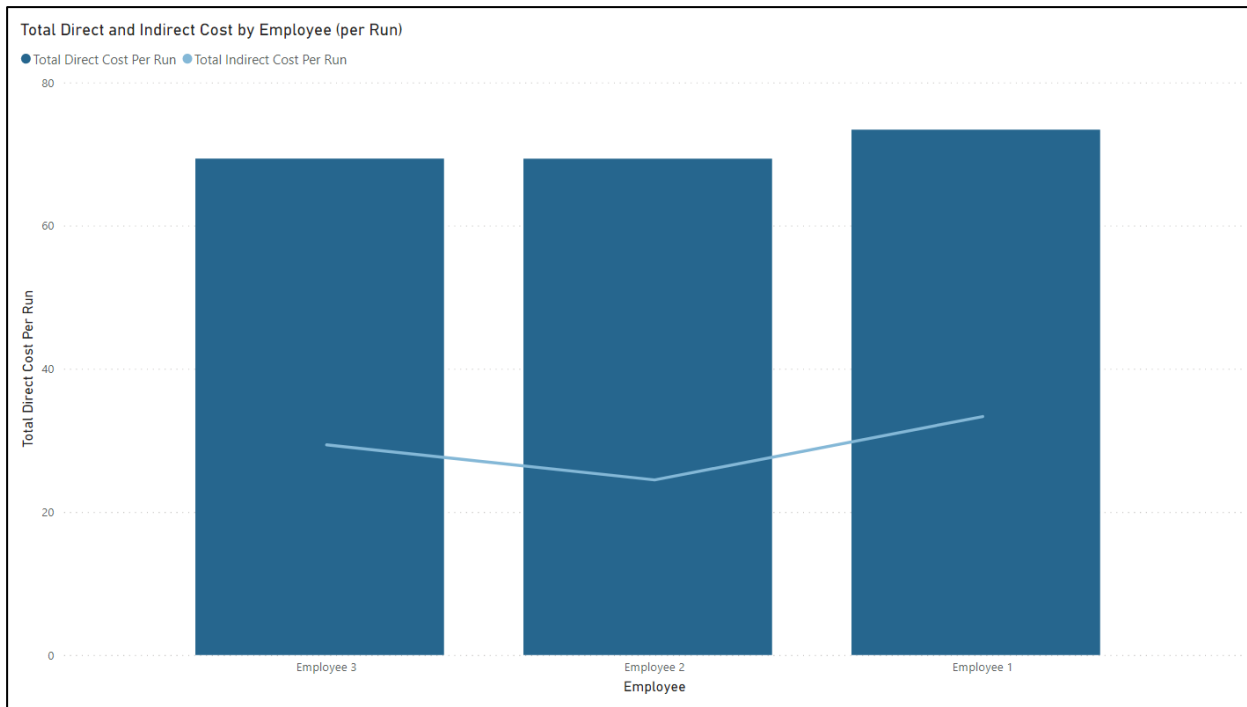


Figure 4-7: Total direct and indirect cost by employee (per run)

Figure 4-7 illustrates the total direct and indirect cost by employee, per run. It can be seen that the indirect cost and direct cost vary between employees. Employee 1 has the highest direct cost of R73.41, while employee 2 has the lowest direct cost of R69.35, which is only 2c lower than employee 3, who has a direct cost of R69.37. The variance in direct cost may be explained by the variable factors that influence direct cost, which include an employee's rate per hour, an employee's level of seniority and the time it takes an employee to execute the process. The indirect cost of employee 1 is the highest at R32.97 per run with employee 2 being the lowest with R24.18 per run. The indirect cost of employee 3 is R29.41 per run.

4.2.3 Scope

Annexure D was used by each employee to reference, search and update package and project-specific information that was necessary to identify whether or not a package should be upgraded.

$$H(s) = \text{Number of process steps used to execute the process.}$$

Each employee executed their own variation of the steps that resulted in a varying number of total tasks executed in order to adhere to the process steps. The execution of each process run, by each employee, was recorded using UiPath task capture. A table detailing the number of windows, actions, mouse clicks, keys pressed, text entries, hotkeys used, tasks and processes

was a section output of the exported task capture document. The number of processes executed by each employee is essential in verifying that the process was executed correctly. The number of average steps recorded per employee are summarised in Table 4-6.

	Employee 1 (e_1)	Employee 2 (e_2)	Employee 3 (e_3)	Average
Windows	35	33	21	30
Actions	494	303	522	440
Mouse clicks	340	165	152	219
Keys pressed	395	190	1030	538
Text entries	78	41	117	79
Hotkeys used	27	20	37	28
Tasks	136	124	122	127
Processes	9	9	9	9
Scope ($H(s)$)	9	9	9	9

Table 4-6: Tabular representation of average human process execution scope

As seen in Table 4-6, the average number of processes executed by each employee is equal to the number of process steps recorded in Figure 4-1, which indicates that the same number of steps was executed by each employee, with each employee executing a different number of clicks, actions and text entries in order to complete the process correctly. Each run by each employee recorded nine process steps being included in the process execution, even though runs with no rollbacks did not action the rollback process steps.

4.3 Conclusion

This chapter was focused on the collection of the cost, time and scope variables in preparation for comparison with the variables gathered while observing the execution of the DPA solution. Three employees who work within the case study environment were used as participants of the study in order to collect the relevant data. The gathered data has been presented as a ‘per run’ measurement as well as on a ‘per annum’ measurement in this chapter in order to compare the different execution methods effectively.

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influence time are: process preparation time, process execution time and process termination time. These three phases equate to the total duration (or time) of having a human execute the process. The duration had to be calculated in hours to cater for the cost calculation.

From the collected data, it can be seen that cost is influenced by direct and indirect cost. Direct cost is affected by two parameters, namely the time it takes each employee to execute the process (in hours) as well as the rate per hour. Indirect cost consists of infrastructure costs (specifically referring to hardware and software licensing costs).

The scope of the process execution is determined by the number of steps that are executed. The difference in the duration of each employee executing the process may be justified with how each step was executed by the employees, but ultimately the number of process steps remains the same between each of the 30 runs.

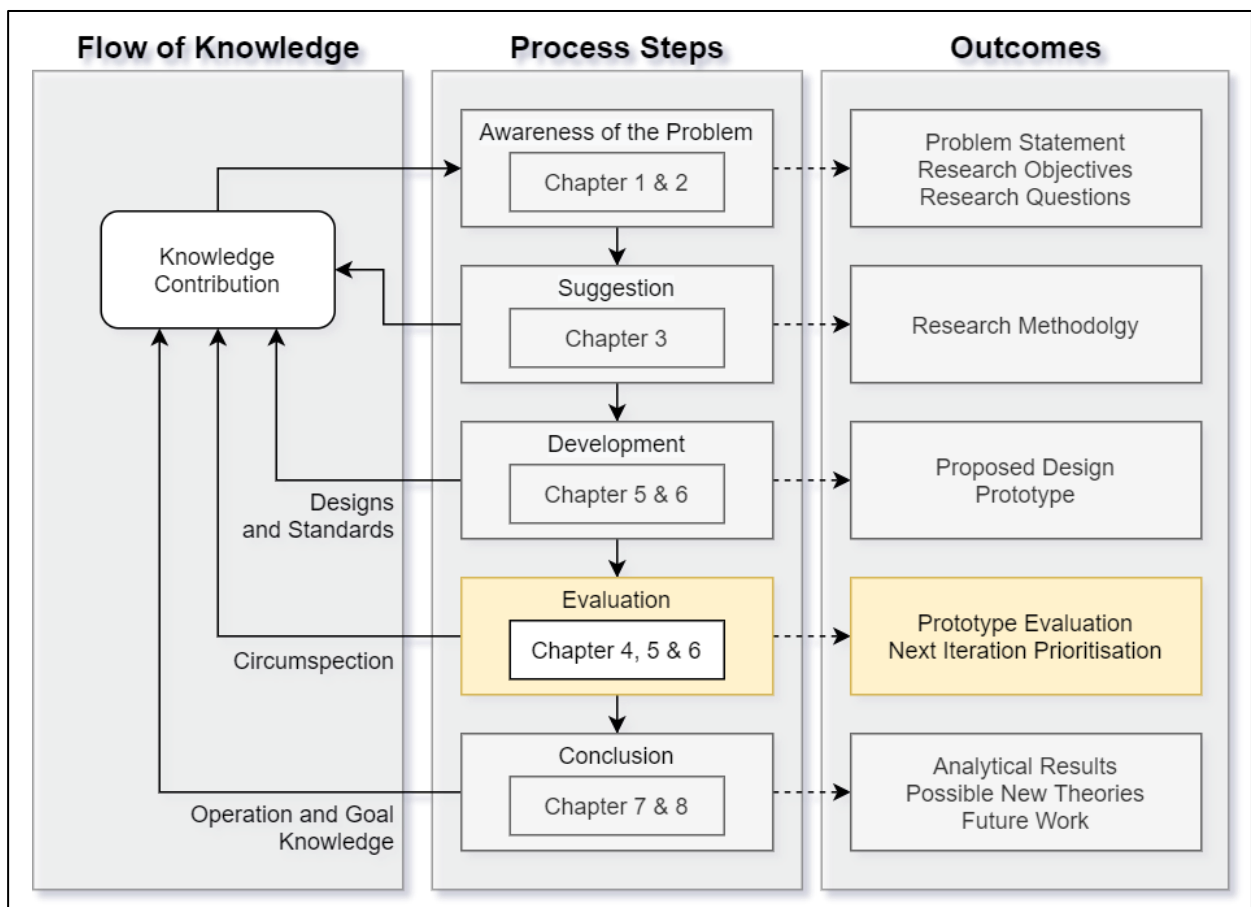


Figure 4-8: Chapter 4 DSR study layout contribution

This chapter has evaluated the factors that contribute to the time, cost and scope of a human executing the process, in line with the DSR study outlined in Figure 4-8. The next chapter introduces the initial design and build as the first iteration of TOGAF. The development of the DPA solution artefact is necessary for further data collection to take place.

CHAPTER 5: INITIAL CONCEPTUAL DESIGN AND BUILD

5.1 Introduction

The open group architecture framework (TOGAF) is an enterprise architecture framework consisting of multiple phases that provide an approach to architectural design, planning, implementation and governance. TOGAF is applied as an iterative cycle through these phases (Josey, 2016).

Blevins *et al.* (2004) propose that the TOGAF architecture development method (ADM) describes the process of migrating from the TOGAF foundation architecture to an architecture that is specific and suitable to an organisation. In a TOGAF ADM guide-through, Visual Paradigm Community Circle (2018) suggests that TOGAF ADM leverages the elements of TOGAF foundation architecture as well as other relevant architectural components. Graves (2013) explains that TOGAF, traditionally, consists of four layers, namely the business layer, data layer, application layer and technology layer. He further proposes that TOGAF ADM bundles the data and application layers into one layer known as the information systems layer, as represented in Figure 5-1.

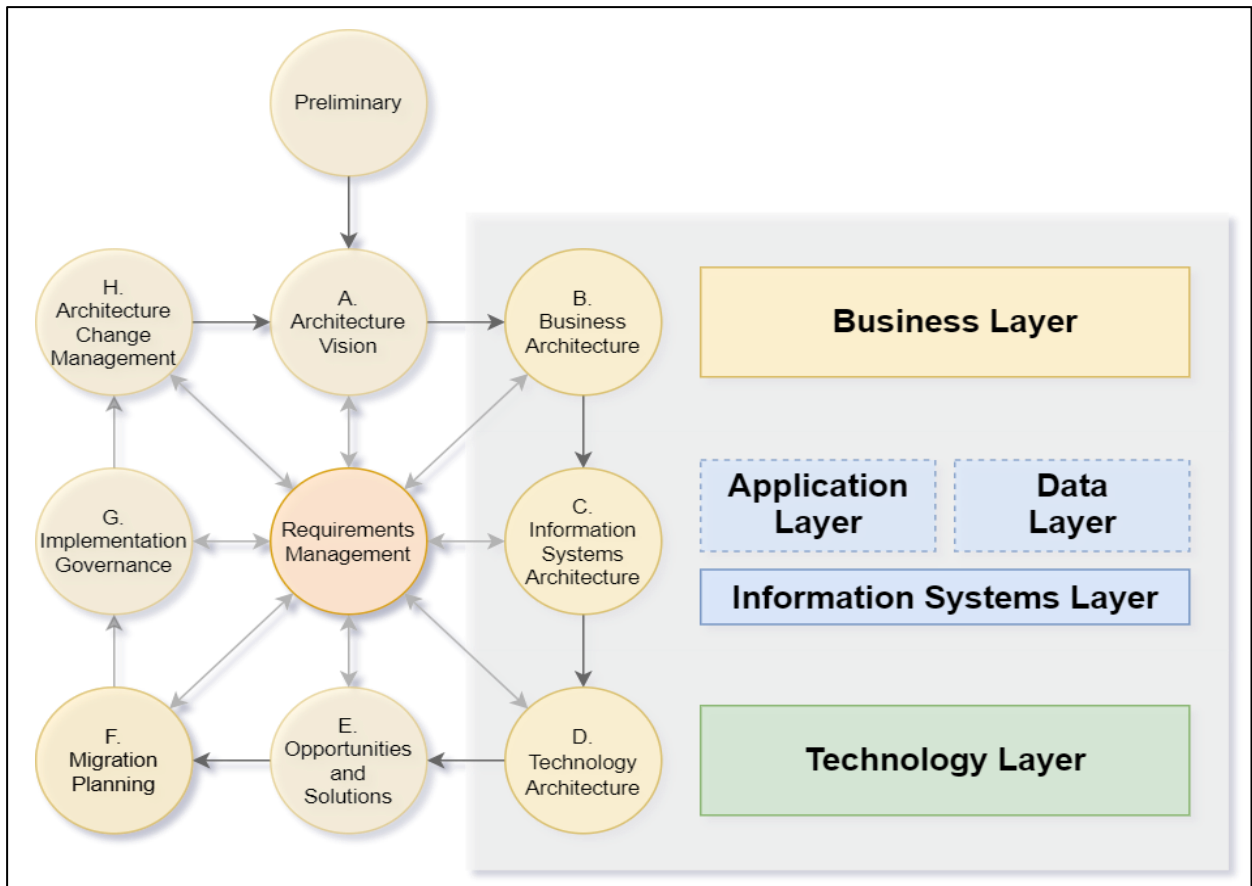


Figure 5-1: The open group architecture framework (TOGAF) architecture development method (ADM) (Graves, 2013)

The case study environment includes the adoption of the scaled agile framework (SAFe) as well as an automation lifecycle used for development. The use of TOGAF ADM is beneficial, for this study, as it will assist in encapsulating and aligning all the elements of the architecture that exist within the case study environment. The focus of this chapter will be on detailing the business architecture, the information systems architecture and the technology architecture as part of the business layer, information systems layer and the technology layer, respectively.

Sub-section 5.2 will provide a process scope overview with sub-section 5.3 expanding on the business layer with specific focus on the organisational overview, the strategy and the operating model. Sub-section 5.4 explains the information systems layer, focusing on the application and data architecture. Sub-sections 5.5 and 5.6 detail the technology layer and report prototype, respectively. Sub-section 5.7 presents the feedback from the JAD workshops, before sub-section 5.8 concludes the chapter. Sub-sections 5.3 to 5.6 contribute to the development process step in the DSR study layout, while the feedback provided in sub-section 5.7 accommodates the evaluation step.

5.2 Overview

The purpose of the artefacts used in this study serves as elements of data collection as the outputs from each of the artefacts are quantifiable and measurable according to the research questions and those outputs are believed to lay the foundation for the work towards satisfying the research objectives. Figure 5-2 defines the flow between process steps with all blue elements highlighting the steps that differ from the process that is executed by the employees. Figure 5-2 is further explained in sub-section 5.4.1.2.

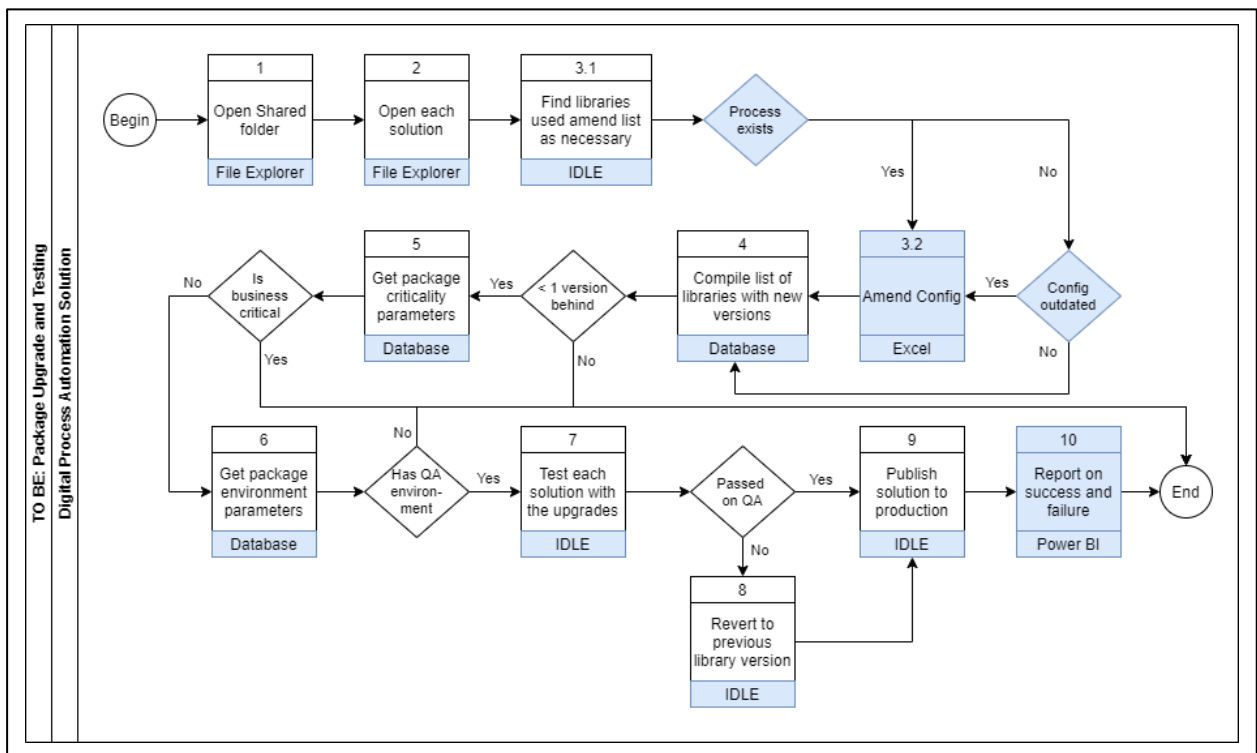


Figure 5-2: Process steps

As seen in Figure 5-2, the process is made up of predefined steps that aim to cater for the requirements defined for this project. The initial requirements of the artefact were focused on creating a solution that would execute the following steps:

- Go through all of the scripts stored in the central code repository (a set number of scripts will exist in the central code repository to be used by both execution methods. This will ensure that both execution methods are evaluated on the same number of executions)
- Gather the libraries used by each script
- Find the current version of each library in use
- Find the latest available version of each library in use
- Determine which libraries are out of date and, therefore, should be updated

- Upgrade the libraries that require an update
- Test the upgrade on processes (in QA/ Test) that meet the following criteria:
 - The current library version is less than one full version behind the latest available version
 - The processes to be tested are not considered as business critical (for example, it should not have permanent or irreversible effects on a production system)
 - The processes to be tested can be tested in a QA (or testing) environment
- Once the processes have been tested, if the outcome is successful, the libraries need to be upgraded in the production environment
- Test the upgrades in the production environment, by using the processes that were used for testing in QA/ Test
- If the upgrade in production is unsuccessful, downgrade (or rollback) the libraries to the previous version and prompt an operations system engineer to investigate further.

A DPA solution, which executes the steps listed above, should be constructed in order to collect the following data:

- The average time duration of a DPA solution completing the process in an automated fashion.
- The cost of a DPA solution completing the process in an automated fashion.
- The scope density of a DPA solution completing the process.

The DPA solution will comprise three distinct parts, namely the package information retrieval, the latest package version retrieval, and the package upgrade decision support system.

5.3 Business layer

The business architecture describes the product and/or service strategy with regard to the organisational environment. The case study environment is placed within a business unit that enables the ICT organisation to provide business services to their clients. The specific business unit, in which the process is being built, is subject to the larger organisation's strategy.

5.3.1 Organisational overview

The organisation used for this case study is a global organisation that focuses on integrating systems, managing services and infrastructure, optimising businesses while evolving with technological innovations in order to enable their client's journey into the digital age.

The vision and mission of the organisation are focused on making a difference through enabling clients to adopt technologies and practices that solve their business problems. The central key values of the organisation include continuous innovation, remaining diverse and maintaining a client-centric approach.

5.3.2 Strategy

The organisation is strategically focused on achieving efficiency, relevancy and maintaining experience:

- **Efficiency:** There is a focus on implementing efficiency through the implementation of process automation, maximising product profitability and focusing on investments that return revenue.
- **Relevancy:** In order to stay relevant in an industry that changes nearly daily, it is important to focus on staying relevant in terms of product, technology and people.
- **Experience:** Experience is obtained through client and staff engagements, working with products, working with businesses (and their problems) as well as delivering services.

5.3.3 Operating model

The operating model (as illustrated in Figure 5-3) consists of the business units that are consolidated under the organisation; clients could be serviced by multiple business, depending on the level of engagement and business problem presented by a client.



Figure 5-3: Case study environment operating model

The business support services (BSS) business unit is made up of multiple teams that focus on enabling the client-facing business units so that the business units can predominantly focus on supporting their clients. The case study environment, therefore, falls within the BSS part of the organisation. The team within the BSS, where the solution is to be implemented, is focused on automating business processes while utilising their own product lifecycle delivery model.

5.3.3.1 Product lifecycle delivery model

The architecture of the artefact that needs to be created for this study will be subject to a lean implementation of this delivery model, which was explained in more detail in Chapter 2, subsection 3.2.1.2. The lean implementation of the model, for the purpose of this study, will be focused on the *prepare automation*, *solution design*, *plan automation*, *build automation*, *test automation* and *stabilise automation* phases of the model. This is illustrated by the shaded block in Figure 5-4.

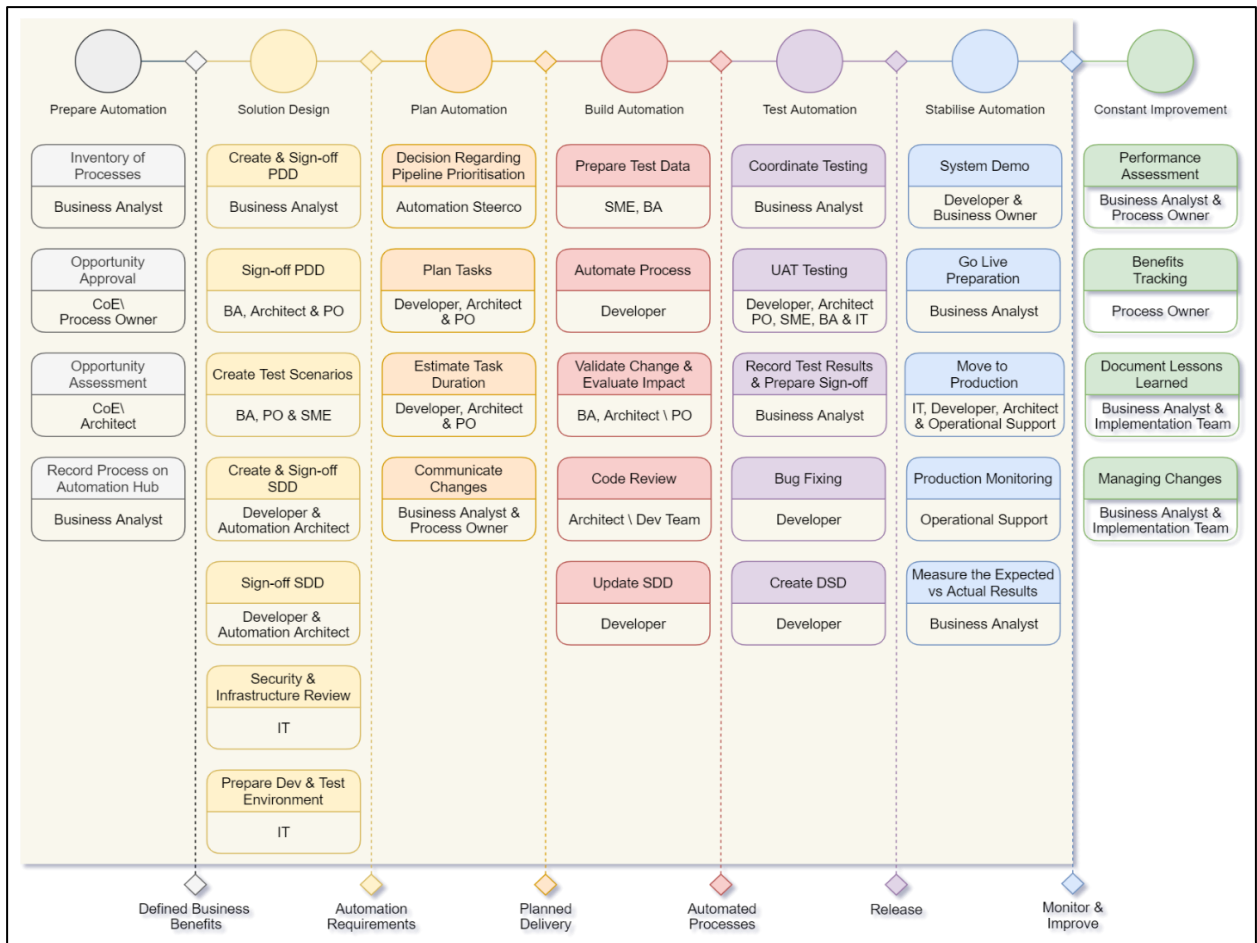


Figure 5-4: The end-to-end automation delivery model (lean study implementation)

5.3.3.2 Gap analysis

Since the case study environment utilises the SAFe methodology, GAP analysis will be used to identify and gaps between the elements of SAFe and the automation delivery model.

SAFe elements	Delivery model phases						
	Prepare	Design	Plan	Build	Test	Stabilise	Improve
Organisational agility	✓	✓	✓	✓	✓	✓	✓
Lean portfolio management	A	B	✓	✓	✓	✓	✓
Enterprise solution delivery	✓	✓	✓	✓	✓	✓	✓
Agile product delivery	A	B	✓	✓	✓	✓	✓
Team and technical ability	✓	✓	✓	✓	✓	✓	✓
Continuous learning culture	✓	✓	✓	✓	✓	✓	✓

Table 5-1: Tabular representation of the gap analysis matrix

The rows in Table 5-1 represent SAFe elements, whereas the columns represent the delivery model phases. The ticks represent which delivery model phases address the SAFe element in the corresponding row. The use of alphabetic letters indicates gaps that are elaborated upon in Table 5-2.

Letter	Gap	Description
A	Planning automation preparation	Lean portfolio management and agile product delivery are grouped together by this gap since requirements are gathered and provided by another team within the business. The requirement gathering, understanding and documentation are not planned for as part of a programme increment. Considering that subsequent tasks, like solution design are affected by the lack of planning (with regard to automation preparation), planning for requirements gathering would enable planning solution design.
B	Planning solution design	Lean portfolio management and agile product delivery are grouped together by this gap since the previous gap explains why planning solution design is challenging.

Table 5-2: Tabular representation of the gap analysis

5.4 Information systems layer

Nama and Kurniawan (2017) explain how the information systems layer is viewed from two aspects, namely the data layer and the application layer. This is supported by Graves (2013), who explains the implementation of this layer from a TOGAF ADM perspective.

5.4.1 Application architecture

Qurratuaini (2018) explains how a target application architecture is mapped to the application structures of the target applications that are involved in the project as well as to identify how they are connected.

5.4.1.1 Detailed AS IS process

Figure 5-5 illustrates the AS IS (or in place) process that is currently being executed manually and has been identified for automation.

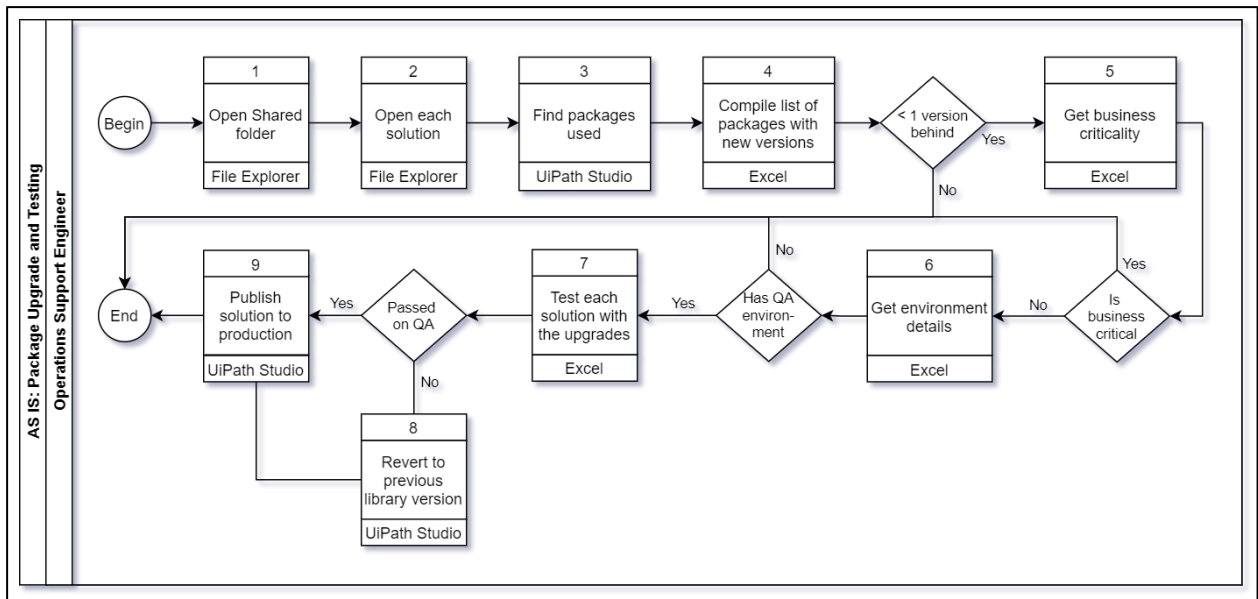


Figure 5-5: Detailed AS IS process

There is only one swim lane as the operations support engineer is the only role responsible for the execution of the end-to-end process. It is important to note that this process is specific to non-business crucial processes as business crucial processes/solutions are upgraded in a different manner as they require more intensive care.

5.4.1.2 Detailed TO BE process

The envisioned TO BE (or future state) process depicts the transition from the process that is currently in place to the desired future state process. The blue decision blocks and process steps in Figure 5-6 depict amendments to the AS IS process. All decision blocks represent the business rules that need to be catered for within the solution.

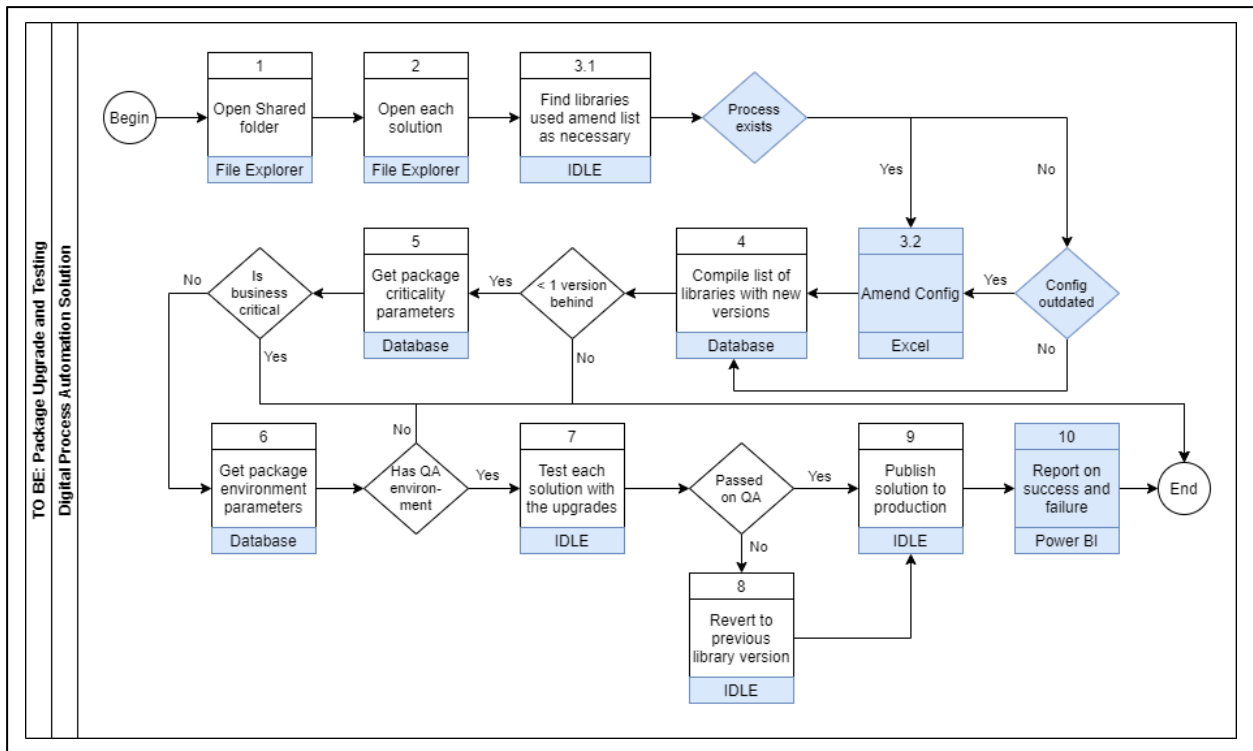


Figure 5-6: Detailed TO BE process

As seen when comparing Figure 5-5 with Figure 5-6, the whole end-to-end solution is to be automated, resulting in Figure 5-6 maintaining the requirement to have only one swim lane as the DPA solution will become solely responsible for executing this process in an automated fashion, end-to-end. IDLE refers to the application that will be used to execute the DPA solution written in Python.

5.4.1.3 Package information retrieval

The package information retrieval process will be the first workflow that makes up the DPA solution. It will go through all of the folders and subfolders that exist on a specified shared drive. These folders and subfolders are assessed as solution folders that make use of libraries and packages available through NuGet (a tool that allows developers to make use of reusable code in the form of libraries and packages).

Figure 5-7 details the steps that make up the package information retrieval process.

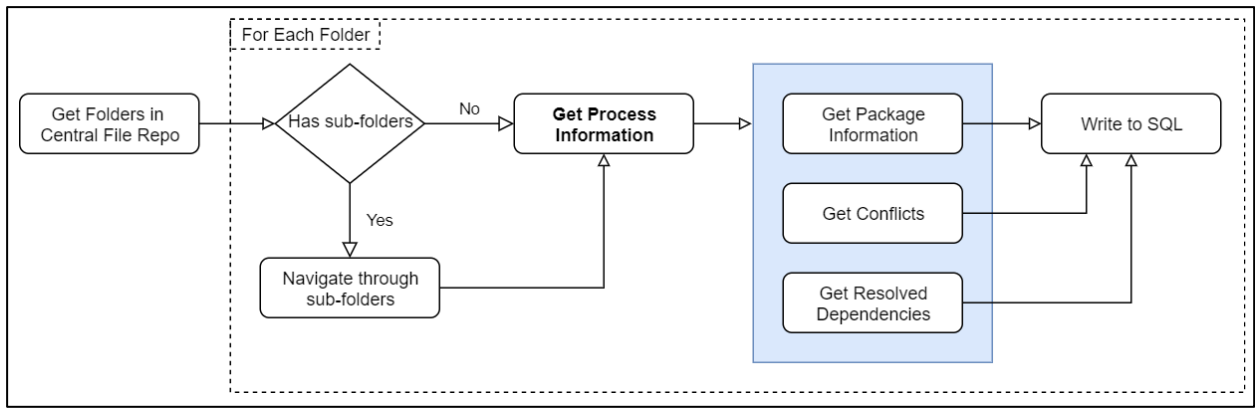


Figure 5-7: Package information retrieval process steps

The next workflow process that proceeds package information retrieval is the retrieval of latest package versions retrieval process, which is described in the subsection below.

5.4.1.4 Latest package version retrieval

The latest package version retrieval process retrieves a complete list of all packages used by any of the solutions found on the shared drive. The process then queries the latest package version and keeps a record of new versions in a SQL database.

Figure 5-8 details the steps that make up the latest package version retrieval process.

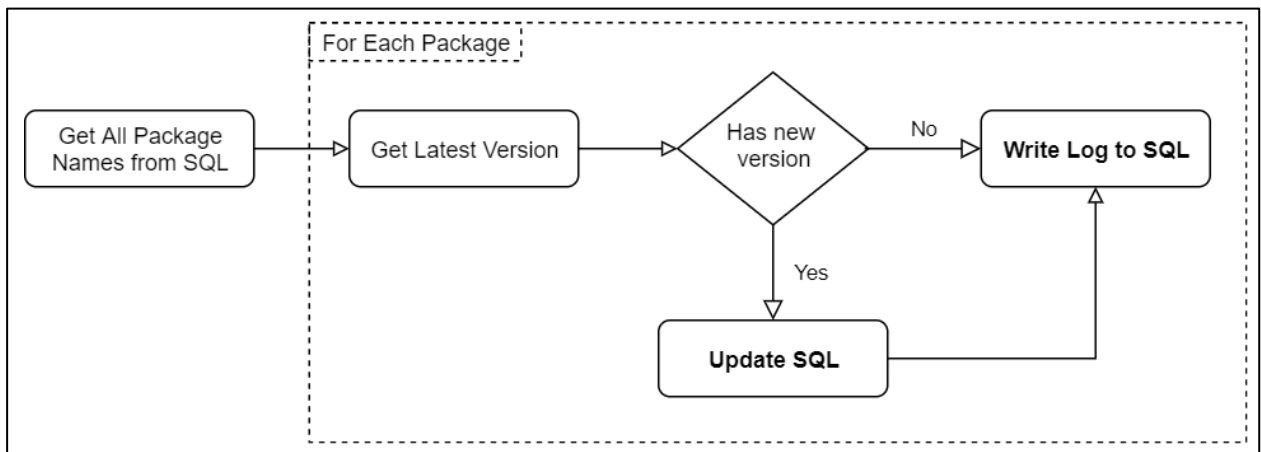


Figure 5-8: Latest package version retrieval process steps

The last workflow process that proceeds latest package version retrieval is the package upgrade DSS process, which is described in the subsection below.

5.4.1.5 Upgrade package decision support system (DSS)

The decision support system is used as the foundation for deciding which packages to upgrade, as well as which solutions to upgrade. The upgrade package DSS process retrieves a complete list of all solutions that require one or more package upgrades. These solutions are then mapped against a mapping table that contains the following information:

- Solution
- Package
- Number of versions behind
- Has QA environment
- Test succeeded on QA
- Test succeeded on production
- Datetime stamp

Only in a case where a package is less than one full version behind, is not critical to other business operations, has a QA environment and succeeds testing in the QA environment, may the package be upgraded in production. Figure 5-9 details the steps that make up the upgrade package DSS process.

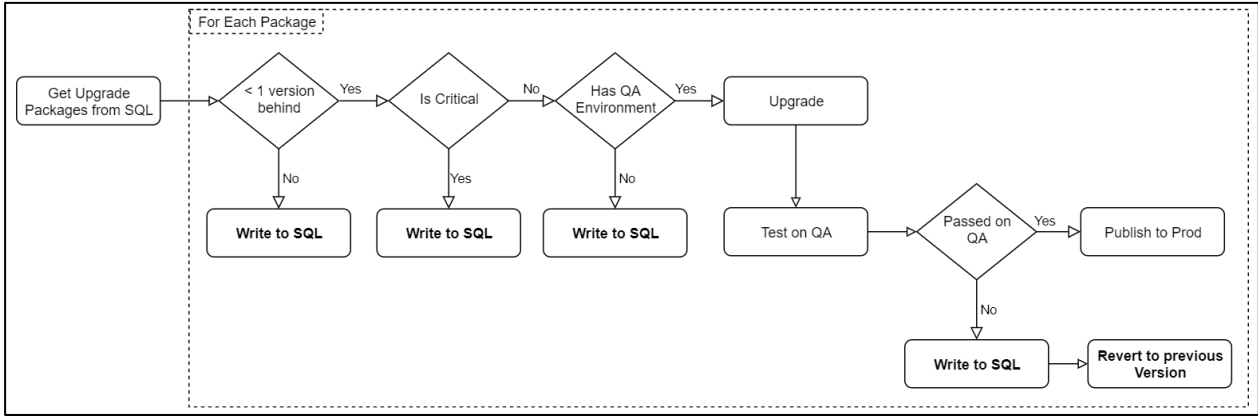


Figure 5-9: Upgrade package DSS process steps

5.4.2 Data architecture

Qurratuaini (2018) explains that the target data architecture can be mapped according to the understanding of the relationship between the current data architecture and the business architecture. The data architecture will be implemented within an existing environment that has already been developed within the case study environment.

5.4.2.1 Database design

The database design depicts the structure that the data will be stored in. Each of the entities presented in Figure 5-10 is grouped according to a logical grouping (tables) of data items (attributes). The database design, in the form of an entity relationship diagram (ERD), is outlined in Figure 5-10.

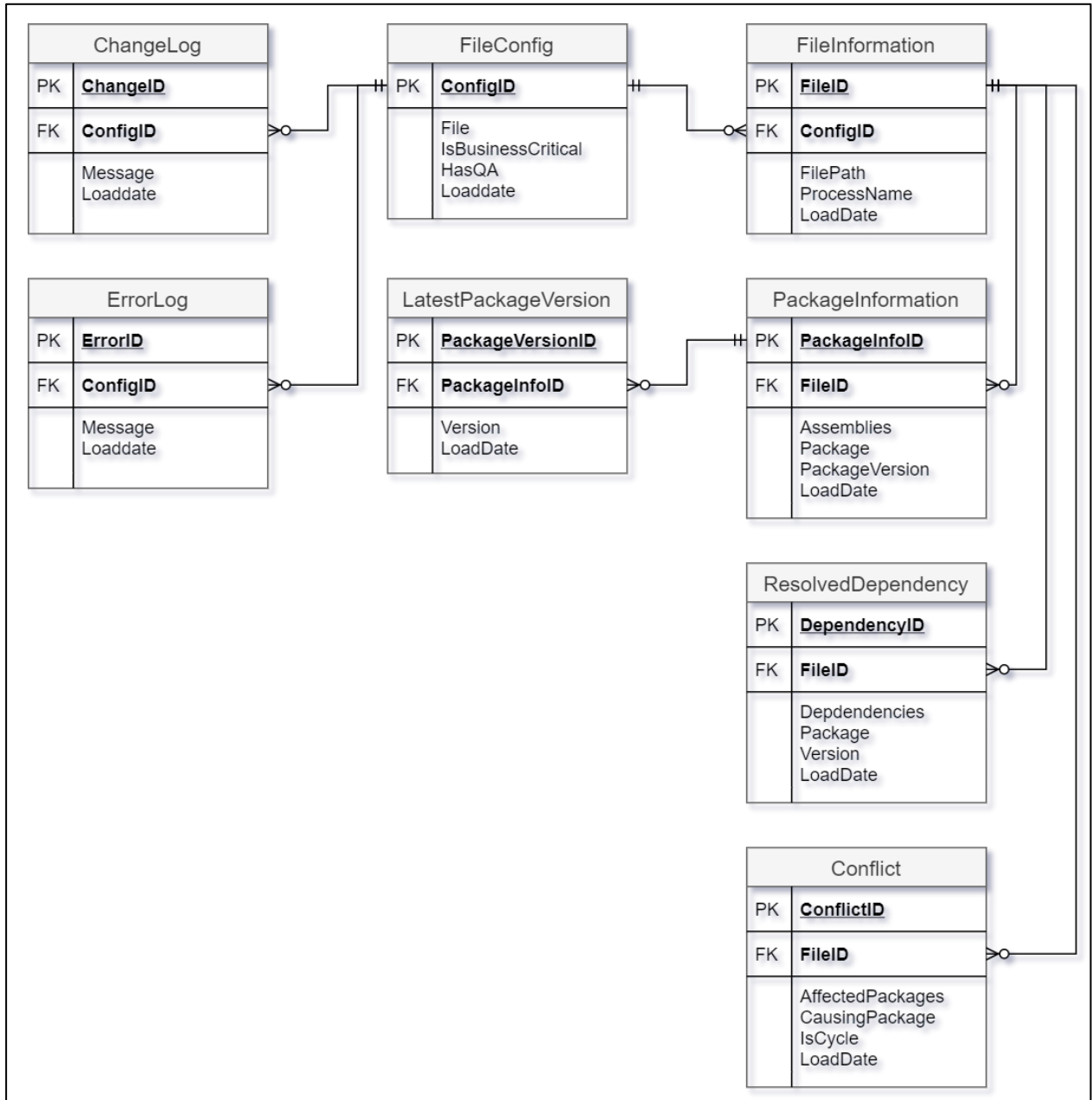


Figure 5-10: Database design

The above design accommodates for all the data that needs to be captured and allows for seamless integration with the rest of the environment. A view, named vw_PackageUpgrades will be built on top of the LatestPackageVersion, PackageInformation and FileInformation tables for reporting purposes. The advantage of using views for reporting instead of reporting directly on top of tables is to ensure that even if the raw data structure changes (for example, renaming a column), reporting does not break and is minimally affected. A new schema will be created on the database, named UiPathUpgrade, which will store the tables listed in Figure 5-10.

5.4.2.2 Data flow diagram

The process relies heavily on the database and the objects that it contains. There are multiple external agents, process steps and data stores used within the data context of the solution. Each of the abovementioned elements is outlined, in accordance with the process mapped out in Figure 5-6 through Figure 5-11.

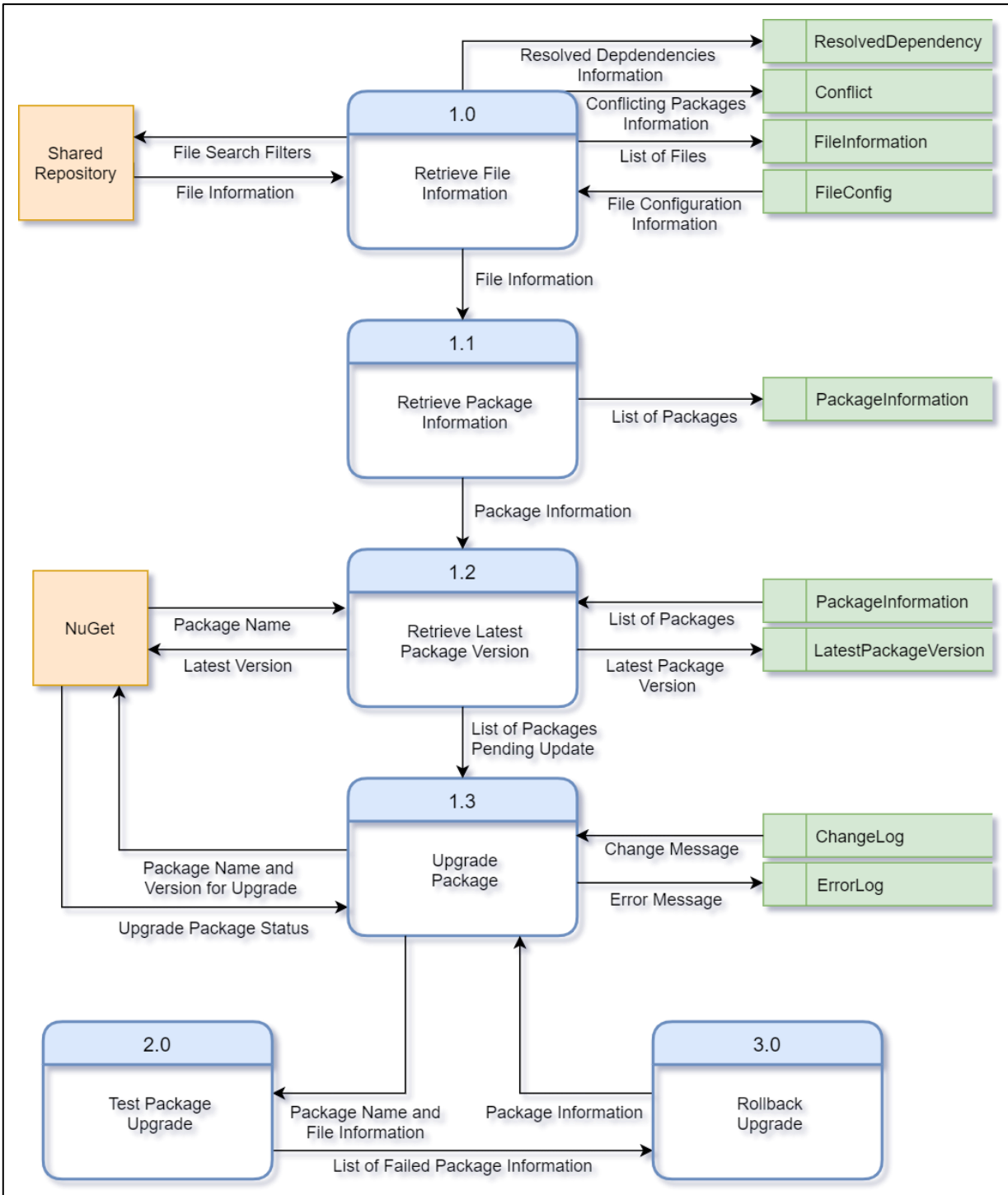


Figure 5-11: Data flow diagram

As seen in Figure 5-11, certain data stores are needed to store the data being processed. The current environment advises that the most feasible collective data store for the solution would be an SQL database hosted on the existing SQL server, which will be further explained in the next section.

5.5 Technology layer

Nama and Kurniawan (2017) explain how the technology architecture encompasses the identification of technology to be used as well as what technology already exists within the current technology architecture.

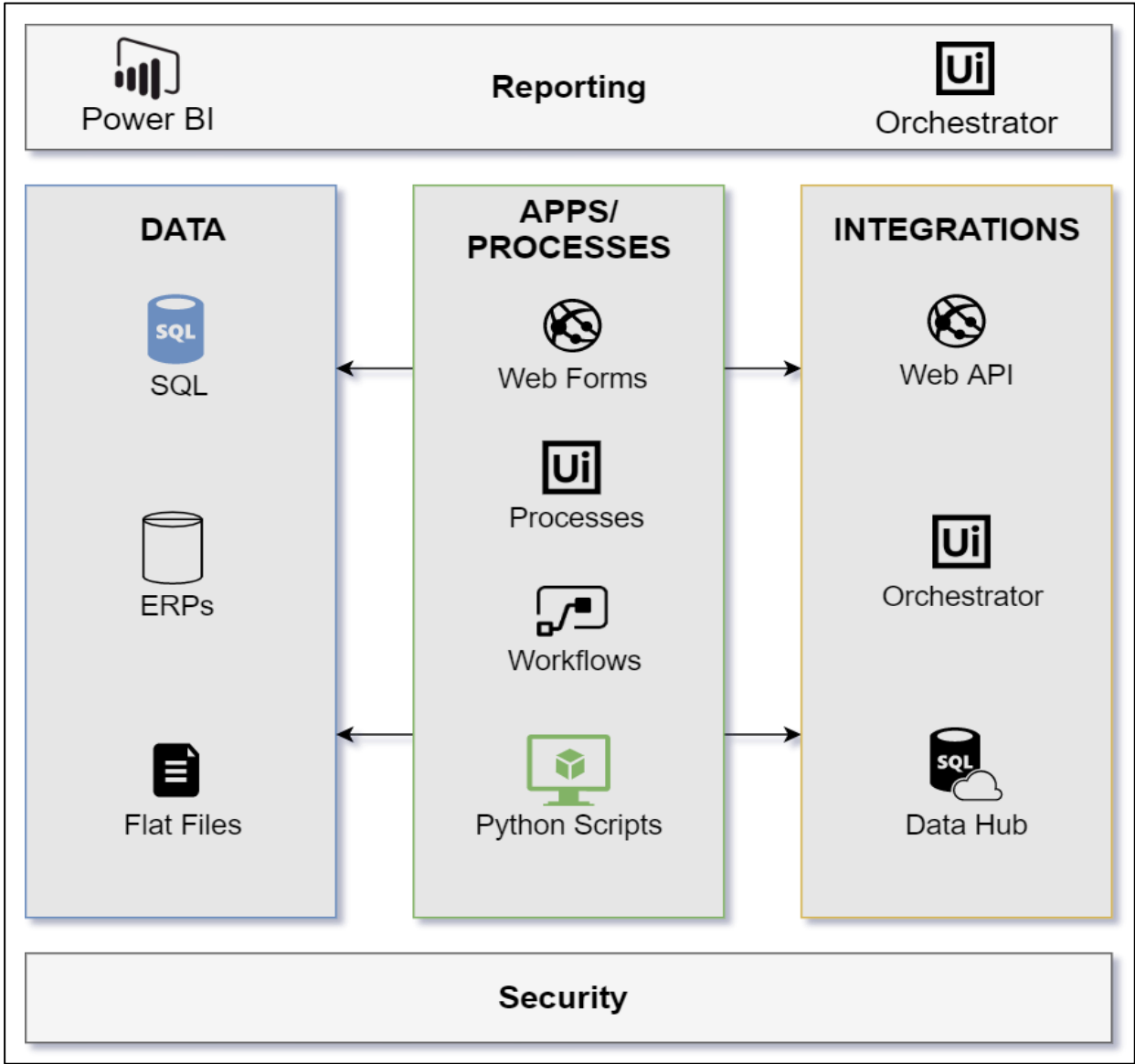


Figure 5-12: Technical landscape

Figure 5-12 illustrates the technical architecture in which all solutions within the case study environment are formed. All solutions developed in the case study environment need to adhere to a secure-by-design approach, utilising the division of solution components into the appropriate layer (security, data, process, integration and reporting). The objective of dividing components of a solution into layers is to maintain modularisation that enables faster development times of solution changes and enhancements.

5.5.1 High-level architecture

The problem at hand requires the use of a file system along with a data storage mechanism and a scripting tool/language. Therefore, the following technologies have been proposed:

- **File storage:** Shared folder or Git repository (on Azure DevOps). A shared drive and Azure DevOps repository are already in use for primary and secondary storage, respectively. As such, the framework will be created to cater for both.
- **Testing framework:** Custom developed Python testing rules engine (as a framework). The framework will be added to the existing scripting environment for scheduled runs.
- **Data storage:** An SQL server already exists within the environment. A new database will be created on the existing server to accommodate for the data storage of this solution.
- **Reporting:** A Power BI report workspace already exists within the environment. This workspace will be used when publishing the report from Power BI desktop for accessibility via the Power BI service.

The high-level architecture of the project is outlined in Figure 5-13.

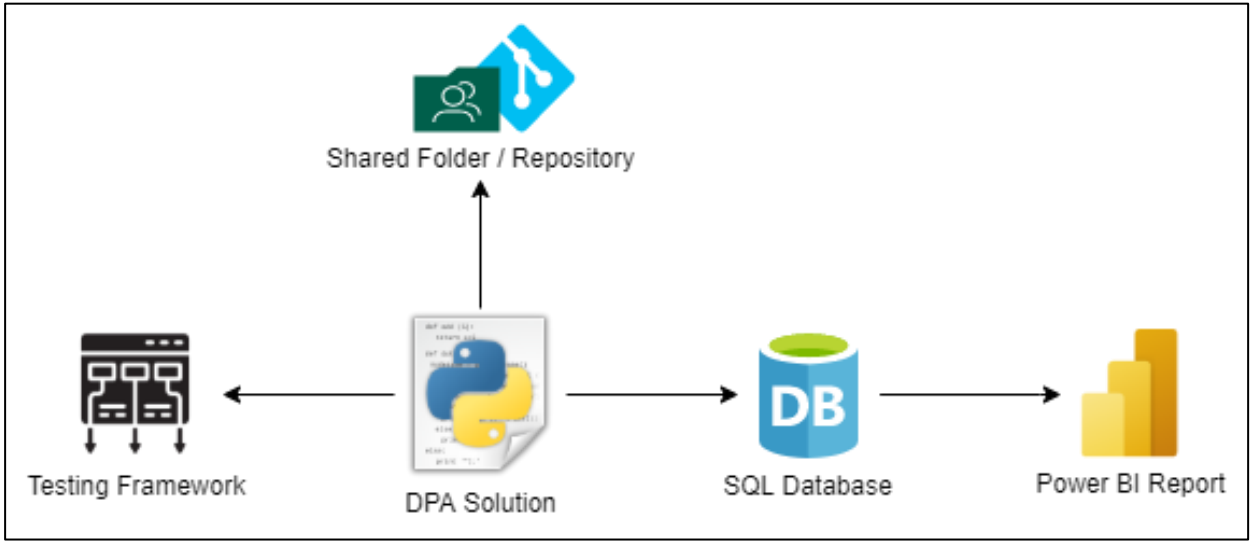


Figure 5-13: High-level architecture

As seen in Figure 5-13, the main DPA solution makes use of a shared folder/repository to centrally store all process solutions so that the process of extracting information about the solutions remains standardised. The DPA solution makes use of a testing framework (built into the DPA solution, according to the business rules outlined in Figure 5-9) to test whether or not the package upgrades were successful per solution. All data is stored in and retrieved from an SQL database, which a Power BI report is built on top of for all and any analytics that need to be done.

5.5.2 Master project runtime details

The design of the DPA solution will be subject to the technical aspects outlined in Table 5-3.

Item name	Description
Master project name	Auto package upgrade
Subprocesses	Package information retrieval Latest package version retrieval Upgrade package decision support system
Technology stack (s)	Python SQL server Power BI
Environments required	Development Testing / quality assurance (QA) Production
Trigger/schedule	Runs daily at 08:00

Table 5-3: Tabular representation of the DPA solution’s technical aspects

5.6 Report prototype

A prototype was built according to the specification outlined in the initial conceptual design, which has been elaborated upon throughout this chapter. The prototype mainly reads data from an SQL database, evaluates the retrieved data according to a set of business rules before gathering new data to be stored in the database. A two-tab power BI report provides high-level insight into the results of the automated solution through visualising the data available in the database.

Figure 5-14 illustrates the summative detail extracted from the data populated by the DPA solution into the data structure depicted by Figure 5-10, through the data flow process outlined in Figure 5-11.

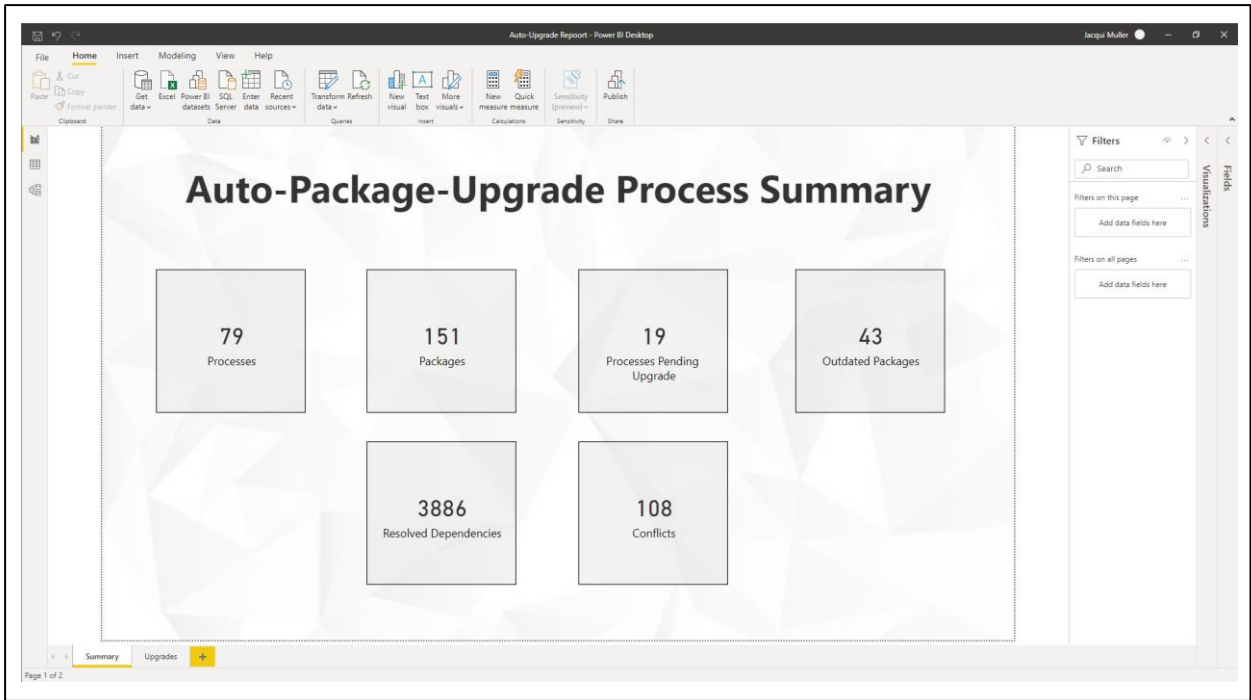


Figure 5-14: Auto-upgrade process summary tab: Power BI report

Figure 5-15 displays a lower level of detail with regard to the number of non-business-critical processes with a QA environment, the number of packages compared by major version difference and the number of pending package upgrades by process name.

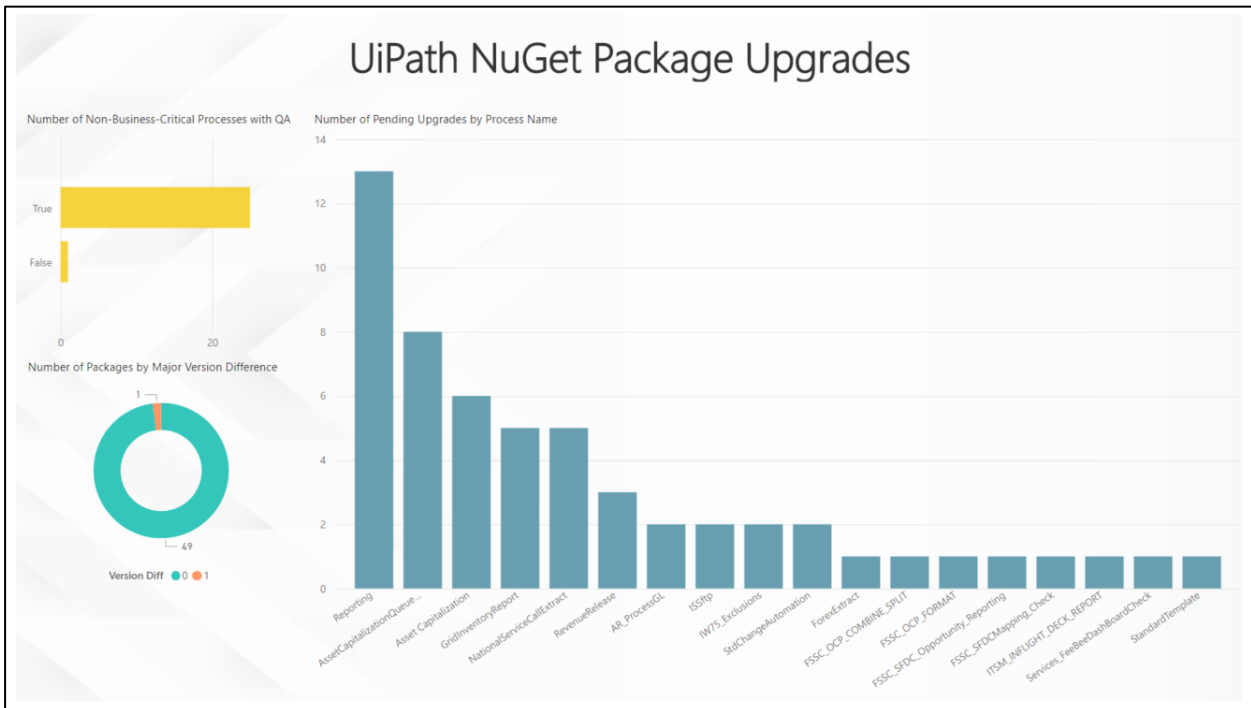


Figure 5-15: Auto-upgrade process upgrades tab: Power BI report

The graph title “Number of Non-Business-Critical Processes with QA” in Figure 5-15 illustrates the number of processes that are business-critical. The scope of the DPA solution is focused on non-business-critical processes, which means that any business-critical processes would be excluded from being processed by the DPA solution. As seen in the abovementioned graph, only one process solution is excluded due to its business-criticality characteristic.

The graph titled “Number of Packages by Major Version Difference” in Figure 5-15 illustrates the number of packages compared by the major version difference between the latest version and the current version. Traditionally, the first number in a version number represents any major changes, the second number represents any minor changes, the third number represents patches or bug fixes, and the fourth number represents a change that is less significant than a patch. Therefore, this graph takes the first number of the current and latest versions into account when working out the difference. The difference is calculated by subtracting the first number of the current version number from the first number of the latest version. Based on the data represented in the report, only one process is a major version behind, which means that it will need to be upgraded manually and not by the DPA solution. The exact process that is to be excluded due to this criterion is identified in Figure 5-16.

The graph titled “Number of Pending Upgrades by Process Name”, illustrated in Figure 5-15 and Figure 5-16, depicts all the processes that make use of libraries that need to be upgraded as well as the number of packages that are due for upgrade within each process solution.

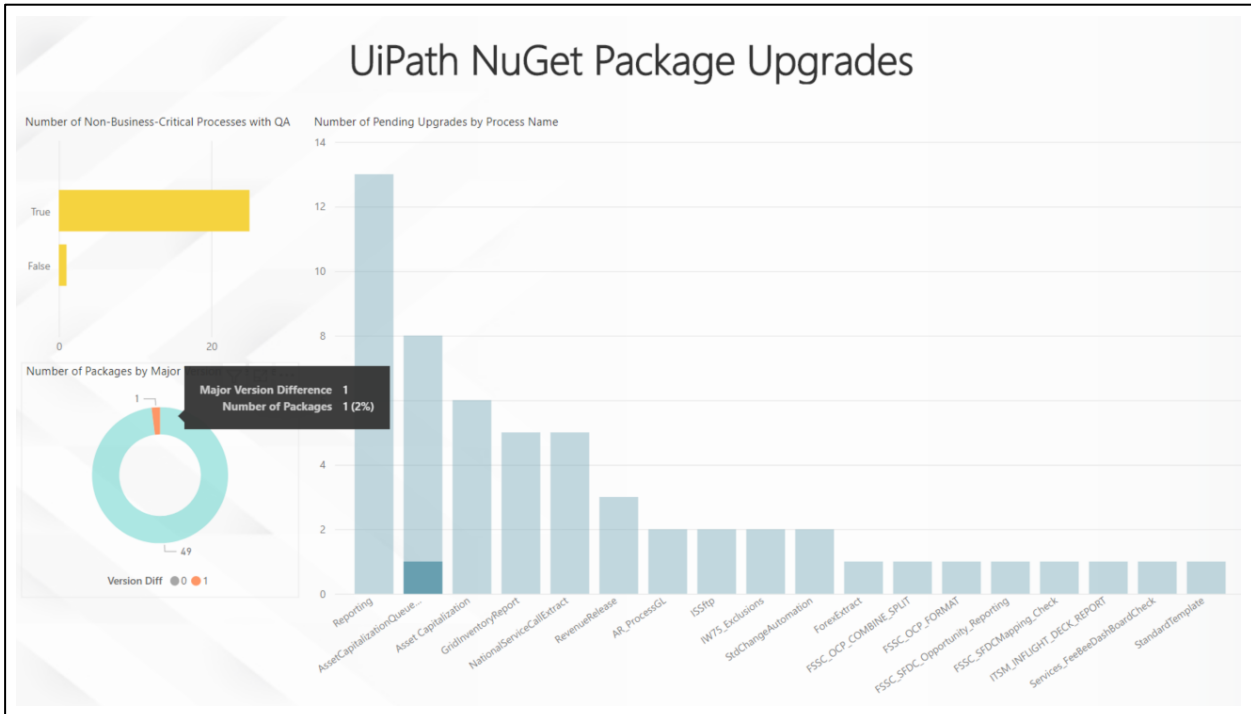


Figure 5-16: Auto-upgrade process upgrades interactive analytics

5.7 Joint application development (JAD) workshop

A system demo and retrospective JAD workshop were held to demonstrate the functionality of the solution and to gather feedback on its capability in order to improve the sustainable quality of the solution. During the system demo, it was suggested by the project stakeholders that the focus should be pivoted from UiPath projects to Python projects as UiPath had released a tool with the same functionality while the DPA solution was being developed. The system demo retrospective workshop focused on answering three main questions: what went well, what could have been done better, what wasn't done correctly? The system demo retrospective was conducted and the feedback from the retrospective was aimed at identifying any items that need to be addressed and fixed, in the next iteration, before the DPA solution execution data may be collected.

5.7.1 What went well?

Typically, this categorical question groups all successful elements of the solution. The following points were extracted from feedback on the demonstration of the prototype during the workshop:

- The business rules have been clearly defined and complexity has been removed.
- The business rules are appropriately evaluated by the DPA solution.
- The process of upgrading UiPath (.xaml) solutions has been standardised.
- The process is executed in a timely manner.
- The reports are aesthetically pleasing and are not cluttered with irrelevant visuals.

5.7.2 What could have been done better?

This categorical question groups elements that may be deemed successful as they meet the requirements but may be enhanced to provide a higher quality solution. From the output of the workshop, the following points could be improved upon:

- Include more detail on the power BI report with regard to how many upgrades passed QA, how many upgrades were published to production and how many upgrades had to be rolled back.
- Consider making this solution more dynamic to cater for different technology stacks.
- This solution is only focused on solutions stored on the shared drive. Consider building a plug-in to cater for multiple types of code repositories.

- Include more detail around the actual process steps as part of the process definition documentation (PDD).

5.7.3 What was not done correctly?

This categorical question groups elements and points that did not meet the requirements or impact the value to be gained by implementing the solution. Only one point was raised during the workshop for this question, namely: considering the emergence of Python scripts within the environment and the use of the existing UiPath testing framework capability, it would add more value to upgrade libraries used in Python projects rather than focusing on UiPath solutions.

5.8 Conclusion

Based on the need for an artefact to be developed for the data collection portion of this study, the outcome of this chapter shows the initial conceptual design. The first iteration through TOGAF has assisted in establishing the organisational environment, along with the business strategy and business model, with relation to the operational model. The operational model provides more context into the case study environment where several gaps became apparent and will be addressed through the next iterative cycle. Based on the gap analysis, incorporating the automation preparation and solution design phases of the delivery model into PI planning may assist in closing the identified gaps. The application architecture details the process that needs to be developed, and although all the steps have been defined, the next iteration may need to include detailed steps to better inform the application architecture. The technology architecture will rely on the existing automation ecosystem and will be designed and implemented to compliment what already exists.

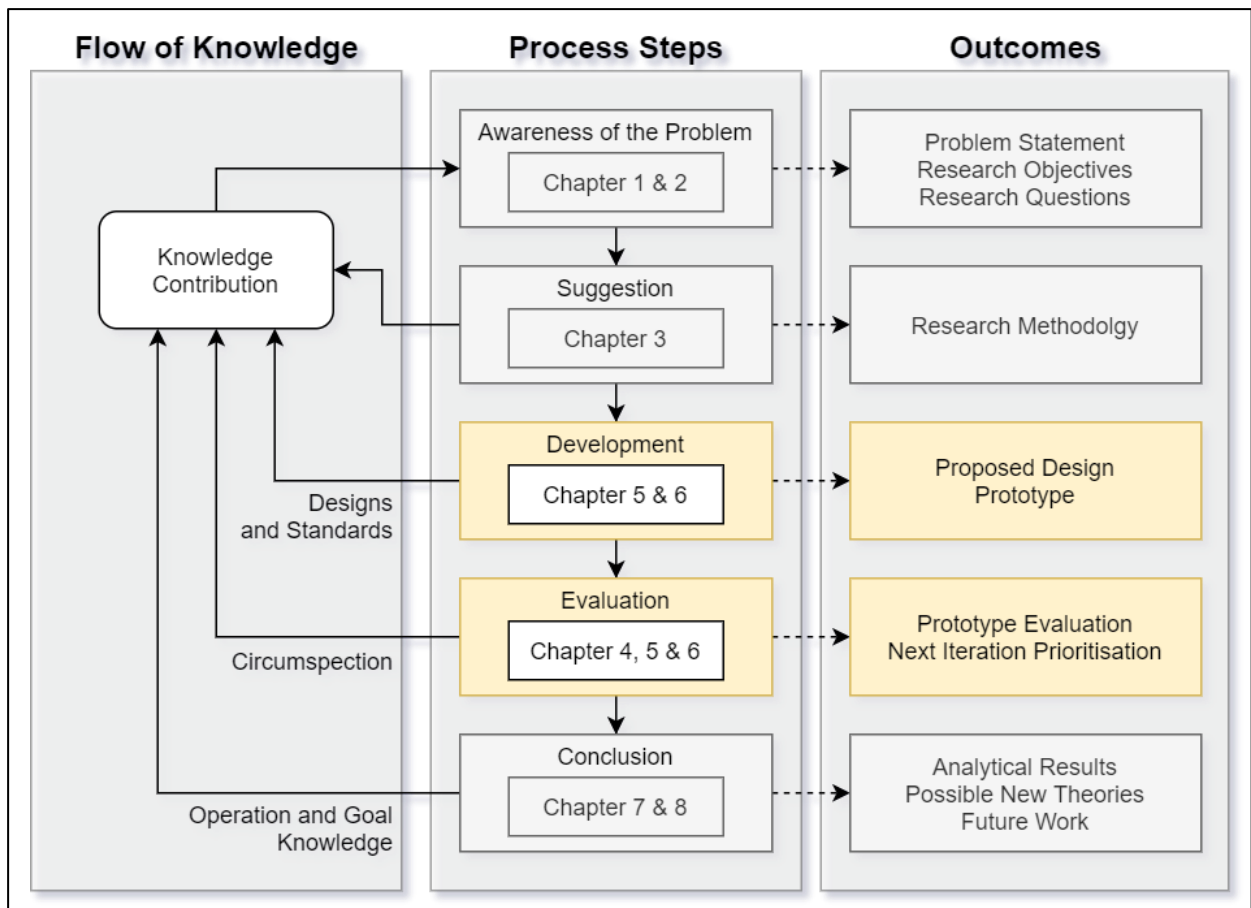


Figure 5-17: Chapter 5 DSR study layout contribution

In this chapter, the initial design and build were developed and evaluated, in line with the DSR study layout presented in Figure 5-17. The next iterative cycle will extend the work presented in this chapter to include more detail around each of the architecture layers as well as address the outcomes of the workshop.

CHAPTER 6: PROPOSED CONCEPTUAL DESIGN AND BUILD

6.1 Introduction

The first iteration through TOGAF rendered the initial conceptual design, as presented in Chapter 5. Since the initial conceptual design has been formed, the organisation faced a restructure, which has had a significant impact on their operating model, resulting in a change to the business architecture. The application architecture in the initial conceptual design also requires more detail regarding the process steps as discovered during the joint application development (JAD) workshop.

The abovementioned workshop rendered the items listed in Table 6-1. Items 1, 4 and 5 from Table 6-1 will be addressed in the current iterative conceptual design and build phase, through all the impacted architecture layers.

Item	Proposed change	Requires focus in current iteration
1	Include more detail on the power BI report with regard to how many upgrades passed QA, how many upgrades were published to production and how many upgrades had to be rolled back.	Yes
2	Consider making this solution more dynamic to cater for different technology stacks	No
3	This solution is only focused on solutions stored on the shared drive. Consider building a plug-in to cater for multiple types of code repositories.	No
4	Include more detail around the actual process steps as part of the process definition documentation (PDD)	Yes
5	Consider the emergence of Python scripts within the environment and the use of the existing UiPath testing framework capability; it would add more value to upgrade libraries used in Python projects rather than focusing on UiPath solutions.	Yes

Table 6-1: Tabular representation of the changes to the initial conceptual design and build proposed

The focus of this chapter will be on implementing the changes indicated in Table 6-1 within the next iterative cycle of TOGAF in terms of the business architecture, the information systems

architecture and the technology architecture as part of the business layer, information systems layer and the technology layer, respectively.

Sub-section 6.2 explains the changes to the business layer with specific reference to factors that contribute to the operating model. Sub-section 6.3 details the application architecture and the data architecture that make up the information systems layer, while sub-sections 6.4 and 6.5 expand on the technology layer and the report prototype, respectively. Sub-section 6.6 presents the feedback from the JAD workshops, with sub-section 6.7 focusing on the data collected from the DPA solution execution before sub-section 6.8 concludes the chapter. Sub-sections 6.2 to 6.5 contribute to the development process step in the DSR study layout, while the feedback provided and data collected in sub-sections 6.6 and 6.7 accommodate the evaluation step.

6.2 Business layer

The business architecture describes the product and/or service strategy with regard to the organisational environment. The case study environment is placed within the business support services function of the organisation's operating model.

The adoption of the new operating model has seen an extension of the strategic objectives towards a go-to-market (GTM) strategy that aligns with the organisation's main focus on their client-centric approach to business.

6.2.1 Operating model

The operating model illustrated in Figure 6-1 shows the GTMs (which previously were the business units) that make up the organisation's strategy. The clients are managed according to their geography through various client management teams, domain specialists and sales specialists. Service delivery is managed by product owners, product specialists, GTM services and client services teams.



Figure 6-1: Case study environment operating model

Each of the GTMs are focused on service delivery to their clients. The encapsulated landscape of the GTMs is still supported by the business support services (BSS). The case study environment still falls within the BSS function that focuses on business enablement through digitisation. The client-facing portion of the business is affected more than the BSS environment, which provides more stability to the case study environment, with the progress made during the previous iteration not needing to be written off due to structure change.

6.2.1.1 Gap analysis

The product lifecycle delivery model has remained the same since the first iteration. An amendment was made to the SAFe implementation of the product lifecycle delivery model, which accommodates the planning of automation preparation and solution design during PI planning. The outcome of the current gap analysis is detailed in Table 6-2.

SAFe elements	Delivery model phases						
	Prepare	Design	Plan	Build	Test	Stabilise	Improve
Organisational agility	✓	✓	✓	✓	✓	✓	✓
Lean portfolio management	✓	✓	✓	✓	✓	✓	✓
Enterprise solution delivery	✓	✓	✓	✓	✓	✓	✓
Agile product delivery	✓	✓	✓	✓	✓	✓	✓
Team and technical ability	✓	✓	✓	✓	✓	✓	✓
Continuous learning culture	✓	✓	✓	✓	✓	✓	✓

Table 6-2: Tabular representation of the gap analysis matrix

The rows in Table 6-2 represent SAFe elements, whereas the columns represent the delivery model phases. The ticks represent which delivery model phases address the SAFe element in the corresponding row. Since there are no alphabetic characters present in the table, it can be seen that there are no new gaps that need to be addressed.

Table 5-1 indicates that two gaps (A and B) were identified during the initial conceptual design iterative cycle. Gap A is identified as the lack of planning of automation preparation. Gap B is identified as the lack of planning solution design. During this iterative cycle, both gaps were addressed by adding automation preparation and solution design to PI planning, which is why Table 6-2 indicates that there are no more gaps.

6.3 Information systems layer

The information systems layer, made up of the application and data architectures, needs to be adapted to incorporate the two selected changes to the initial conceptual design, as proposed, namely:

- Including more detail in the Power BI report with regard to how many upgrades passed quality assurance (QA), how many upgrades were published to production and how many upgrades had to be rolled back.
- Including more detail around the actual process steps as part of the process definition documentation (PDD).
- Considering the emergence of Python scripts within the environment and the use of the existing UiPath testing framework capability, it would add more value to pivot towards creating a framework to test the upgrading of pip packages.

6.3.1 Application architecture

The abovementioned changes lead to the addition of detailed AS IS process steps, input data description for each process step, a detailed TO BE process flow and exception handling specification. The detailed AS IS and TO BE process steps remain the same even though the scope of the DPA solution is to be pivoted from focusing on UiPath processes to Python scripts.

6.3.1.1 Detailed AS IS process steps

Figure 6-2 illustrates the AS IS (or in place) process that is currently being executed manually and has been identified for automation.

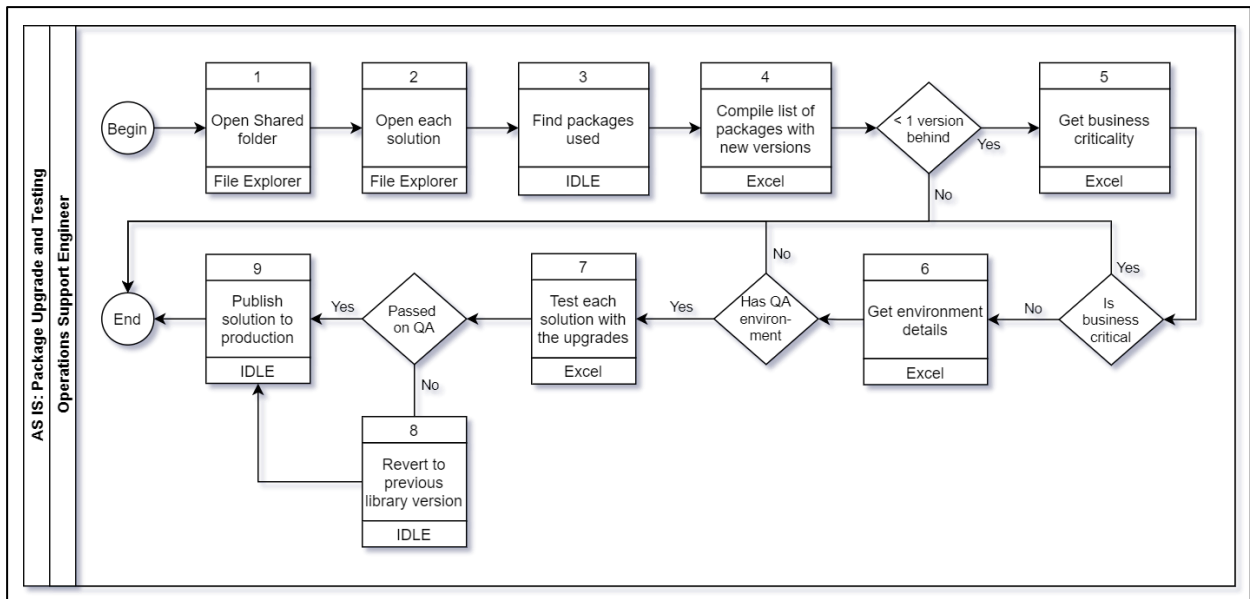


Figure 6-2: Detailed AS IS process

Figure 6-2 contains the steps that need to be executed in order for the process to work as expected. Table 6-3 elaborates on each step, providing more information around the input, description, exception handling and possible actions of each step:

# Step	Input	Description	Exceptions handling	Possible actions
1	Open shared folder	A shared folder is used to store all of the solutions that need to be accessed as the first step of the process. Sub-folders also need to be catered for as some solution folders may contain different solutions specific to different versions and environments. This step is done by going through the files through File Explorer.	1. The shared folder is inaccessible 2. Sub-folders may be in use by other processes	1. Retry later 2. Retry later
2	Open each solution	Each solution found in the shared folder (and subfolders)	1. Sub-folders may be in	1. Retry later

		<p>needs to be opened in order to find the package names.</p> <p>This step is done by going through the files through File Explorer.</p>	use by other processes	
3	Find packages used amend list as necessary	<p>A list of libraries used in each solution is compiled and amended as necessary. As soon as a new library is found, it is added to the list. As soon as a new solution is found, it should be added to the list along with the libraries used in the solution and the current version.</p> <p>This step is done by going through the solutions in the relevant IDE (either notepad++ or PyCharm). The list is to be maintained in an Excel document</p>	N/A	N/A
4	Compile list of packages with new versions	<p>The distinct list of libraries should be used in order to get the list of packages</p> <p>This step is executed by running a command in the command prompt. The list is to be recorded in an Excel document.</p>	N/A	N/A

5	Get package criticality parameters	<p>The Excel sheet is filtered to find each solution. The corresponding value from the field marked "Is Business Critical" is extracted.</p> <p>This step is executed using Excel.</p>	N/A	N/A
6	Get package environment parameters	<p>The Excel sheet is filtered to find each solution and the applicable parameters</p> <p>This step is executed through Excel.</p>	N/A	N/A
7	Test each solution with the upgrades	<p>Once the solutions feasible for testing have been identified, these solutions are used to test the success of the upgrade on the QA environment.</p> <p>The execution of the solution is done through IDLE (or any relevant IDE).</p>	N/A	N/A
8	Revert to previous library version	<p>In a case where a test on QA fails, the upgrades need to be reverted (on QA).</p> <p>This is done through the IDLE (or any relevant IDE).</p>	1. Failed to revert	1. Retry later
9	Publish solution to production	Successful QA upgrades should be replicated on the production environment.	1. Failed to publish	1. Raise alert for manual oversight

		This is done through IDLE (or any relevant IDE)		in next attempt
10	Report on success and failure	A report showing the number of successes and failures should be refreshed with the latest results. The report is a power BI report and can be viewed in Power BI Desktop or on the online Power BI service.	N/A	N/A

Table 6-3: Tabular representation of the detailed AS IS process steps

6.3.1.2 Input data description

Each of the steps detailed in Table 6-3 require some form of input data. Table 6-4 contains information regarding what form the input data should be provided in:

# Step	Input type	Are inputs * <i>Native Digital?</i>	Are the Inputs * <i>Structured Data?</i>
1	File Explorer	Yes	Yes
2	File Explorer	Yes	Yes
3	Excel/ Notepad ++	Yes	Yes
4	Excel	Yes	Yes
5	Excel	Yes	Yes
6	Excel	Yes	Yes
7	IDLE	Yes	Yes
8	IDLE	Yes	Yes
9	IDLE	Yes	Yes
10	Power BI	Yes	Yes

Table 6-4: Tabular representation of the AS IS process input data description

* **Native digital:** This is data that was originally created digitally, e.g. Excel, database or application reports etc. The non-native digital inputs are usually scanned images.

* **Structured data** has a predictable format and exists in fixed fields (e.g. an Excel cell or a field in a form) and is easily detectable via search algorithms.

6.3.1.3 Detailed TO BE process

The envisioned TO BE (or future state) process has changed slightly from the first iterative cycle, in terms of scope. The output of the reporting component (marked in yellow in Figure 6-3) could be improved upon by using an existing telemetry module. The telemetry module captures granular detail of every step of the process and stores it in a text file that may then be extracted for reporting purposes. The telemetry logs could be used in the future for heuristics mining, as mentioned in Chapter 2.

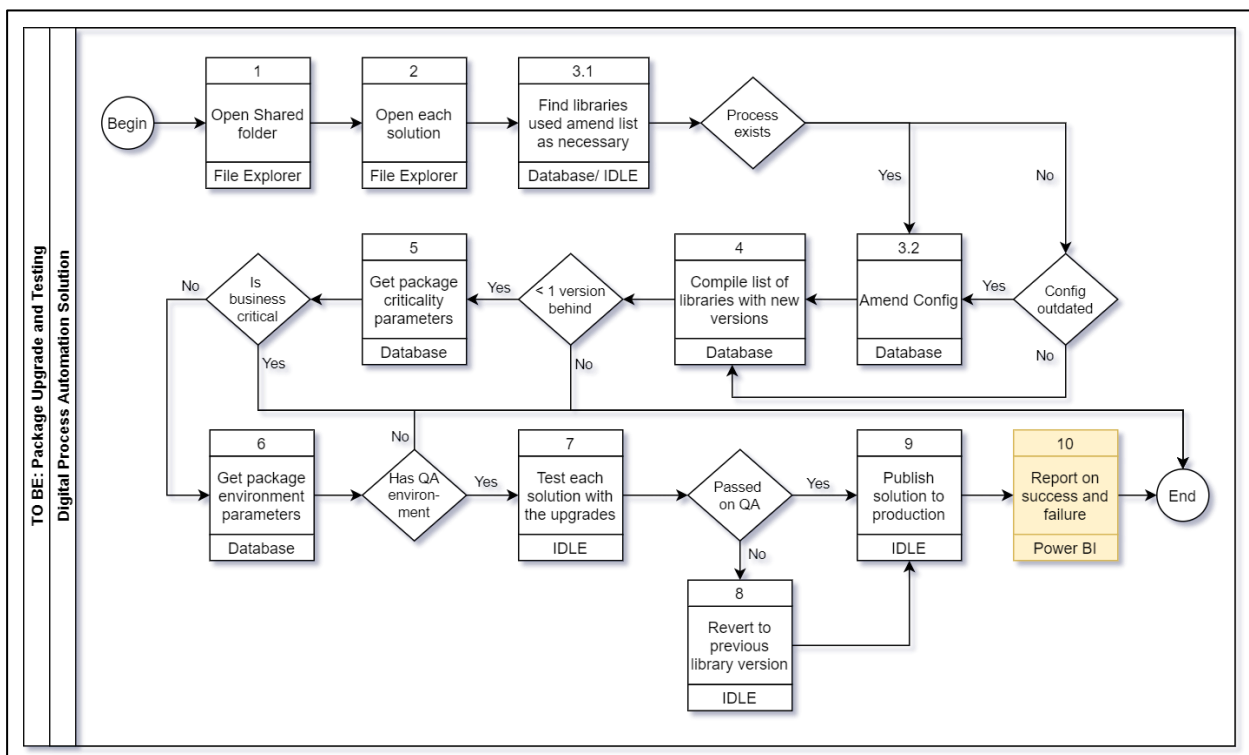


Figure 6-3: Detailed TO BE process

The detailed TO BE process structure remains the same with the greatest difference being the shift from upgrading NuGet packages (for UiPath processes) to upgrading pip libraries (for Python scripts).

6.3.1.4 Package information retrieval

The package information retrieval process will be the first workflow that makes up the DPA solution. It will go through all of the folders and subfolders that exist on a specified shared drive. These folders and subfolders are assessed as solution folders that make use of libraries and packages available through pip (a tool that allows Python developers to make use of reusable code in the form of libraries and packages). Figure 6-4 details the steps that make up the package information retrieval process.

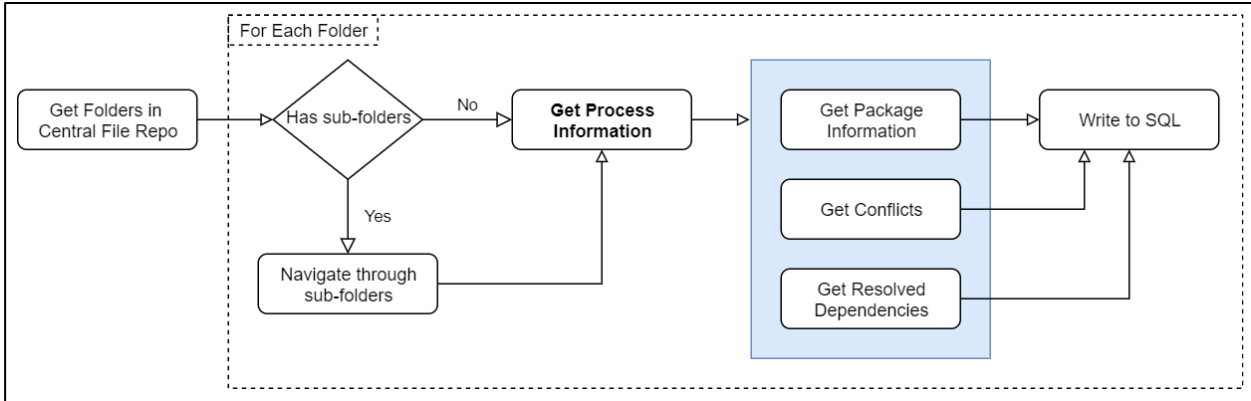


Figure 6-4: Package information retrieval process steps

The next workflow process that proceeds package information retrieval is the retrieval of latest package versions retrieval process, which is described in the subsection below.

6.3.1.5 Latest package version retrieval

The latest package version retrieval process retrieves a complete list of all packages used by any of the solutions found on the shared drive. The process then queries the latest package version and keeps a record of new versions in an SQL database. Figure 6-5 details the steps that make up the latest package version retrieval process.

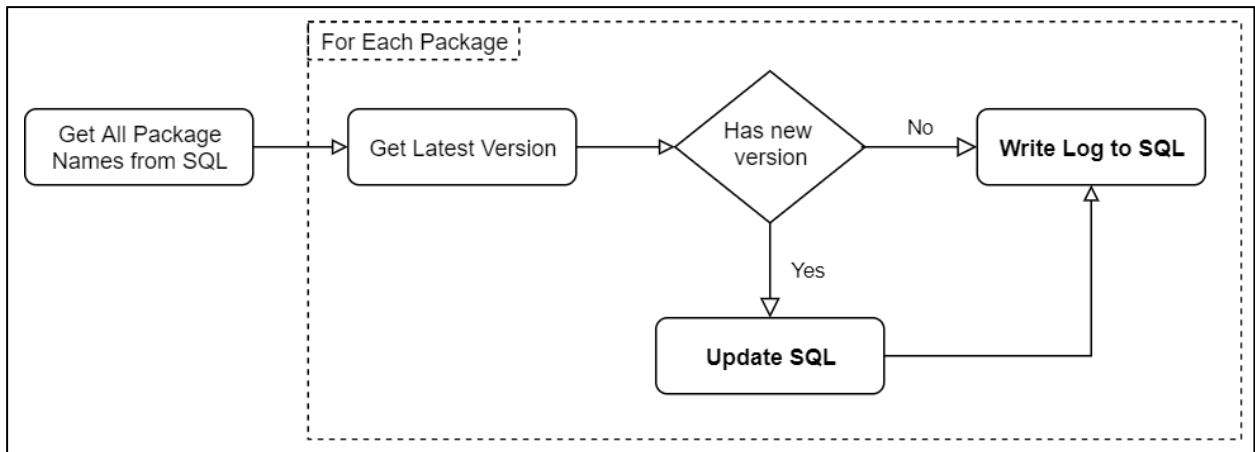


Figure 6-5: Latest package version retrieval process steps

The last workflow process that proceeds latest package version retrieval is the package upgrade decision support system (DSS) process, which is described in the subsection below.

6.3.1.6 Upgrade package decision support system (DSS)

This is the portion of the solution where the decision support system is used as the foundation for deciding which packages to upgrade, as well as which solutions to upgrade. The upgrade package DSS process retrieves a complete list of all solutions that require one or more package upgrades. These solutions are then mapped against a mapping table that contains the following information:

- Solution
- Package
- Number of versions behind
- Has QA environment
- Test succeeded on QA
- Test succeeded on Production
- Datetime stamp

Only in a case where a package is less than one full version behind, is not critical to other business operations, has a QA environment and succeeds testing in the QA environment, may the package be upgraded in production. Figure 6-6 details the steps that make up the upgrade package DSS process.

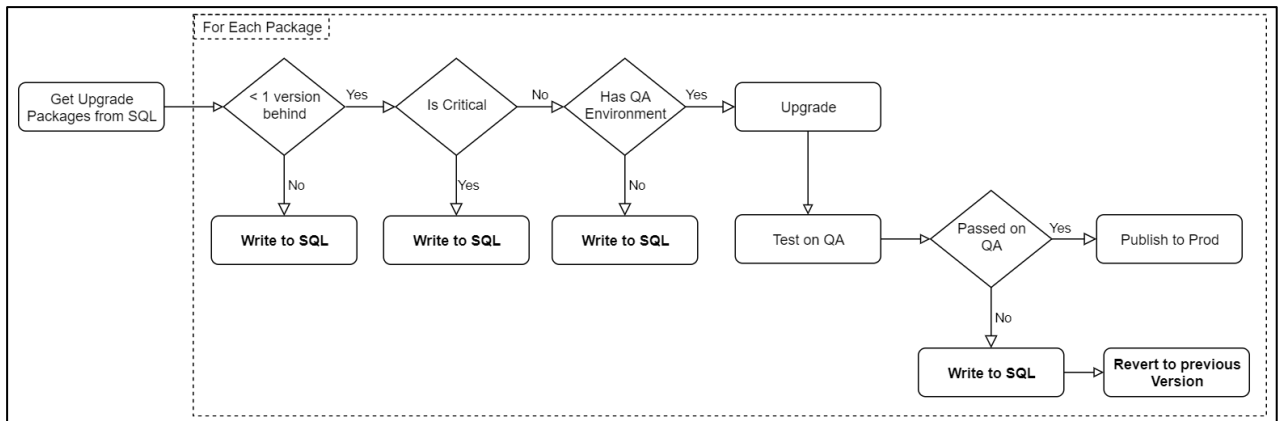


Figure 6-6: Upgrade package DSS process steps

6.3.1.7 Exception handling

Exception handling is an important part of any software development as any errors that occur from the code would need an appropriate response, even if the response is in the form of sequential actions being performed. In the case of automation, there are two types of exceptions that need to be addressed:

- **Known exceptions:** Previously encountered. A scenario is defined with clear actions and workarounds for each case. Both exceptions listed in Table 6-5 are known exceptions.
- **Unknown exceptions:** New situation that was not encountered before. It cannot be predicted and in case it happens it needs to be flagged and communicated to an authorised person for evaluation.

For all the other unanticipated or unknown business (process) exceptions, the robot should:

- Send an email notification to the operations team with the error message screenshot attached.

Exception type	Exception name	Action to be taken
Known	File not accessible	Rerun the process.
Known	Use of deprecated method	Notify the developer

Table 6-5: Tabular representation of the known and unknown exceptions

6.3.2 Data architecture

The pivot in scope from UiPath processes to Python scripts impacts the database design and data flow diagram as a different data structure is needed to store the necessary information. The additional reporting components also require new views to be created, which will be represented by the database design.

6.3.2.1 Database design

The database design depicts the structure that the data will be stored in. Each of the entities in Figure 6-7 is grouped according to a logically grouping (tables) of data items (attributes). The database design, in the form of an entity relationship diagram (ERD) is outlined in Figure 6-7.

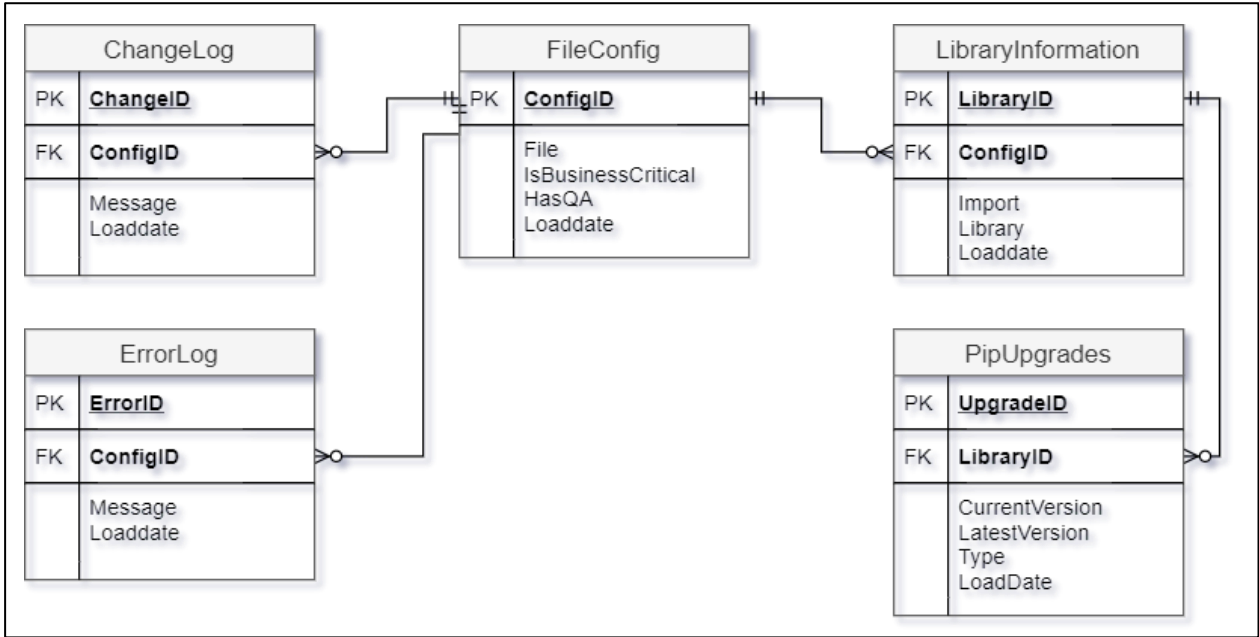


Figure 6-7: Database design

The design presented in Figure 6-7 accommodates for all the data that needs to be captured and allows for seamless integration with the rest of the environment. Three views need to be created on top of the database design:

- **vw_PendingUpgrades:** All processes that utilise a library that meets the automated upgrade criteria are retrieved through this view, along with libraries that require an upgrade. This view is compiled by joining multiple tables:
 - o PipUpgrades
 - o LibraryInformation

- FileConfig
- **vw_SuccessfulUpgrades:** All processes for which all libraries that require an upgrade have been successfully upgraded are retrieved through this view, along with the number of libraries that were upgraded.
 - vw_PendingUpgrades
 - ChangeLog
- **vw_UpgradeRollbacks:** All processes for which one or more libraries that require an upgrade had to be rolled back (due to unsuccessful upgrade) are retrieved through this view, along with the number of libraries that were upgraded and the error message.
 - vw_PendingUpgrades
 - ErrorLog

The advantage of using views for reporting instead of reporting directly on top of tables is to ensure that even if the raw data structure changes (for example, renaming a column), reporting does not break and is minimally affected. A new schema will be created on the DPAQA database, named AutoUpgrade, which will store the tables listed in Figure 6-7.

6.3.2.2 Data flow diagram

The process relies heavily on the database and the objects that it contains. There are multiple external agents, process steps and data stores used within the data context of the solution. Each of the abovementioned elements is outlined, in accordance with the process mapped out in Figure 6-3, in Figure 6-8.

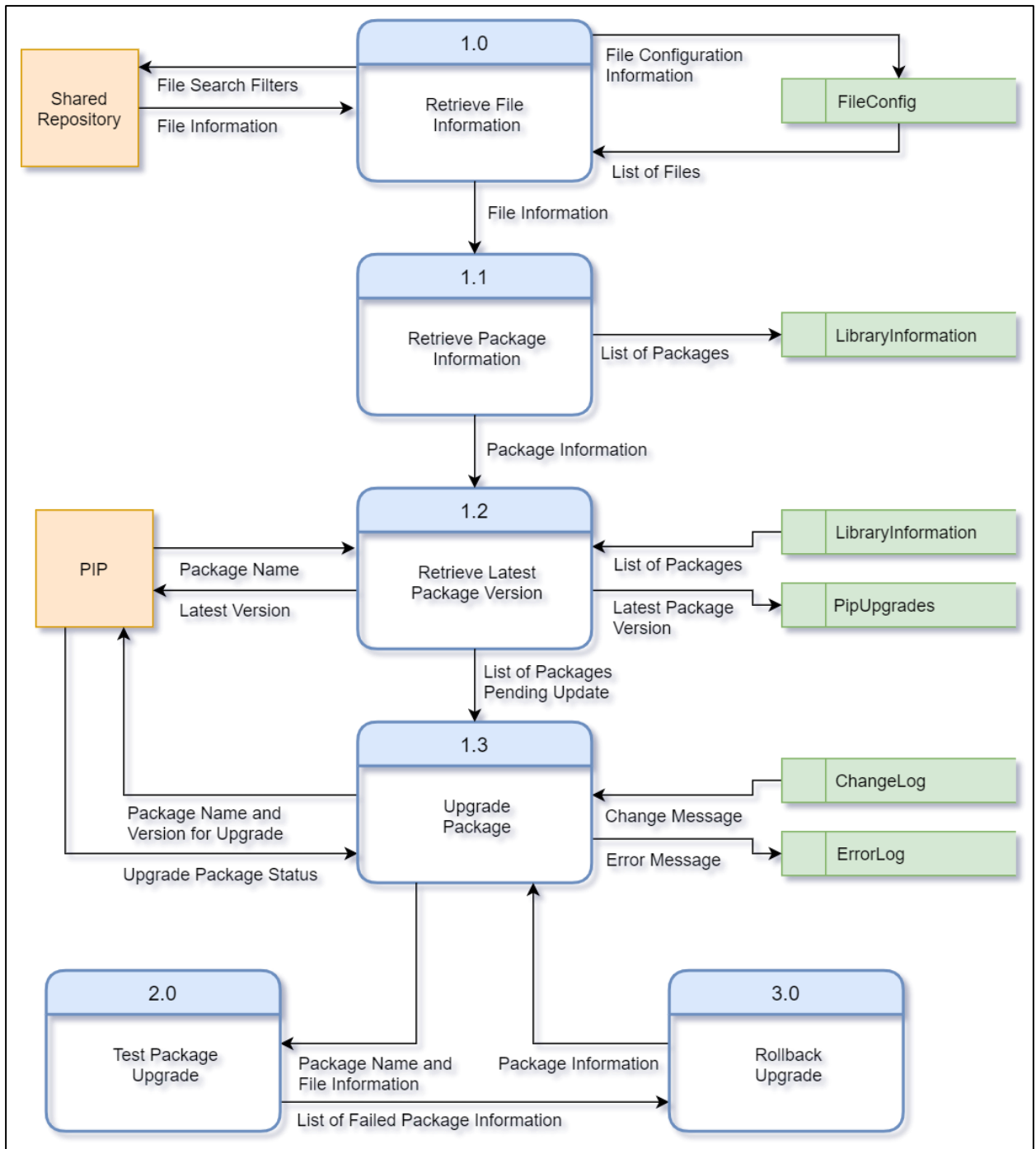


Figure 6-8: Data flow diagram

As seen in Figure 6-8, certain data stores are needed to store the data being processed. The current environment advises that the most feasible collective data store for the solution would be an SQL database hosted on the existing SQL server.

6.4 Technology layer

The technology architecture that was presented as part of the initial conceptual design was a high-level overview of what technology should be used. During the current iterative cycle, the overall technical ecosystem was defined in order to establish and identify where the solution lives and what existing technology it can utilise. This technical landscape is illustrated through Figure 6-9.

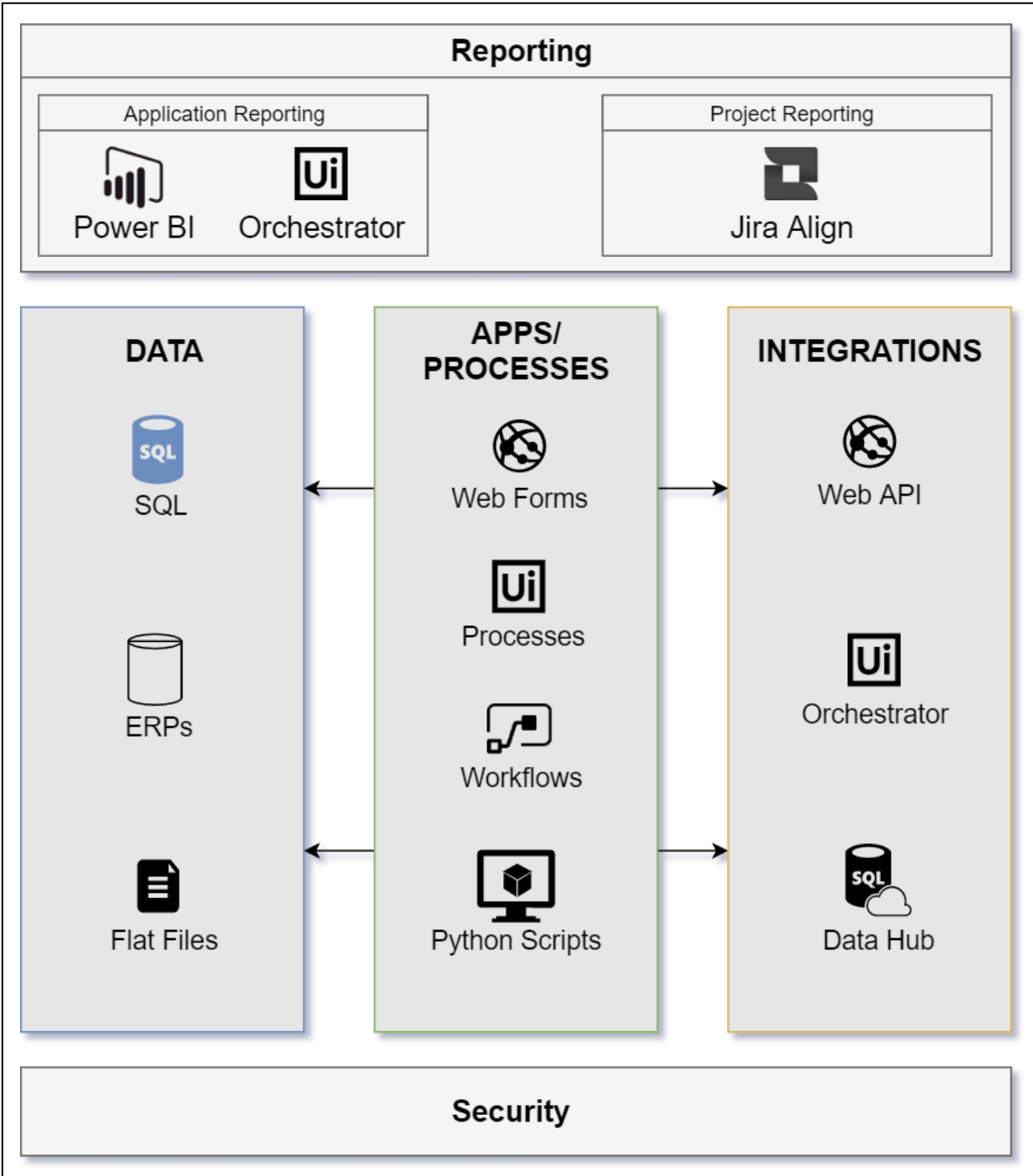


Figure 6-9: Technical landscape
113

As seen in Figure 6-9, there are five layers that exist within the technical landscape:

- **Data:** The data layer comprises all data sources that are used to store the raw and processed data needed to enable the apps and workflows.
- **Apps/processes:** The apps/processes layer encapsulates all automation (and automation enablement) solutions in whichever form they are presented (K2 forms or workflows, web forms, UiPath processes, Power Automate workflows or Python scripts).
- **Integrations:** The integrations layer represents any third-party platforms that the apps and processes need to interface with, either for more data or for more functionality.
- **Reporting:** The reporting layer stretches across the data, apps/processes and integration layers as reporting needs to be conducted in an end-to-end fashion to ensure that there is visibility over all aspects of the environment.
- **Security:** Like the reporting layer, the security layer also spans across the three layers as security is a crucial part of ensuring the integrity of the ecosystem. Security is implemented within each individual component of the ecosystem.

For the sake of the solution at hand, only three components will be utilised within the ecosystem, namely SQL database, the Power BI reporting and the Scripts virtual machine, which is where Python scripts are run from as per their configured schedule.

6.5 Report prototype

A prototype was built according to the specification outlined in the current iterative cycle of the proposed design, which has been elaborated upon throughout this chapter. The prototype mainly reads data from an SQL database, evaluates the retrieved data according to a set of business rules before gathering new data to be stored in the database. A two-tab Power BI report provides high-level insight into the results of the automated solution through visualising the data available in the database.

Figure 6-10 illustrates the summative detail extracted from the data populated by the DPA solution into the data structure depicted by Figure 6-7, through the data flow process outlined in Figure 6-8.



Figure 6-10: Auto-upgrade process last run summary tab: Power BI report

It can be seen from Figure 6-10 that the high-level summary includes figures detailing the success and failure of processes and libraries that were upgraded during the last run. The sum of “*Process Upgrades*” and “*Process Rollbacks*” equals “*Processes Pending Upgrades*” while the sum of “*Library Upgrades Within Processes*” and “*Library Rollbacks Within Processes*” equals “*Pending Library Upgrades*”.

Figure 6-11 displays a lower level of detail with regard to the number of non-business-critical processes with a QA environment, the number of libraries compared by major version difference, the number of pending upgrades by library, the number of library upgrades by process and the number of library rollbacks by process.

The sum of the number of pending upgrades by library seen in the “Number of Pending Upgrades by Library” visual in Figure 6-11 corresponds with the “Processes Pending Upgrade” visual in Figure 6-10. It can also be seen that the sum of the upgrades and rollbacks depicted by “Number of Library Upgrades by Process” and “Number of Library Rollbacks by Process” equates to the “Pending Library Upgrades” in Figure 6-10.

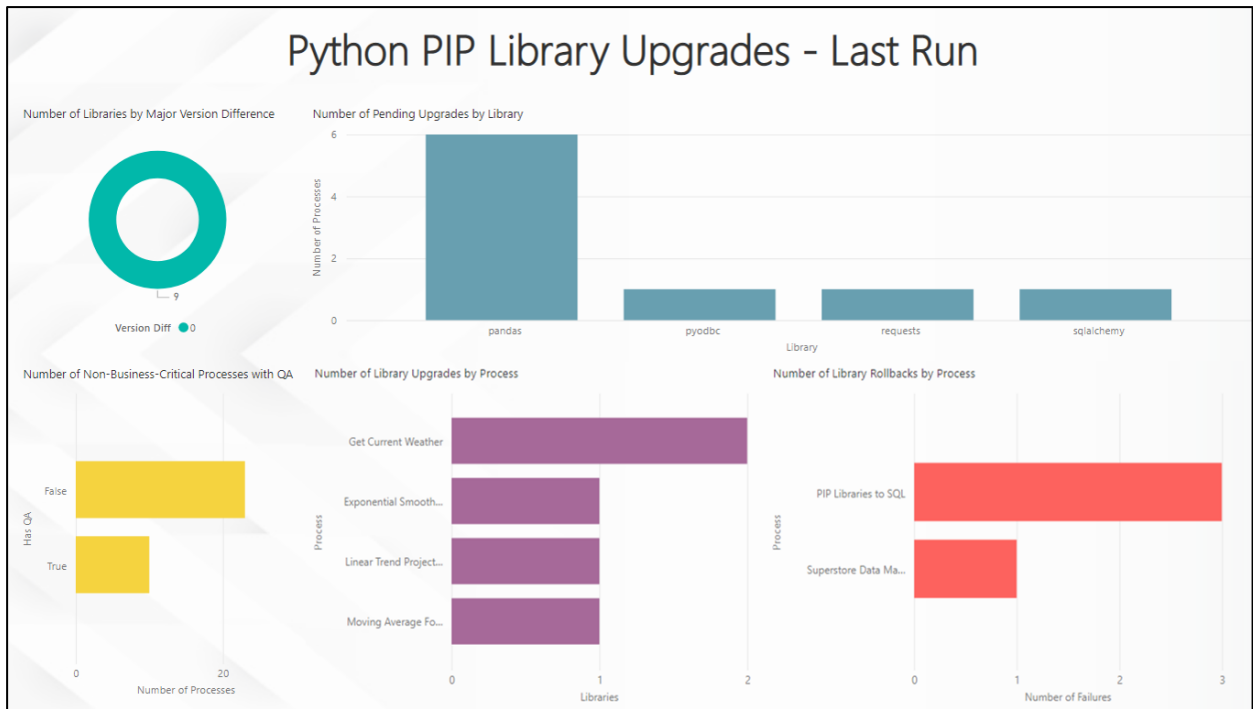


Figure 6-11: Auto-upgrade process upgrades tab: Power BI report

The graph title “Number of Non-Business-Critical Processes with QA” in Figure 6-11 illustrates the number of processes that are business-critical. The scope of the DPA solution is focused on non-business-critical processes, which means that any business-critical processes would be excluded from being processed by the DPA solution. As seen in the graph presented in Figure 6-11, ten of the 33 processes are excluded due to their business-criticality characteristic.

The graph titled “Number of Libraries by Major Version Difference” in Figure 6-11 illustrates the number of libraries compared by the major version difference between the latest version and the current version. The difference is calculated by subtracting the first number of the current version number from the first number of the latest version. Based on the data represented on the report, there are no processes that are a major version behind, which means that there are no processes that currently need to be upgraded manually and not by the DPA solution.

The graph titled “Number of Pending Upgrades by Library”, illustrated in Figure 6-11, depicts all the libraries that require an upgrade by the number of processes that use the library, therefore requiring an upgrade.

6.6 Joint application development (JAD) workshop

A second JAD workshop was executed to demonstrate the functionality of the solution and to gather more feedback on its capability in order to improve the sustainable quality of the solution. The JAD workshop conducted during this iterative cycle was conducted as a solution demo (presented to all project stakeholders involved in a system demo) with the aim of gathering feedback on whether or not the proposed design and build have met the requirements, based on the incorporation of the proposed changes from the initial conceptual design and build.

Post-solution demo, the following questions were asked and answered unanimously by a panel of project stakeholders (including the business owner, product manager and product owner) within the case study environment:

Question	Answer
Does the current solution meet all requirements?	Yes
Have the proposed changes been incorporated appropriately in the solution?	Yes
Is the proposed solution ready for use in QA?	Yes
Does anything need to be changed before this prototype may be deployed to QA?	No

Table 6-6: Tabular representation of the JAD workshop feedback

6.7 Data collection

The prototype designed and built as part of this chapter serves as a tool to collect the cost, time and scope variables of DPA Execution (Determining D: DPA Execution, as seen in Figure 6-12 adapted from Figure 3-8). The grey shapes in Figure 6-12 indicate which variables have already been determined, while the white and yellow shapes indicate which variables still need to be determined and will be determined in this chapter, respectively.

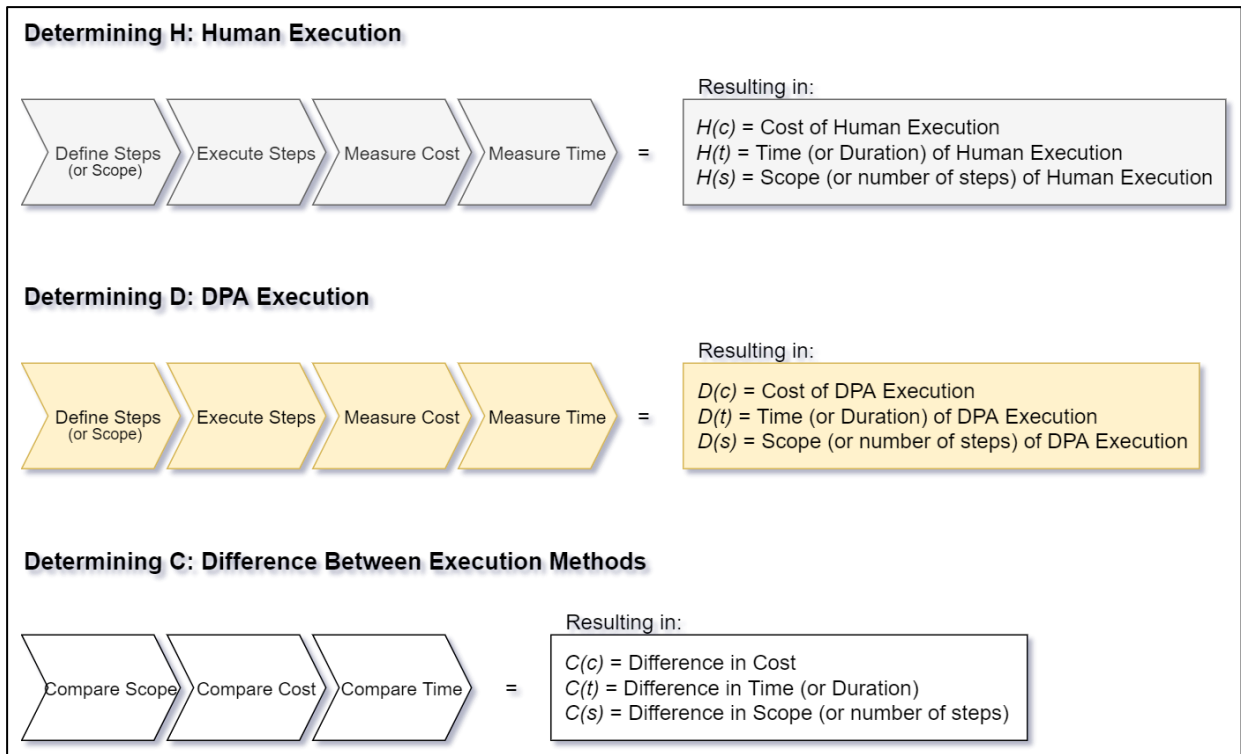


Figure 6-12: The determination of variables D in terms of c, t and s for DPA execution

As seen in Figure 6-12, collecting and determining each of these data points can be formalised into calculations. The variable keys relevant to Determining D: DPA Execution are expanded below:

- D = DPA solution execution of the process
- $D(c)$ = Cost ($c_1 + c_2$)
 - c_1 = Indirect cost
 - c_2 = Direct cost
- t = Time ($t_1 + t_2 + t_3$)
 - t_1 = Process preparation time
 - t_2 = Process execution time
 - t_3 = Process termination time
- s = Scope (in terms of the number of steps)

	Cost (c)	Time (t)	Scope (s)
DPA Solution Execution (D)	$D(c)$	$D(t)$	$D(s)$

Table 6-7: Tabular representation of analytical calculations to be performed while determining D through DPA execution

6.7.1 Time

The execution time of the process may include multiple factors, which may vary between cases and may also be adapted to include development time of an artefact. For the sake of this study, only execution time will be considered (excluding development time). Time is, therefore, calculated as Y below:

$$t = \text{Time } (t_1 + t_2 + t_3)$$

- t_1 = Process preparation time
 - Includes tasks like:
 - Opening the shared folder
 - Opening each solution
 - Opening lists
- t_2 = Process execution time
 - Includes tasks like:
 - Evaluating business rules
 - Testing the upgrades
 - Initiating rollbacks
 - Applying changes to production
- t_3 = Process termination time
 - Includes tasks like:
 - Cleaning the environment after the rollback
 - Reporting on progress

Preparation, execution and termination may be split up to contain different tasks depending on the scope of the process. In a case where a design pattern is used, $D(t)$ could be split up to calculate the time it takes per layer (for example: initialisation, data access, domain logic, helpers, presentation, clean up). Preparation could typically include the initialisation of an environment and the appropriate variables. Execution would include the steps that work towards solving the problem once the environment has been initialised and before the environment is disposed. Termination would refer to tasks needed to clean up, close and dispose of an environment and the elements used within the process. Table 6-8 represents the execution time of the DPA solutions split up between the preparation, execution and terminate phases.

Run	Rollbacks	Runtime per run (in seconds)			
		$t_1 =$ preparation	$t_2 =$ execution	$t_3 =$ termination	Time (t) = $t_1 + t_2 + t_3$
Run 1	4	9	27	17	53
Run 2	4	9	28	17	54
Run 3	4	9	27	16	52
Run 4	4	9	26	16	51
Run 5	4	9	26	17	52
Run 6	4	9	28	16	53
Run 7	3	9	26	16	51
Run 8	3	9	27	14	50
Run 9	3	9	26	15	50
Run 10	3	9	26	15	50
Run 11	3	9	26	16	51
Run 12	3	9	25	17	51
Run 13	2	9	27	14	50
Run 14	2	9	24	16	49
Run 15	2	9	23	17	49
Run 16	2	9	25	15	49
Run 17	2	9	26	15	50
Run 18	2	9	25	15	49
Run 19	1	9	24	15	48
Run 20	1	9	24	14	47
Run 21	1	9	24	16	49
Run 22	1	9	22	16	47
Run 23	1	9	24	15	48
Run 24	1	9	22	16	47
Run 25	0	9	8	0	17
Run 26	0	9	10	0	19
Run 27	0	9	10	0	19
Run 28	0	9	8	0	17
Run 29	0	9	9	0	18
Run 30	0	9	9	0	18
Average time: $f(t) = \frac{\sum_{n=1}^3(t_n)}{30}$		43.57 seconds			

Table 6-8: Tabular representation of DPA solution process execution time

It can be seen from Table 6-8 that, on average, it takes approximately 44 seconds to run the DPA solution. In all 30 runs, all 33 solutions were evaluated of which only eight met the business rules. Tests were run in sets of six with each set focused on a specific number of rollbacks. It can be

seen that if there are no rollbacks, the process is executed faster. There is a gradual increase in process execution time as the number of rollbacks increases.

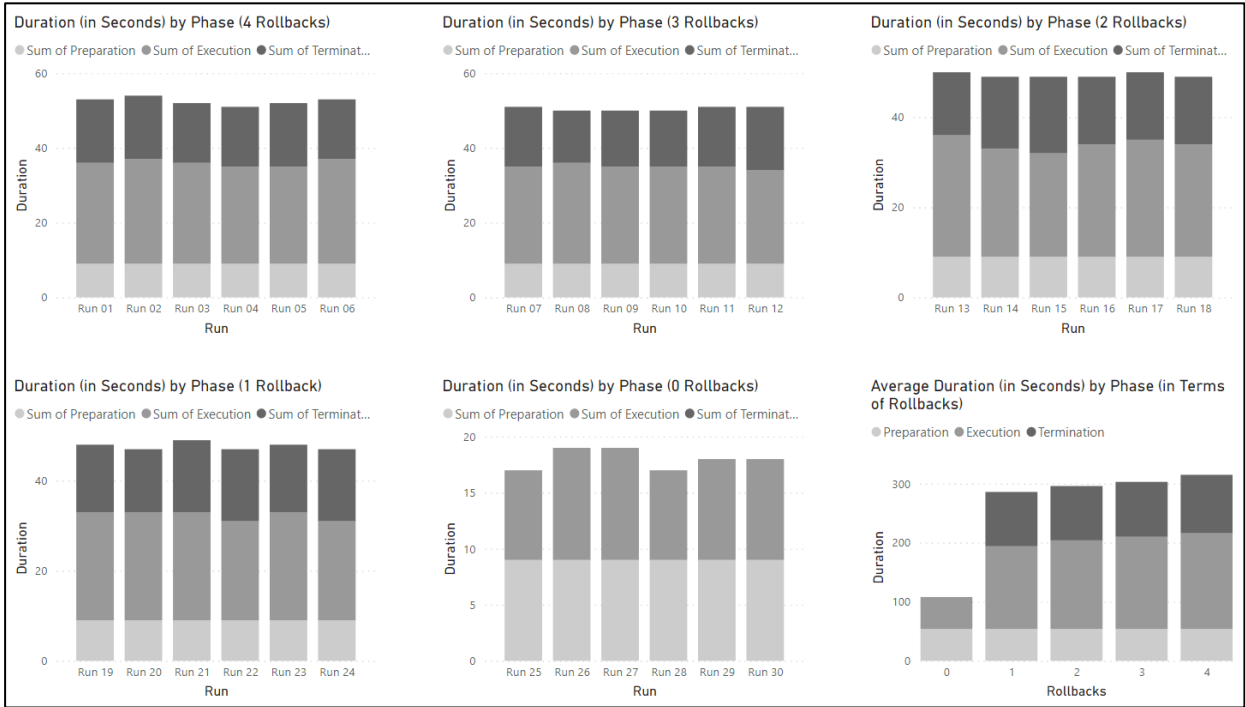


Figure 6-13: The duration (in seconds) of DPA execution by phase

It can also be seen from Table 6-8 and Figure 6-13 that runs with no rollbacks have no termination time, which is due to the environment not needing to be cleaned up and reset after the execution phase. When a rollback occurs, the environments need to be restored to their previous state before the next process run can be started. No rollbacks also result in shorter execution phase runtimes (or durations) that, together with no termination time, account for the approximate 35 second difference between runs with no rollbacks and runs with four rollbacks. It can be seen from Figure 6-13 that the duration remains consistent between runs grouped by the number of rollbacks with no significant (unexplainable) outliers able to skew any averages denoted by the collected data.

6.7.2 Cost

The cost of execution can be broken up into direct and indirect costs, which may be adapted as necessary for different use cases.

For the sake of this study, cost is calculated as follows:

$$c = \text{Cost} (c_1 + c_2)$$

- c_1 = Direct cost
 - Includes:
 - Direct labour
- c_2 = Indirect cost
 - Includes:
 - Infrastructure costs

The direct cost of the DPA solution implementation is made up of the cost of direct labour, which refers to the cost of having a human execute the process. Since this process is to be executed by the DPA solution, any manual intervention will be managed as operational support, which means that there is no need for a dedicated resource for the process. The direct cost of the implementation of the DPA solution is, therefore, R0.00. Development cost is excluded from the cost calculation of this execution method. The calculation of additional costs will be addressed in the next chapter to explain the impact that additional costs would have on the realised cost saving of implementing the DPA solution. As part of the indirect infrastructure costs, an Azure SQL database (hosting the database design) and an app service (to host the Python script) are needed. An estimate extracted from Microsoft’s Azure pricing calculator (as detailed in table 6-9) shows that the annual infrastructure cost is estimated at R18 200.00.

Service type	Region	Description	Estimate cost per run	Estimated monthly cost	Estimated annual cost
Azure SQL Database	South Africa North	Single Database, vCore, RA-GRS Backup Storage, General Purpose, Provisioned, DC-series, Local Redundancy, 1 8 vCore instance(s) x 730 Hours, 16 GB Storage, 0 GB Backup Storage	R0.06	R45.00	R540.00
App Service	South Africa North	Basic Tier; 1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 730 Hours; Windows OS	R2.02	R1 470.00	R17 660.00
Support		Support	R0.00	R0.00	R0.00
Total			R2.08	R1 515.00	R18 200.00

Table 6-9: Tabular representation of the indirect cost of the DPA solution implementation

The estimated annual cost of this solution is, therefore, calculated as follows:

$$c = (c_1 + c_2)$$

$$c = R0,00 + R18\,200,00$$

$$c = R18\,200,00$$

The annual cost of implementing the DPA solution is R18 200.00. The cost per run is R2.08.

6.7.3 Scope

As seen in Figure 6-3, the process consists of nine high level steps, namely:

1. Open shared folder.
2. Open each solution.
3. Amend configuration.
 1. Find packages used and amend the existing list as necessary.
 2. Amend solution configuration.
4. Compile list of packages with new available version.
5. Get package criticality parameters.
6. Get package environment parameters.
7. Test each solution with the upgrades (if they meet the criteria).
8. Revert to previous library version (on test failure).
9. Publish solution to production.

Step 10 (report on success and failure) happens automatically in the background and is not considered as an applicable process step.

The number of steps depends on the journey that each solution takes through the process flow. The maximum number of high-level steps to be executed is eight steps, as seen in Table 6-10.

New solution	Has config change	Meets business rules	Passed test	Number of steps
FALSE	FALSE	FALSE	FALSE	4
FALSE	FALSE	FALSE	TRUE	4
FALSE	FALSE	TRUE	FALSE	6
FALSE	FALSE	TRUE	TRUE	7
FALSE	TRUE	FALSE	FALSE	5
FALSE	TRUE	FALSE	TRUE	5
FALSE	TRUE	TRUE	FALSE	7
FALSE	TRUE	TRUE	TRUE	8
TRUE	FALSE	FALSE	FALSE	5
TRUE	FALSE	FALSE	TRUE	5
TRUE	FALSE	TRUE	FALSE	7
TRUE	FALSE	TRUE	TRUE	8
TRUE	TRUE	FALSE	FALSE	5
TRUE	TRUE	FALSE	TRUE	5
TRUE	TRUE	TRUE	FALSE	7
TRUE	TRUE	TRUE	TRUE	8

Table 6-10: Tabular representation of the use cases depicted by the journey through the process flow

6.8 Conclusion

Based on the need for an artefact to be developed for the data collection portion of this study, the outcome of this chapter is the proposed design and prototype. The second iteration through TOGAF has assisted in establishing the updated organisational environment, along with the business strategy and business model, with relation to the dimension data's operational model. This iterative cycle has rendered the observation that there are no longer any gaps that exist within the case study environment. The application architecture has been refined with more detail in this chapter, along with the data architecture and technology layer.

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influence time are process preparation time, process execution time and process termination time. These three phases equate to the total duration (or time) of having a DPA solution execute the process.

From the collected data, it can be seen that cost is influenced by direct and indirect cost. There was no direct cost associated to having a DPA solution execute the process; however, indirect cost consists of infrastructure costs (specifically referring to hardware, support and licensing costs).

The scope of the process execution is determined by the maximum number of steps that can be executed within a process run.

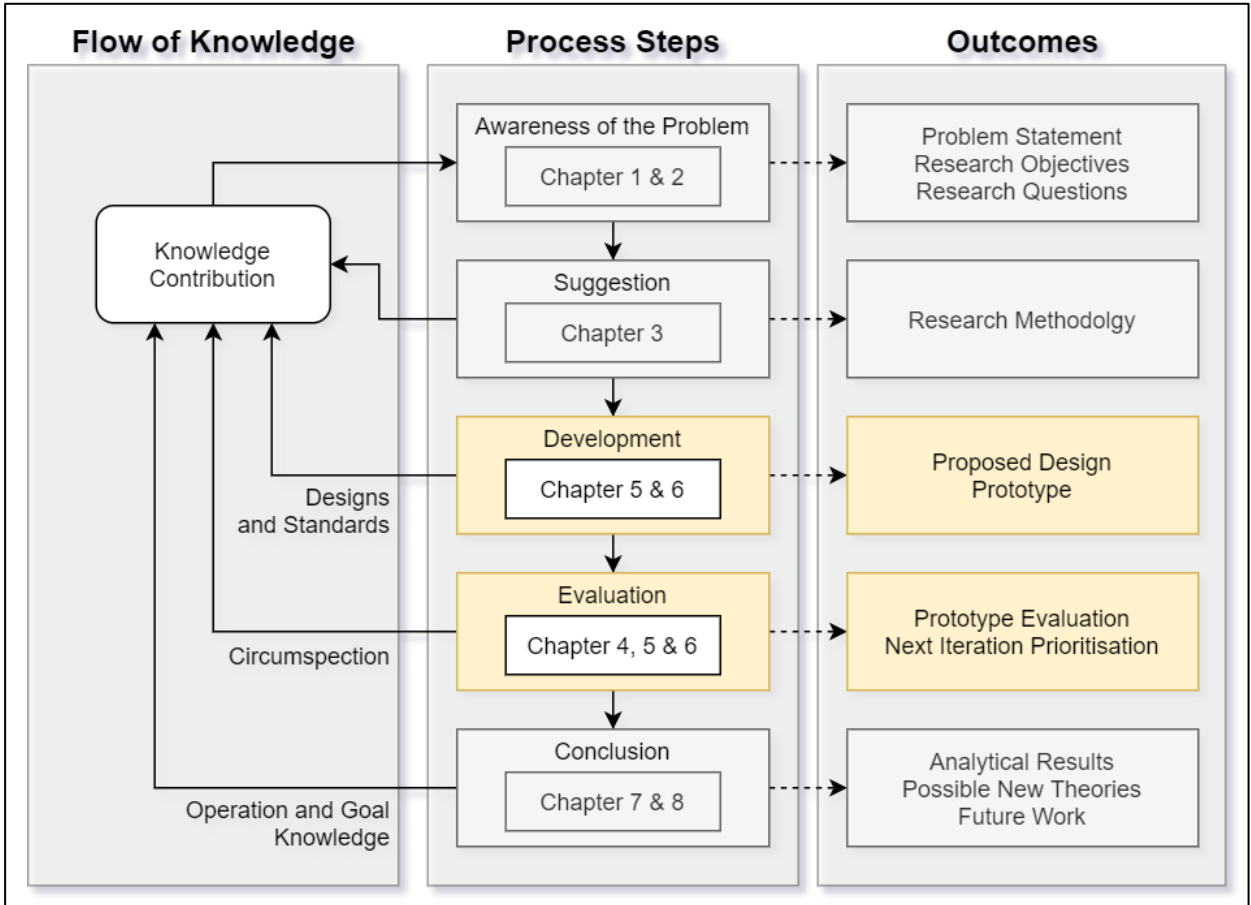


Figure 6-14: Chapter 6 DSR study layout contribution

In this chapter, the proposed design and build were developed and evaluated, in line with the DSR study layout seen in Figure 6-14. The next chapter will focus on utilising the proposed design and prototype to analyse the data that has been collected.

CHAPTER 7: DATA ANALYSIS

7.1 Introduction

The data collection portion of this study (detailed in Chapters 4, 5 and 6) rendered two datasets (human execution and DPA solution execution) measuring three variables (time, scope and cost).

The analysis of the quantitative data that has been collected entails the following:

- Comparing the average time duration of a person completing the process and completing the process through the use of a DPA solution.
- Comparing the estimated cost of a person completing the process and completing the process through the use of a DPA solution.
- Comparing the scope dynamics of a person completing the process and completing the process through the use of a DPA solution.

Based on the above and Figure 7-1, it can be summarised that three comparisons need to be made in order to derive conclusions from the data analysis (Determining C: Difference Between Execution Methods, as seen in Figure 7-1 adapted from Figure 3-8). The grey shapes in Figure 7-1 indicate which variables have already been determined, while the yellow shapes indicate which variables will be determined in this chapter.

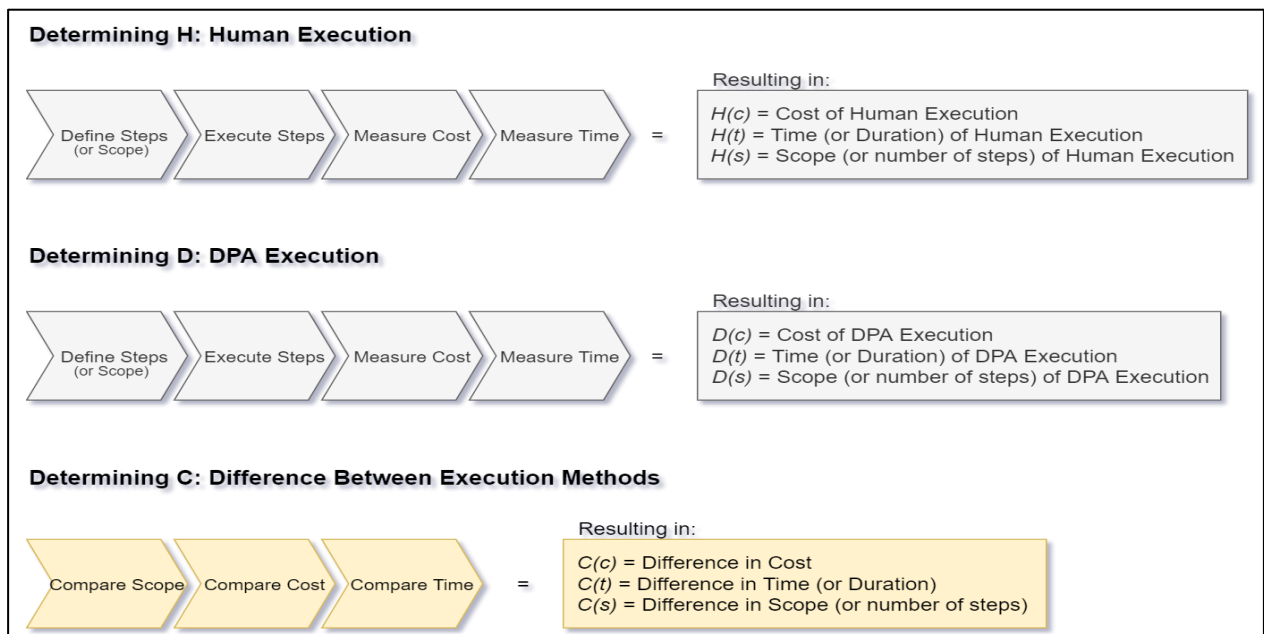


Figure 7-1: The determination of variables C in terms of c, t and s for human execution

As seen in Figure 7-1, collecting and determining each of these data points can be formalised into calculations. The variable keys relevant to Determining C: Difference Between Execution Methods are expanded below:

- C = Difference between execution methods
- $C(c) = \text{Cost } (H(c) - D(c))$
 - $H(c)$ = Human execution cost
 - $D(c)$ = DPA solution execution cost
- $C(t) = \text{Time } (H(t) - D(t))$
 - $H(t)$ = Human execution time
 - $D(t)$ = DPA solution execution time
- $C(s) = \text{Scope (in terms of the number of steps) } (H(s) - D(s))$
 - $H(s)$ = Human execution scope
 - $D(s)$ = DPA solution execution scope

	Cost (c)	Time (t)	Scope (s)
Difference between execution methods (c)	$C(c)$	$C(t)$	$C(s)$

Table 7-1: Tabular representation of analytical calculations to be performed while determining A through human execution

Table 7-1 summarises the variables to be identified by this chapter. Sub-section 7.2 provides more insight into the analysis and comparison of the time variable between execution methods, followed by sub-section 7.3 that explains the cost analysis with some attention on the impact of additional costs. Sub-sections 7.4 and 7.5 present the scope analysis and discussion on analysis, respectively, before sub-section 7.6 concludes the chapter.

7.2 Time analysis

The average time gathered from the different execution methods has been measured on a ‘per run’ basis but will also need to be evaluated on a ‘per annum’ basis. The process is expected to run once a week, every week, resulting in 52 runs per annum. The summary of time analysis for both methods of execution, as well as the comparison thereof are summarised in Table 7-2.

	Runtime per run (in seconds)		
Phase	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Process preparation time (t_1)	892	9	883
Process execution time (t_2)	161	22	139
Process termination time (t_3)	374	13	361
Total	1 427	44	1 383
	Runtime per annum (in seconds)		
Phase	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Process preparation time (t_1)	46 381	468	45 913
Process execution time (t_2)	8 381	1 144	7 237
Process termination time (t_3)	19 465	676	18 789
Total	74 227	2 288	71 939

Table 7-2: Tabular representation of average runtime per run (in seconds)

It can be seen in Table 7-2 that there is a time saving of approximately 23 minutes and 3 seconds (1 383 seconds) per run and approximately 19 hours 59 minutes (71 939 seconds) per year. This may be referred to as the actual realised time saving since the difference in runtime (or duration) between human execution and the DPA solution has been measured while executing the exact same process instead of the time-saving being inferred (which is usually done as part of requirements gathering).

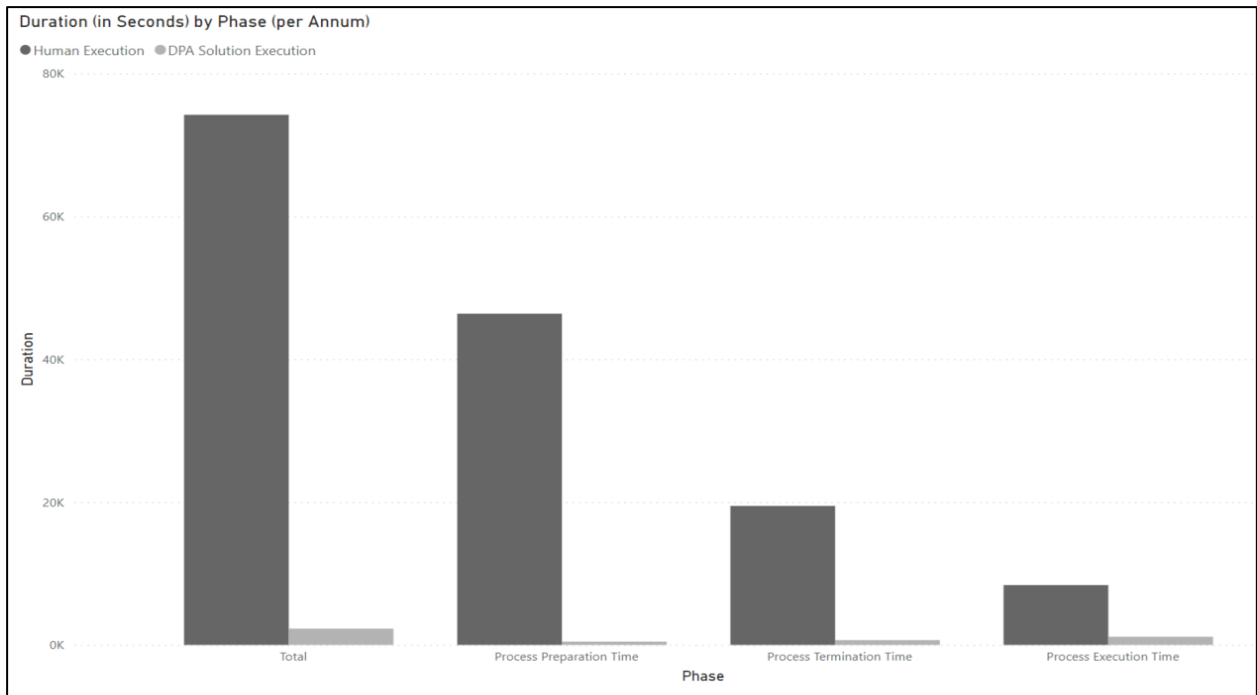


Figure 7-2: The duration (in seconds) by phase (per annum)

Figure 7-2 shows the duration (in seconds), on average, of the process being executed by human execution and DPA solution execution, split up by phase. It can be seen that there is a significant difference between the time it takes a human to execute the process as compared to the DPA solution executing the process. The preparation phase takes the longest when a human executes the process, but is the shortest phase when executed by the DPA solution. Similarly, the execution phase is the shortest when executed by a human, but takes the most time when being executed by the DPA solution. The significant difference in time may be attributed to some of the following factors:

- Less wait time when switching between application screens
- Reduced search and filter time when trying to match and join versions, packages and projects
- Minimised risk of skipping a step in the process
- Minimised risk of incorrectly entering version numbers

7.3 Cost analysis

The average cost gathered from the different execution methods has been measured on a 'per run' basis but will also need to be evaluated on a 'per annum' basis. The process is expected to run once a week, every week, resulting in 52 runs per annum. The summary of cost analysis for both methods of execution, as well as the comparison thereof are summarised in Table 7-3.

	Cost per run (in rands)		
Cost	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Direct cost (c_1)	R70.71	R0.00	R70.71
Indirect cost (c_2)	R27.11	R2.08	R25.03
Total	R97.82	R2.08	R95.74
	Cost per annum (in rands)		
Cost	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Direct cost (c_1)	R3 676.92	R0.00	R3 676.92
Indirect cost (c_2)	R1 409.72	R108.16	R1 301.56
Total	R5 086.64	R108.16	R4 978.48

Table 7-3: Tabular representation of average cost per run (in rands)

It can be seen in Table 7-3 that there is a cost saving of approximately R96.00 per run and approximately R4 978.00 per year. This may be referred to as the actual realised cost saving since the direct and indirect cost difference between human execution and the DPA solution has been measured while executing the exact same process instead of being inferred (which is usually done as part of requirements gathering).

Since the cost of the human execution is partially dependent on how long it takes an employee to execute the process, the factors mentioned in section 7.2 could also influence the difference in cost between execution methods.

In order to mitigate the difference in cost, focus should be placed on how to reduce the indirect cost of human execution. Indirect cost per employee within the case study environment amounts to approximately R140 692.00 per annum, with R108 692.00 allocated to licensing and R32 000.00 allocated to hardware. By decreasing licensing costs per employee, the cost difference between execution methods would decrease too. Although the difference per run may be small, the saving per annum would be more noticeable.

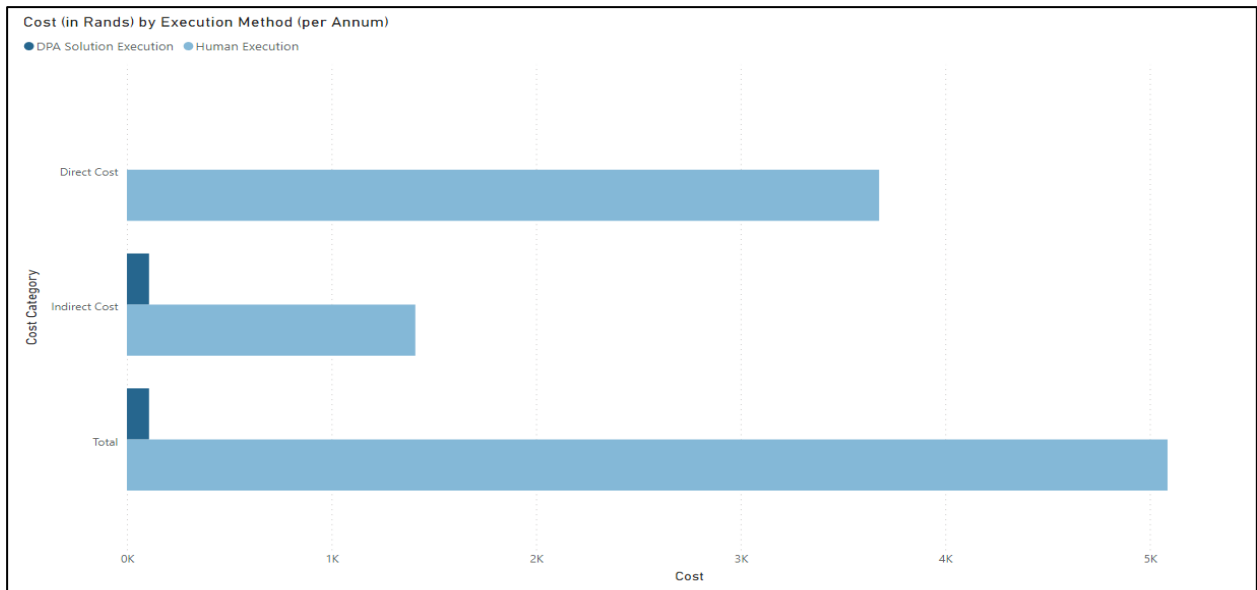


Figure 7-3: The cost (in rands) by execution method (per annum)

It can be seen in Figure 7-3 that no direct cost is captured for the DPA solution execution as it was not deemed applicable in the case study environment. Direct cost could be used to identify the difference between long-term and short-term actual realised cost saving. This is expanded upon in section 7.3.1. The exclusion of direct cost for the DPA solution execution method significantly impacts the comparison between execution methods due to direct cost being higher than indirect cost for the human execution method. The exclusion of direct cost results in the total cost for DPA solution execution being determined solely by indirect cost, which increases the margin between execution methods substantially.

Additional factors that contribute to the difference in cost between execution methods include:

- Employees needing multiple developer licenses due to the wide scope of work that is addressed by the employee (for example, building Power BI reports, building web applications using ASP.NET Core and developing RPA solutions in UiPath. All three technology stacks have their own costs associated to them). By narrowing down the scope of work of each employee, the number of licenses may also be reduced. By making automations part of the team, it allows for the redistribution of licenses to reduce costs on a team level. For example, having one user license for the whole team (linked to one dedicated user), rather than one user license per employee in a team.
- The specifications of the hardware used by each employee also impact the price (for example, an 8th Gen i7 Dell laptop would cost more than a 7th Gen i5 laptop).
- Since development and operational support is excluded from the DPA solution execution direct cost calculation, the difference in direct cost is higher.

7.3.1 Additional costs

Annexure A shows that, based on the information collected about the process, the estimated duration of the implementation is 860 hours. It was therefore decided that 800 hours would be allocated to the requirements gathering and the development lifecycle of the DPA solution, of which only 80 hours were used on the development of the second prototype and 20 hours were spent on testing and review. This is categorised as development cost.

The balance of the 700 hours is divided into requirements gathering and the development of the first prototype. The first prototype took approximately 200 hours to build (including 40 hours of testing) with the remaining 500 hours spent on requirements gathering. Since there was a decision to pivot between prototypes, the development time of the first prototype was written off by the business as sunken cost, even though there were components that were reused between prototypes, reducing the number of hours that needed to be spent on the second prototype. The time spent on requirements gathering could be categorised as requirements gathering costs and would be calculated in a similar way to development cost, if it were to be included in the cost calculation.

7.3.1.1 Requirements gathering cost

Requirements gathering cost was not taken into consideration when calculating the direct cost of the DPA solution. This section serves to evaluate how requirements gathering cost would be calculated if it were taken into consideration. The requirements gathering took 500 hours. With the rate per hour of the business analyst (BA) taken into consideration, the requirements gathering cost accounts to R85 000. In a case where requirements gathering was outsourced, the rate per hour would be significantly more.

In order to calculate the number of runs that it would take for the costs to break even, the increase of cost per annum would need to be taken into consideration. This means that the cost per execution method should increase by approximately 10% per year to cater for the increase in direct and indirect costs of both execution methods.

To account for the annual inflation in cost and to calculate how long it would take for the initial requirements gathering cost to be recovered, it was found that the present value formula would best fit the situation to calculate the time period that would lapse before a point of equilibrium is achieved.

$$P = A \times \frac{(1+i)^n - 1}{i}$$

Where:

- P (Requirements gathering cost) = R85 000.00
- A (Cost per year) = R4 978.48
- i (Annual percentage increase) = 0.10 (10%)
- n (Number of years)
- r (Number of runs) = $52n$

To calculate n , this formula is transformed into:

$$n = \log_{(1+i)}\left(\frac{Pi}{A} + 1\right)$$

$$n = \log_{(1.10)}\left(\frac{85\,000(0.10)}{4\,978.48} + 1\right)$$

$$n = \log_{(1.10)}(2.71)$$

$$n = 10.46$$

$$\therefore r = 52(10.46)$$

$$r = 543.92$$

By including a 10% increase in direct and indirect cost per year, the solution would need to be in production for approximately 11 years (544 runs) before cost saving is realised, when taking requirements gathering cost into account. It can be seen in Figure 7-4 that a breakeven point is achieved towards the middle of year 10 where the diminishing cost (which refers the declining difference in cost between execution methods and the requirements gathering cost) breaks even with the fixed requirements gathering cost while there is a significant actualised cost saving during years 12 and 13.

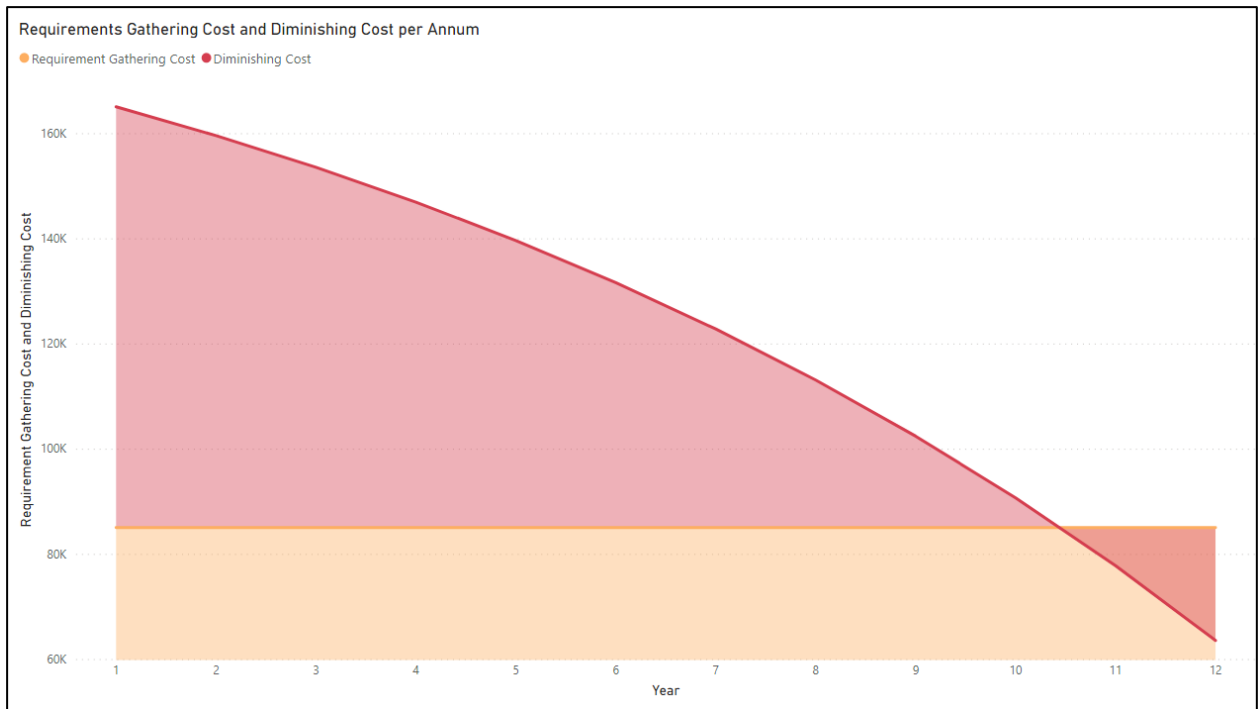


Figure 7-4: Requirements gathering cost and diminishing cost per annum

Figure 7-5 shows that between runs 520 and 544 the diminishing cost (which refers the declining difference in cost between execution methods and the requirements gathering cost) converges towards an equilibrium (or breakeven) point. The calculation above takes a 10% annual increase into account; however, there is an actual increase of 0.19% per run. This means that on an annual basis (as seen in Figure 7-4), taking the annual cost into account, breakeven occurs somewhere in the fifth month of the fifth year of the solution being in production.

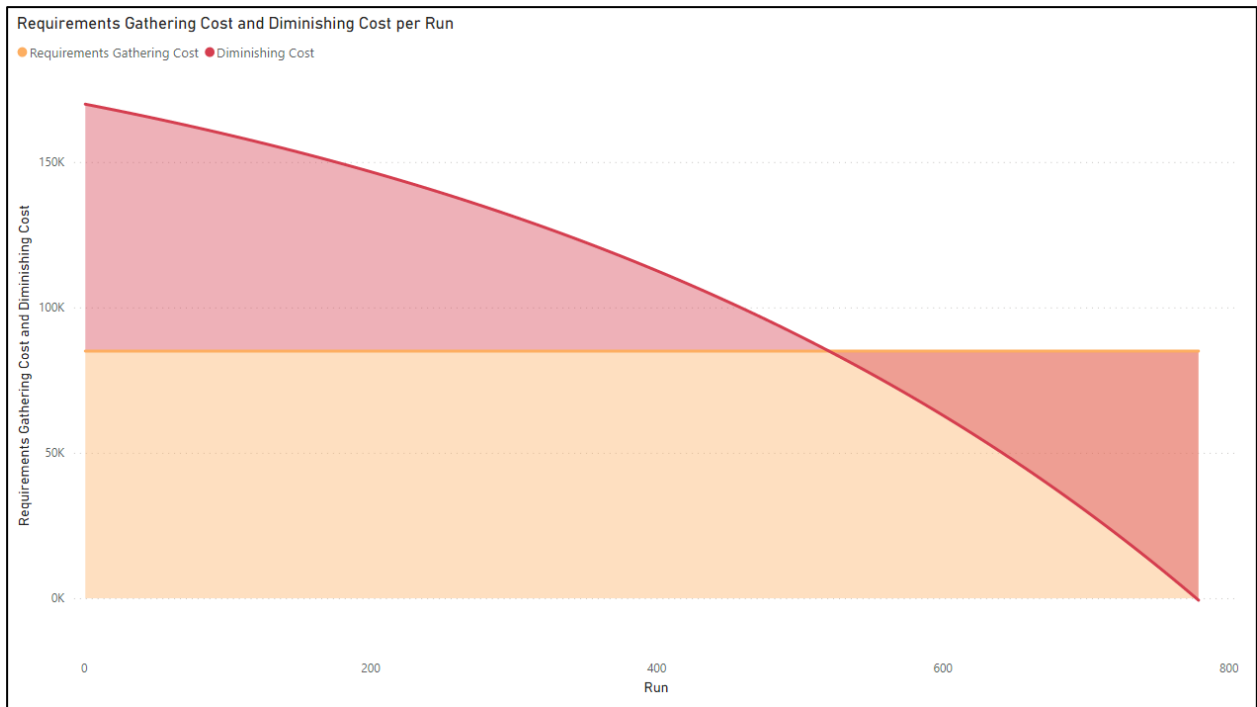


Figure 7-5: Requirements gathering cost and diminishing cost per run

Figure 7-5 takes the 0.19% per run into account and shows that the breakeven appoint occurs on the 146th run. It can, therefore, be stated that the actual time saving of this solution is realised between runs 147 and 152.

7.3.1.2 Sunken cost: First prototype development cost

Sunken cost was not taken into consideration when calculating the direct cost of the DPA solution. This section serves to evaluate how sunken cost would be calculated if it were taken into consideration. The first prototype took 160 hours to develop and 40 hours to test. With the rate per hour of the developer and tester taken into consideration, the development (including testing) cost of the second prototype amounts to R32 280. In a case where development was outsourced, the rate per hour would be significantly more.

In order to calculate the number of runs that it would take for the costs to break even, the increase of cost per annum would need to be taken into consideration. This means that the cost per execution method should increase by approximately 10% per year to cater for the increase in direct and indirect costs of both execution methods.

To account for the annual inflation in cost and to calculate how long it would take for the initial development (now sunken) cost to be recovered, it was found that the present value formula

would best fit the situation to calculate the time period that would lapse before a point of equilibrium is achieved.

$$P = A \times \frac{(1+i)^n - 1}{i}$$

Where:

- P (Sunken cost) = R32 280.00
- A (Cost per year) = R4 978.48
- i (Annual percentage increase) = 0.10 (10%)
- n (Number of years)
- r (Number of runs) = $52n$

To calculate n , this formula is transformed into:

$$n = \log_{(1+i)}\left(\frac{Pi}{A} + 1\right)$$

$$n = \log_{(1.10)}\left(\frac{32\,280(0.10)}{4\,978.48} + 1\right)$$

$$n = \log_{(1.10)}(1.65)$$

$$n = 5.25$$

$$\therefore r = 52(5.25)$$

$$r = 273$$

By including a 10% increase in direct and indirect cost per year, the solution would need to be in production for approximately six years (273 runs) before cost saving is realised, when taking sunken cost into account. It can be seen in Figure 7-6 that a breakeven point is achieved towards the middle of year five where the diminishing cost (which refers the declining difference in cost between execution methods and the sunken cost) breaks even with the fixed sunken cost, while there is a significant actualised cost saving during years six and seven.

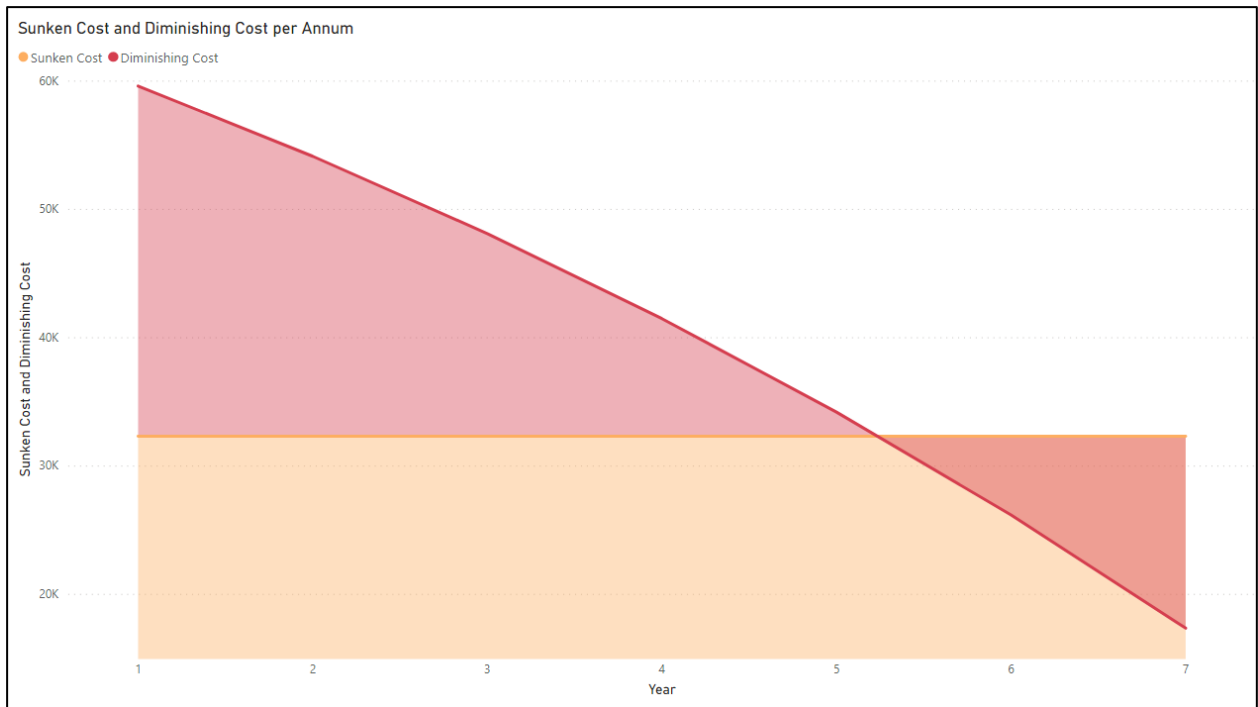


Figure 7-6: Sunken cost and diminishing cost per annum

Figure 7-7 shows that between runs 260 and 273 the diminishing cost (which refers the declining difference in cost between execution methods and the sunken cost) converges towards an equilibrium (or breakeven) point. The calculation above takes a 10% annual increase into account; however, there is an actual increase of 0.19% per run. This means that on an annual basis (as seen in Figure 7-6), taking the annual cost into account, breakeven occurs somewhere in the third month of the fifth year of the solution being in production.

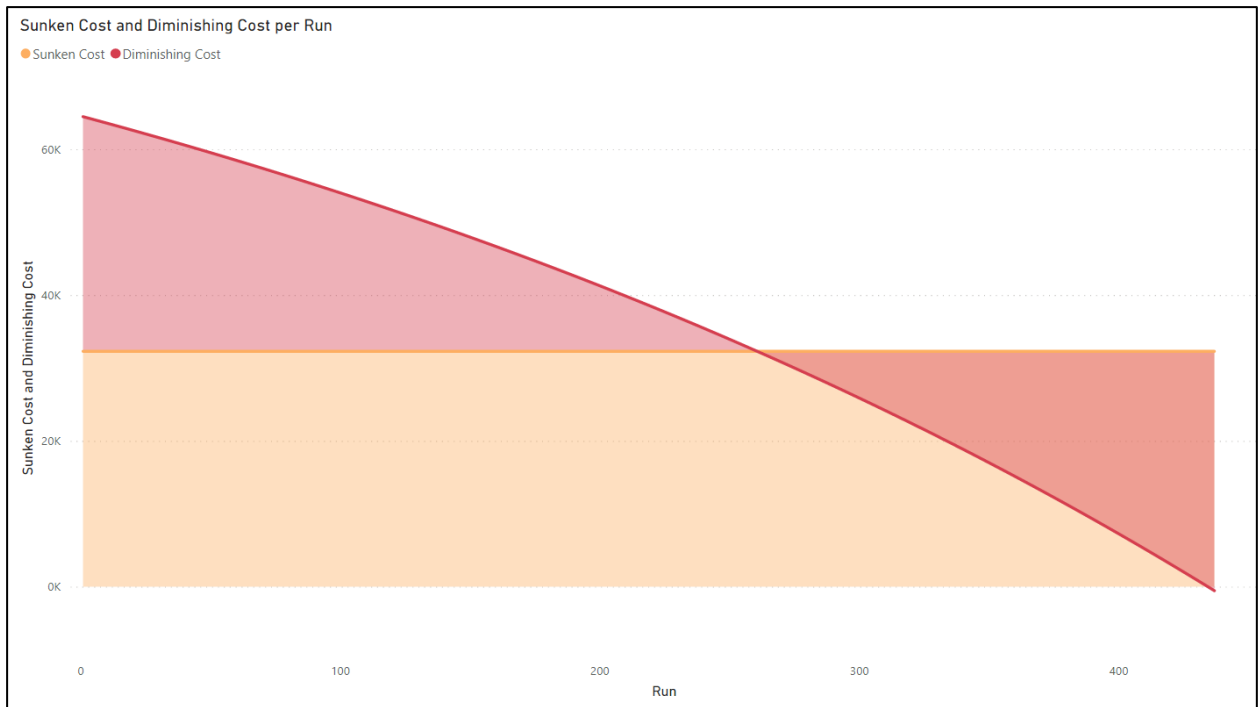


Figure 7-7: Sunken cost and diminishing cost per run

Figure 7-7 takes the 0.19% per run into account and shows that the breakeven point occurs on the 146th run. It can, therefore, be stated that the actual time saving of this solution is realised between runs 147 and 152.

7.3.1.3 Development cost: Second prototype

Development cost was not taken into consideration when calculating the direct cost of the DPA solution. This section serves to evaluate how development cost would be calculated if it were taken into consideration. The second prototype took 80 hours to develop and 20 hours to test. With the rate per hour of the developer and tester taken into consideration, the development (including testing) cost of the second prototype amounts to R16 140. In a case where development was outsourced, the rate per hour would be significantly more.

In a case where the development cost is not determined to be sunken cost, when calculating the number of runs that it would take for the costs to break even, the increase of cost per annum would need to be taken into consideration. This means that the cost per execution method should increase by approximately 10% per year to cater for the increase in direct and indirect costs of both execution methods.

To account for the annual inflation in cost and to calculate how long it would take for the initial development cost to be recovered, it was found that the present value formula would best fit the situation to calculate the time period that would lapse before a point of equilibrium is achieved.

$$P = A \times \frac{(1 + i)^n - 1}{i}$$

Where:

- P (Development cost) = R16 140.00
- A (Cost per year) = R4 978.48
- i (Annual percentage increase) = 0.10 (10%)
- n (Number of years)
- r (Number of runs) = $52n$

To calculate n , this formula is transformed into:

$$n = \log_{(1+i)}\left(\frac{Pi}{A} + 1\right)$$

$$n = \log_{(1.10)}\left(\frac{16\,140(0.10)}{4\,978.48} + 1\right)$$

$$n = \log_{(1.10)}(1.32)$$

$$n = 2.91$$

$$\therefore r = 52(2.91)$$

$$r = 151.32$$

By including a 10% increase in direct and indirect cost per year, the solution would need to be in production for approximately two years and 11 months (152 runs) before cost saving is realised. It can be seen in Figure 7-8 that a breakeven point is achieved towards the end of year two where the diminishing cost (which refers the declining difference in cost between execution methods and the development cost) breaks even with the fixed development cost while there is a significant actualised cost saving during years three and four.

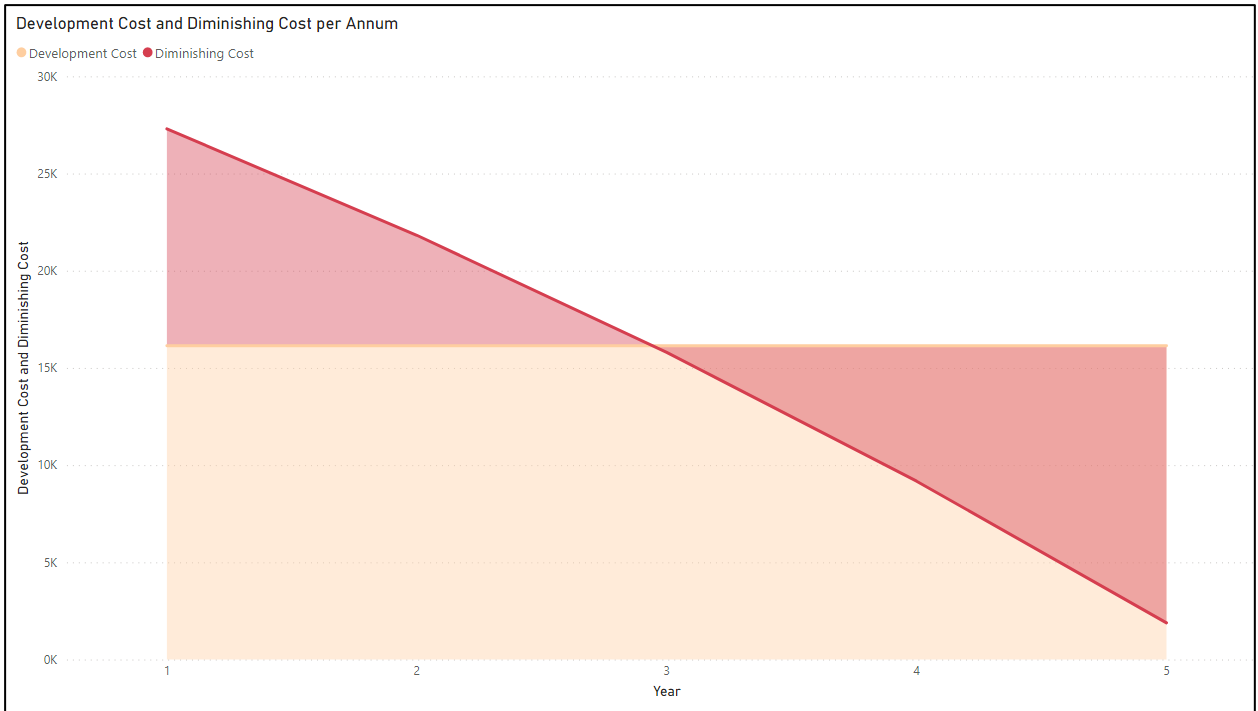


Figure 7-8: Development cost and diminishing cost per annum

Figure 7-9 shows that between runs 147 and 152 the diminishing cost (which refers the declining difference in cost between execution methods and the development cost) converges towards an equilibrium (or breakeven) point. The calculation above takes a 10% annual increase into account; however, there is an actual increase of 0.19% per run. This means that on an annual basis (as seen in Figure 7-8), taking the annual cost into account, breakeven occurs somewhere in the 11th month of the second year of the solution being in production.

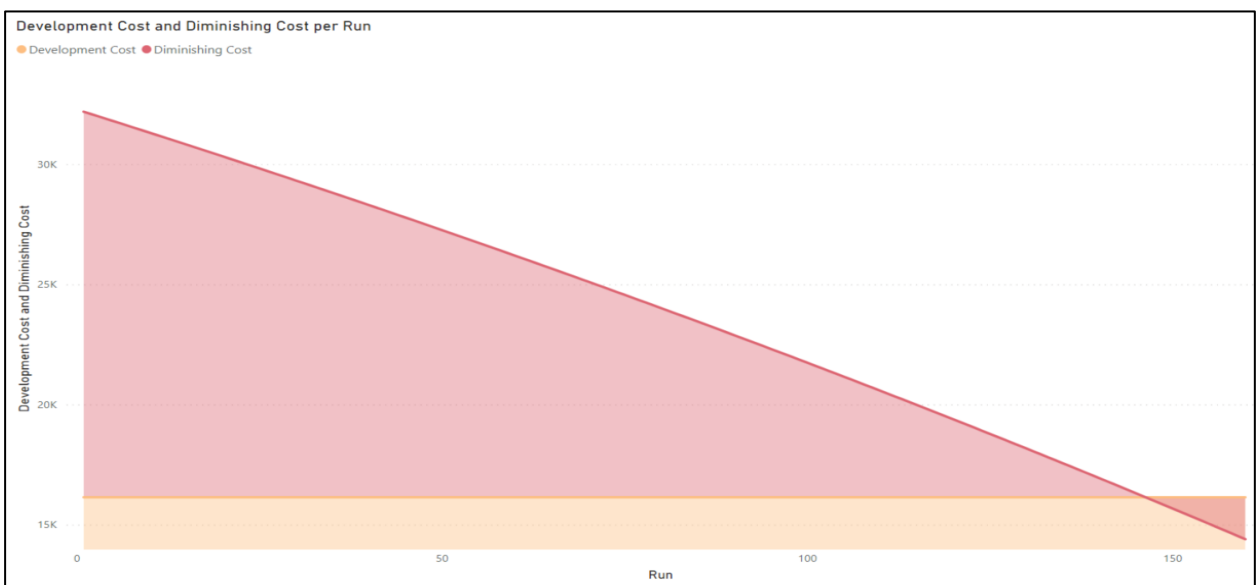


Figure 7-9: Development cost and diminishing cost per run

Figure 7-9 takes the 0.19% per run into account and shows that the breakeven point occurs on the 146th run. It can, therefore, be stated that the actual time saving of this solution is realised between runs 147 and 152.

7.3.1.4 Additional cost analysis

There are multiple different categories that cost can be broken into. These categories may be divided to be more granular depending on what is applicable within the designated use case. For the case study used within this study, the categories that make up additional cost are identified as requirements gathering cost, development cost and sunken cost.

Cost category	Number of runs to break even
Requirements gathering cost	544
Development cost	152
Sunken cost	273
Total	969

Table 7-4: Tabular representation of additional cost analysis

Based on Table 7-4, it can be seen that it would require 969 runs to break even with the cost of having an employee execute the process. In a case where the process is run once a week (52 times a year), it would take roughly 18 and a half years to breakeven. Due to the time saved by automating the process, the frequency of executing the process could be increased to run more times in a year. Table 7-5 shows the impact of increasing the number of runs in a year.

Frequency	Number of runs per year	Number of years to break even	Number of months to break even
Once a week, every week	52	18.63	224
Twice a week, every week	104	9.31	112
Three times a week, every week	156	6.21	75
Once a day, every day	365	2.65	32
Twice a day, every day	730	1.32	16
Three times a day, every day	1 095	0.88	11
Once every two hours, every day	2 190	0.44	5
Once every hour, every day	4 380	0.22	3

Table 7-5: Tabular representation of number of runs to break even

It can be seen in Table 7-5 that as the frequency increases, the measure of time (in terms of years and months) to the breakeven point (or equilibrium) decreases. Increasing the run frequency to anything more than twice a day, every day brings the breakeven point down to under two years after implementation. By increasing the process run frequency to once an hour, every hour (resulting in 4 380 runs per annum), the breakeven point would occur roughly three months after implementation.

In a case where the additional costs are included in the calculation, there is a short-term disadvantage to automating the process, but there is, however, still a long-term advantage to automating the process after the difference in cost has reached equilibrium. It is, therefore, advisable to increase the run frequency as far as possible and as far as value would be added. Considering the case study environment, libraries and packages can be updated at any time, which means that by increasing the frequency, the likelihood of upgrading processes as libraries are updated increases too. It would be feasible in the case study environment to increase the frequency from once a week, every week (52 runs per annum), to once every hour, every day (4 380 runs per annum).

7.4 Scope analysis

The average number of process steps gathered from the different execution methods has been measured on a 'per run' basis, but will also need to be evaluated on a 'per annum' basis. The

process is expected to run once a week, every week, resulting in 52 runs per annum. The summary of scope analysis for both methods of execution, as well as the comparison thereof are provided in Table 7-6.

	Number of steps per run		
	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Scope (<i>s</i>)	9	8	1
Total	9	8	1
	Number of steps per annum		
	Human execution (<i>H</i>)	DPA solution execution (<i>D</i>)	Difference between methods (<i>C</i>)
Scope (<i>s</i>)	468	416	52
Total	468	416	52

Table 7-6: Tabular representation of average number of steps per run

It can be seen in Table 7-6 that there is one step different between execution methods. This is due to the implementation of a database in the DPA solution, which allows for more efficient extraction and amendment of the base data. Although the DPA solution includes the optimisation of tasks when executing each process step, the employees remain more adaptable to change. The knowledge base that lives behind a DPA solution is defined by the developer and the business rules built into it, whereas a human does not have a predefined (or limited) knowledge base. This, therefore, indicates that a DPA solution’s ability to adapt is limited compared to that of a human, even though the DPA solution’s ability to execute the same process is cheaper and timelier than that of a human.

7.5 Discussion on analysis

This chapter contains the comparison of human execution and DPA execution of the same process in terms of time, cost and scope. The comparison was done on per run and per annum basis. Figure 7-10 shows the breakdown of how to collect and analyse data appropriately.

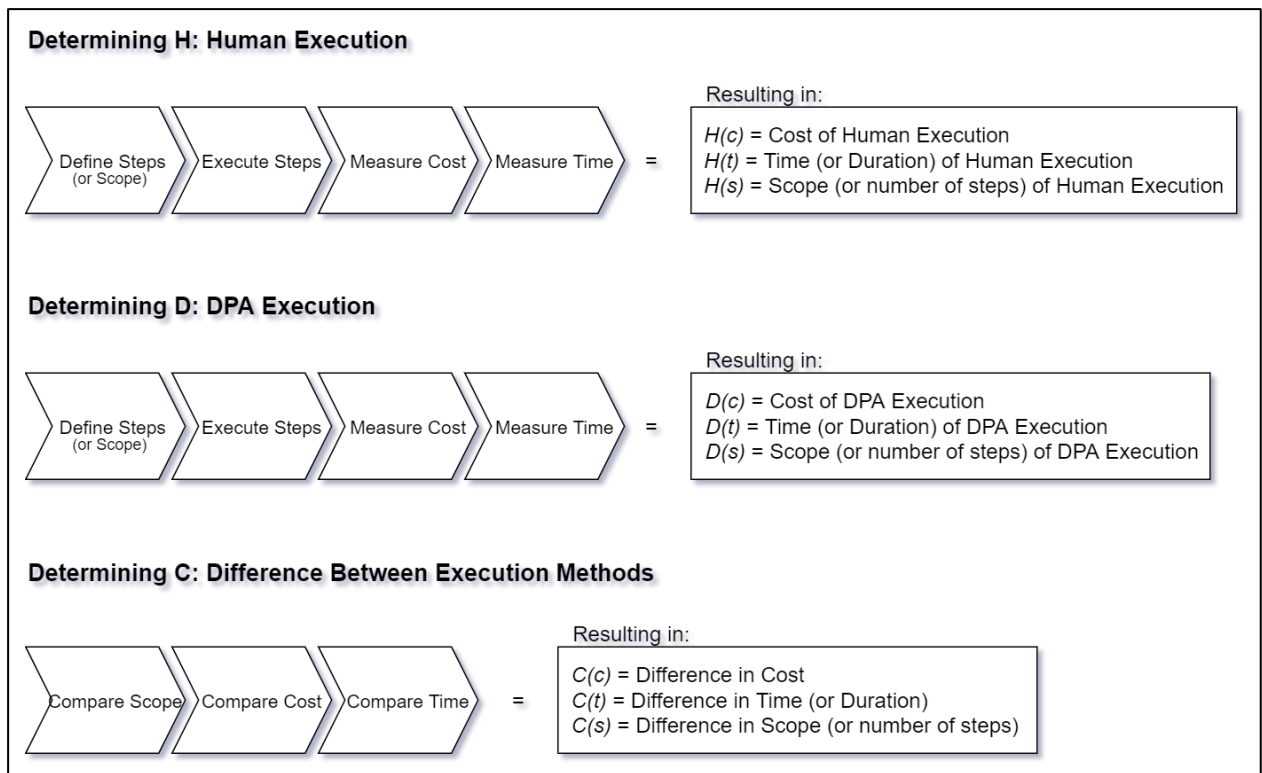


Figure 7-10: The determination of variables A, B and C in terms of X, Y and Z

As seen in Figure 7-10, collecting and determining each of these data points can be formalised into calculations. These calculations are summarised and expanded upon in Table 7-7 with regard to the following variable keys:

- H = Human execution of the process
- D = DPA solution execution of the process
- C = Difference between the execution methods
- c = Cost ($c_1 + c_2$)
 - c_1 = Direct Cost
 - c_2 = Indirect Cost
- t = Time ($t_1 + t_2 + t_3$)
 - t_1 = Process preparation time
 - t_2 = Process execution time
 - t_3 = Process termination time
- s = Scope (in terms of the number of steps)
- P = Proposed value
- R = Realised value

Calculations	Measurement per run		
	Cost (<i>c</i>)	Time (<i>t</i>)	Scope (<i>s</i>)
Human execution (<i>H</i>)	<i>H(c)</i> R97.82	<i>H(t)</i> 1 427	<i>H(s)</i> 9
DPA solution execution (<i>D</i>)	<i>D(c)</i> R2.08	<i>D(t)</i> 44	<i>D(s)</i> 8
Delta (<i>C</i>)	$C(c) = H(c) - D(c)$ R95.74	$C(t) = H(t) - D(t)$ 1 383	$C(s) = H(s) - D(s)$ 1
Proposed value (<i>P</i>)	$Let P(c) = C(c) > 0$ <i>1</i>	$Let P(t) = C(t) > 0$ <i>1</i>	$Let P(s) = C(s) \geq 0$ <i>1</i>
Realised value (<i>R</i>)	<p>$P(c) \rightarrow R(c)$, where <i>R(c)</i> represents a realised cost benefit of automation. The value of <i>R(c)</i> should either be 1 or 0 (true of false)</p> <p>$P(c) = 1 = R(c)$</p>	<p>$P(t) \rightarrow R(t)$, where <i>R(t)</i> represents a realised time benefit of automation. The value of <i>R(t)</i> should either be 1 or 0 (true of false)</p> <p>$P(t) = 1 = R(t)$</p>	<p>$P(s) \rightarrow R(s)$, where <i>R(s)</i> represents a realised scope benefit of automation. The value of <i>R(s)</i> should either be 1 or 0 (true of false)</p> <p>$P(s) = 1 = R(s)$</p> <p><i>A difference in scope does not contribute to realised value for the process used within the case study. The scope is therefore excluded from the determination of realised value.</i></p>
Realised value (<i>R</i>) calculation	$(R(c) \wedge R(t)) = (1 \wedge 1)$ <i>R = 1, therefore, there has been actual realised value of automating the human process</i>		

Table 7-7: Tabular representation of analytical calculations to be performed

Based on the findings in Table 7-7, it can be seen that the value of automating the process has been realised. The additional costs have been excluded from the calculations in Table 7-7, which results in short-term actual realised time saving. As explained in section 7.3.1, including additional costs into the calculation would result in a long-term actual time saving, which would be realised after approximately 969 process runs.

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influence time are process preparation time, process execution time and process termination time. Cost is influenced by direct and indirect cost. There was no direct cost associated to having a DPA solution execute the process; however, direct cost for human execution is influenced by the average duration of process execution and rate per hour (per employee). Indirect cost consists of infrastructure costs (specifically referring to hardware, support and licensing costs). The scope of the process execution is determined by the number of steps that are executed. The impact or saving that could be realised through analysing scope would be the measurement of quality, which was excluded from the scope of this study. Consistent execution leads to the consistent delivery of consistent quality.

7.6 Conclusion

The analysis done in this chapter takes into account the data collected in Chapters 4 and 6 in order to obtain the difference (or delta) between the execution methods. The analysis of the data provided understanding of the variability of factors like direct and indirect cost that make up execution cost. The introduction of additional costs offers a different perspective that might be beneficial in other situations (like case studies where development and requirements gathering would need to be outsourced). The division of time into phases helps understand the variability in execution time with possible influencing reasons, for example the number of process steps being iterated in a single run. The evaluation of scope impacting time and cost also helps identify how changes may be managed on future projects.

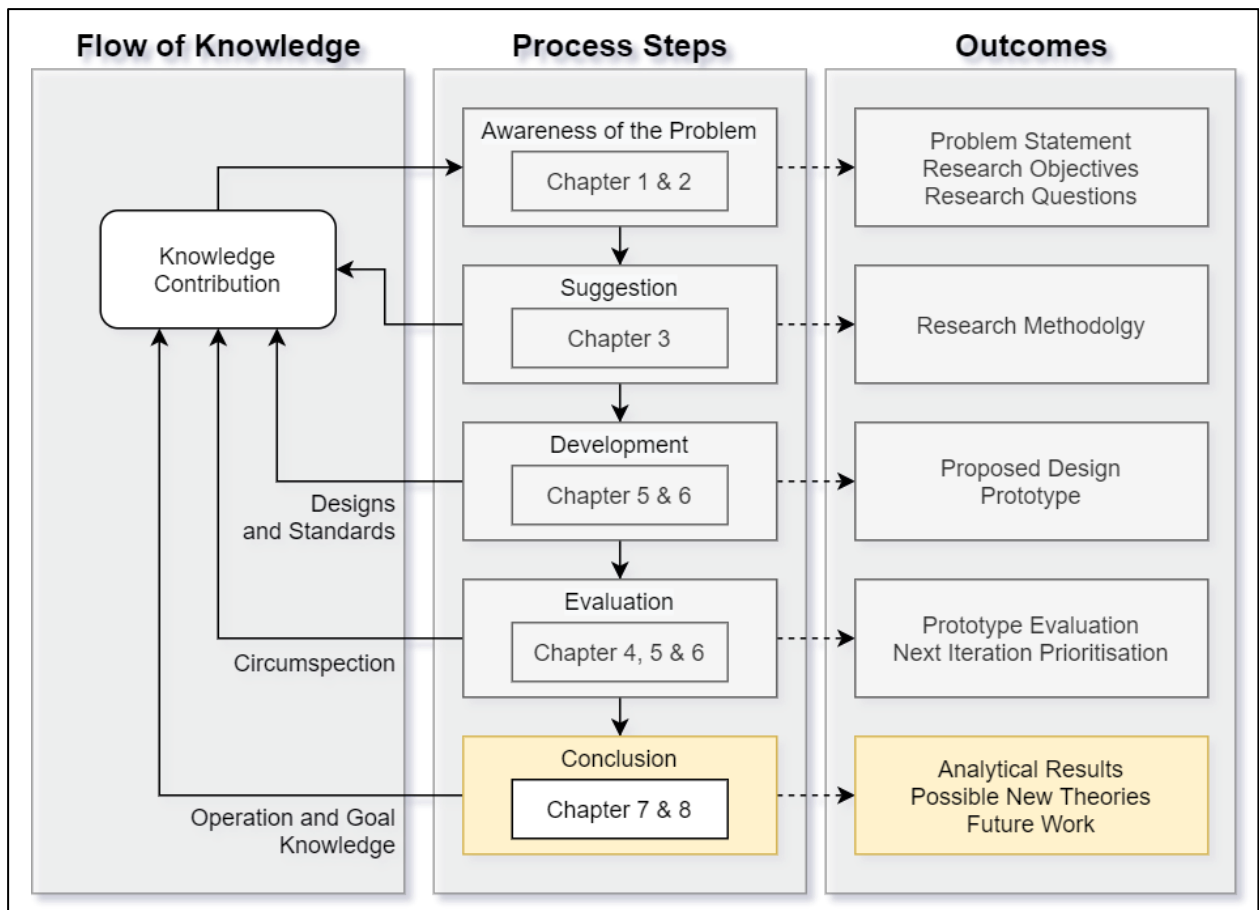


Figure 7-11: Chapter 7 DSR study layout contribution

This chapter provides analytical results on the study, which is in line with the conclusion process step of the DSR study layout, presented in Figure 7-11. The next chapter will conclude the study.

CHAPTER 8: CONCLUSION

8.1 Introduction

The aim of this research was to evaluate the difference of executing the same process between human execution and the execution of a digital process automation (DPA) solution, specifically in terms of cost, time and scope. Time, cost and scope make up the triple constraint model. The theory behind the triple constraint model, as presented by Baratta (2006), suggests that only two of the constraints may be optimised at a time and that each of the constraints impact one another. The discovery of these constraints forms part of the process analysis phase and the ability to accurately measure them once the automation is in place, allows the actual value of the automation to be realised. In order to measure the cost, time and scope, the DPA solution had to be built according to the steps outlined in the documentation. Once the solution was built, the following was obtained:

- The cost, scope and time of executing the DPA solution
- The cost, scope and time of human execution of the process
- The result of comparison between the different execution methods (in terms of cost, scope and time)

Each of the abovementioned aspects had multiple influencing factors, which may vary between case studies, but could be grouped through the use of similar terms or parameters. Identifying these parameters and how they can be evaluated was the focus of this study, applied within a case study as a proof of concept.

Sub-section 8.2 presents the summary of results with sub-section 8.3 presenting the summary of study, providing a summary of each chapter with specific reference to how the research objectives were met. Sub-section 8.4 focuses on the study limitations and future work with sub-section 8.5 concluding the study. This chapter contributes to the evaluation step of the DSR study layout as it presents the future work, possible theories and summarises the analytical findings.

8.2 Summary of results

The human execution and DPA solution execution of the same process were observed in order to collect the relevant quantitative data. The relevant business rules were applied during both methods of process execution and any additional business-specific information was recorded in Chapter 7.

The measurement of cost, time and scope (on average) is detailed in Table 8-1.

	Measurement per run (on average)		
Calculations	Cost (<i>c</i>)	Time (<i>t</i>)	Scope (<i>s</i>)
Human execution (<i>H</i>)	$H(c)$ R92.55	$H(t)$ 1 342 seconds	$H(s)$ 9 steps
DPA solution execution (<i>D</i>)	$D(c)$ R2.08	$D(t)$ 44 seconds	$D(s)$ 8 steps
Difference between methods (<i>C</i>)	$C(c) = H(c) - D(c)$ R90.47	$C(t) = H(t) - D(t)$ 1 298 seconds	$C(s) = H(s) - D(s)$ 1 step

Table 8-1: Tabular representation of collected and analysed data

The collected data was then compared and the variables that significantly impacted the study have been identified as follows:

- Time
 - Process preparation time
 - Process execution Time
 - Process termination time
- Cost
 - Direct cost (specific to human execution)
 - Rate per hour
 - Duration of process execution (in hours)
 - Indirect cost
 - Infrastructure cost
 - Hardware
 - Licensing
 - Support (DPA solution specific)
- Scope
 - Number of process steps

The analysis of the difference between execution methods determined that there is roughly a 21-minute time saving per process run, which equates to just under 19 hours per year. There is an approximate cost saving of R90.00 per run, which results in roughly R4 704.00 saving per year.

There is only a one-step difference between the execution methods. This is due to the implementation of a database in the DPA solution, which allows for more efficient extraction and amendment of the base data.

8.2.1 Time

The significant difference in time may be attributed to some of the following factors:

- Less wait time when switching between application screens.
- Reduced search and filter time when trying to match and join versions, packages and projects.
- Minimised risk of skipping a step in the process.
- Minimised risk of incorrectly entering version numbers.

8.2.2 Cost

Since the cost of the human execution is partially dependent on how long it takes an employee to execute the process, the factors mentioned above could also influence the difference in cost between execution methods.

In order to mitigate the difference in cost, focus should be placed on how to reduce the indirect cost of human execution. Indirect cost per employee within the case study environment amounts to approximately R140 692 per annum, with R108 692 allocated to licensing and R32 000 allocated to hardware. By decreasing licensing costs per employee, the cost difference between execution methods would decrease too. Although the difference per run may be small, the saving per annum would be more noticeable.

Additional factors that contribute to the difference in cost between execution methods include:

- Employees needing multiple developer licenses due to the wide scope of work that is addressed by the employee (for example, building Power BI reports, building web applications using ASP.NET Core and developing RPA solutions in UiPath. All three technology stacks have their own costs associated to them).
- The specifications of the hardware used by each employee also impact the price (for example, an 8th Gen i7 Dell laptop would cost more than a 7th Gen i5 laptop).
- Since development and operational support is excluded from the DPA solution execution direct cost calculation, the difference in direct cost is higher.

8.2.2.1 Additional costs

The additional costs (requirements gathering cost, sunken cost and development cost) were not taken into consideration when calculating the direct cost of the DPA solution. 800 hours were allocated to the requirements gathering and the development lifecycle of the DPA solution. The number of runs required to reach equilibrium is 969, including a 10% increase in direct and indirect costs per annum.

By investigating the impact on additional costs, it can be seen that the possibility of increasing the frequency of process runs could be an unrealised benefit until the change in frequency is implemented. The change in frequency is dependent on the feasibility of increasing the frequency as well as the additional value realisation of changing the frequency. The ideal frequency within the case study environment should be one run, once every hour, every day. This would result in 4 380 process runs per annum instead of 52. This leads to a shorter lapse in time between solution implementation and the breakeven point.

Based on the above, it can be seen that there is a short-term advantage to automating the process with additional costs excluded. In a case where additional costs are included in the calculation, there is a short-term disadvantage to automating the process, but there is, however, still a long-term advantage to automating the process after the difference in cost has reached equilibrium. An additional benefit to automation, which assists in mitigating the time lapse between implementation and breakeven point, is increasing the frequency of process runs.

8.2.3 Scope

Although the DPA solution includes the optimisation of tasks when executing each process step, the employees remain more adaptable to change. The knowledge base that lives behind a DPA solution is defined by the developer and the business rules built into it whereas a human does not have a predefined (or limited) knowledge base. This, therefore, indicates that a DPA solution's ability to adapt is limited compared to that of a human even though the DPA solution's ability to execute the same process is cheaper and timelier than that of a human. The number of steps executed in each process run has an impact on the time (and therefore the cost) of the process execution. The scope stability can be seen through the consistency in execution time per iteration through the process steps. A benefit of having a DPA solution execute the same steps repeatedly is consistency. The impact or saving that could be realised through analysing scope would be the

measurement of quality, which was excluded from the scope of this study. Consistent execution leads to the consistent delivery of consistent quality. The scope can become more flexible in cases where only certain portions of a solution is automatable, or in a case where the process is automated in phases.

8.3 Summary of study

The model proposed by Hevner *et al.* (2004) has been amended, as seen in Figure 8-1, to represent the position and layout that this study followed. Figure 8-1 illustrates how the development of theories and artefacts took in all environmental needs and requirements as inputs, utilising any applicable knowledge from the available knowledge base. The iterative design cycles were evaluated in each iteration, which outputted the environmental application and knowledge base additions to the environment and knowledge base, respectively. The relevance cycle represents the flow of all environmental needs and environmental application between the environment and the information system (IS) research. The rigour cycle represents the flow of knowledge between the knowledge base and the IS research.

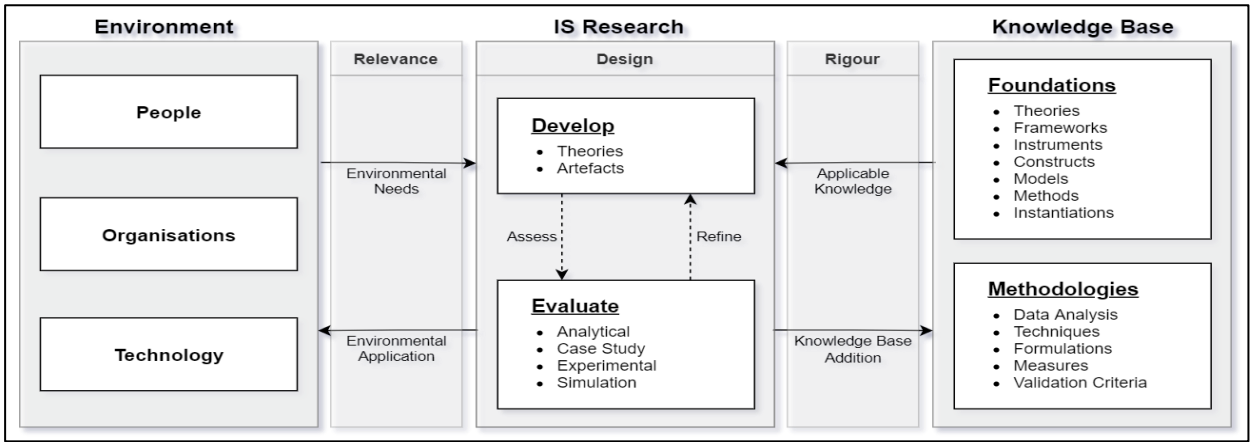


Figure 8-1: Design science research within the context of this study

Gerber *et al.* (2015) propose a model that illustrates the different process steps within DSR (along with the outputs for each step) as well as how each step contributes to the knowledge base. This model has been amended (as seen in Figure 8-2) to represent the contribution that each chapter had to the DSR process and the knowledge base, respectively.

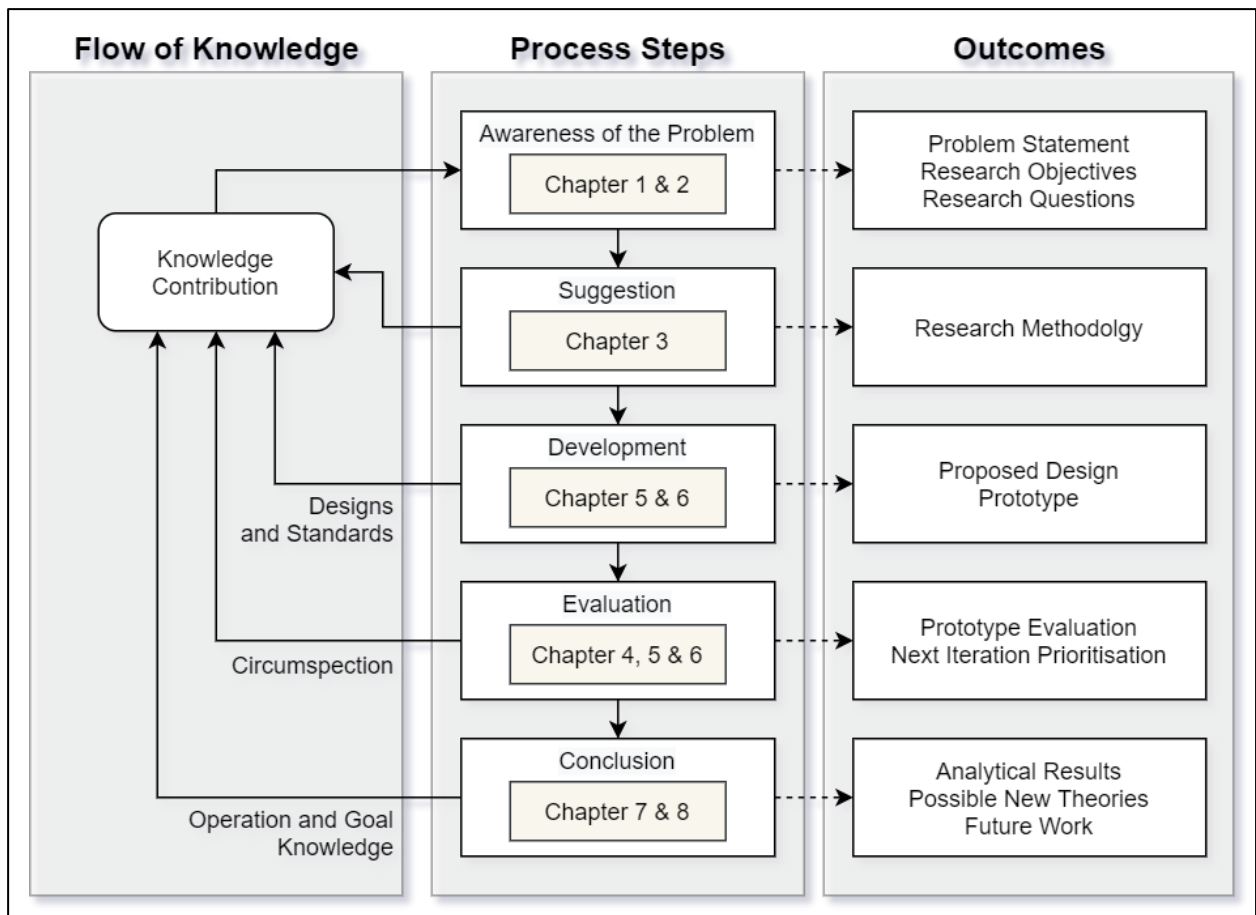


Figure 8-2: Chapter layout in terms of DSR

As seen in Figure 8-2, Chapters 1 and 2 created the context and awareness of the problem with Chapter 3 having suggested a methodology. Chapter 4 provided insight into the evaluation of the study. The development and evaluations occurred, iteratively, in Chapters 5 and 6. Chapters 7 and 8 concluded the research, presenting all results and findings.

8.3.1 Chapter 1: Introduction

This chapter started with the necessary background information regarding this study to explain the context within which the study took place. An explanation of the problem and dissertation statements was also provided in this chapter to explain the gap that was identified to motivate this research. The information provided in the problem statement section introduced the research questions and objectives, which led to the significance of the study. The limitations and assumptions of the study are also addressed in this chapter. The terms and concepts used in this study were defined and the chapter layout, in line with DSR process steps. The aim of Chapter 1 was to create awareness of the problem which extends into Chapter 2.

8.3.2 Chapter 2: Literature review

With automation and the triple constraint model central to this study, concepts like the automation evolution and lifecycle introduced the history and process of automation implementation, respectively. By understanding the process analysis and types of process documentation (along with their uses), the value of analysing a process properly became evident. It is important to understand the advantages and disadvantages of automation as many shortfalls of implementing automation can be avoided when process analysis is done thoroughly, bedding down the scope in the appropriate phase of the lifecycle. By clearly defining the scope, the variance of time and cost becomes more measurable and manageable. The context surrounding how scope, time and cost fit into the study forms the necessary awareness of the problem, which enabled the next step in the DSR model, presented in Figure 8-1, the suggested research methodology.

8.3.3 Chapter 3: Research methodology

The dissection of the research onion throughout the chapter positioned this study as having a pragmatic philosophy, an inductive approach, with experiments conducted within a case study focused strategy, while making use of DSR. The use of frameworks like SAFe and TOGAF catered for the iterative development of the artefact needed for data collection, in line with the purpose of the DSR design cycle. The research instruments defined enabled the collection of data, in the different forms that they were presented. The observation of human execution made use of task capture while the data collected from the DPA solution execution (including development and evaluation) used GAP analysis and JAD workshops. Once the data had been collected, it was analysed to provide the realised saving. Ethical considerations as well as validity and reliability were used and evaluated throughout this study. The output of this chapter was aligned to the 'suggested methodology' proposed by the DSR study layout.

8.3.4 Chapter 4: Human execution data collection

This chapter focused on the collection of the cost, time and scope variables in preparation for comparison with the variables gathered while observing the execution of the DPA solution. Three employees who work within the case study environment were used as participants of the study in order to collect the relevant data. The gathered data has been presented as a 'per run'

measurement as well as on a 'per annum' measurement in this chapter in order to compare the different execution methods effectively.

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influence time are process preparation time, process execution time and process termination time. These three phases equate to the total duration (or time) of having a human execute the process. The duration had to be calculated in hours to cater for the cost calculation.

From the collected data, it can be seen that cost was influenced by direct and indirect cost. Direct cost was affected by two parameters, namely the time it takes each employee to execute the process (in hours) as well as the rate per hour. Indirect cost consisted of infrastructure costs (specifically referring to hardware and licensing costs).

The scope of the process execution was determined by the number of steps that were executed. The difference in the duration of each employee executing the process may be justified by how each step was executed by the employees, but ultimately the number of process steps remains the same between each of the 30 runs.

This chapter evaluated the factors that contribute to the time, cost and scope of a human executing the process, in line with the DSR study outline.

8.3.5 Chapter 5: Initial conceptual design and build

Based on the need for an artefact to be developed for the data collection portion of this study, the outcome of this chapter presented the initial conceptual design. The first iteration through TOGAF assisted in establishing the organisational environment, along with the business strategy and business model, with relation to the operational model. The operational model provided more context into the case study environment where several gaps became apparent and was addressed through the next iterative cycle. Based on the gap analysis, incorporating the automation preparation and solution design phases of the delivery model into PI planning assisted in closing the identified gaps. The application architecture detailed the process that needed to be developed and although all the steps have been defined, the next iteration needed to include detailed steps to better inform the application architecture. The technology architecture relied on the existing automation ecosystem and was designed and implemented to compliment what already exists. In this chapter, the initial design and build were developed and evaluated, in line with the DSR study layout.

8.3.6 Chapter 6: Proposed conceptual design and build

Based on the need for an artefact to be developed for the data collection portion of this study, the outcome of this chapter was the proposed design and prototype. The second iteration through TOGAF assisted in establishing the updated organisational environment, along with the business strategy and business model, with relation to the organisation's operational model. This iterative cycle has rendered the observation that there are no longer any gaps that exist within the case study environment. The application architecture was refined with more detail in this chapter, along with the data architecture and technology layer.

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influence time were process preparation time, process execution time and process termination time. These three phases equate to the total duration (or time) of having a DPA solution execute the process.

From the collected data, it can be seen that cost was influenced by direct and indirect cost. There was no direct cost associated to having a DPA solution execute the process; however, indirect cost consisted of infrastructure costs (specifically referring to hardware, support and licensing costs).

The scope of the process execution was determined by the maximum number of steps that can be executed within a process run.

In this chapter, the proposed design and build were developed and evaluated, in line with the DSR study layout.

8.3.7 Chapter 7: Data analysis

The analysis done in this chapter took into account the data collected in Chapters 4 and 6 in order to obtain the difference (or delta) between the execution methods. The analysis of the data provided understanding of the variability of factors like direct and indirect cost that made up execution cost. The introduction of additional costs offered a different perspective that might be beneficial in other situations (like case studies where requirements gathering and development would need to be outsourced or included). The division of time into phases helped understand the variability in execution time with possible influencing reasons, for example the number of process steps being iterated in a single run. The evaluation of scope impacting time and cost also

helped identify how changes may be managed on future projects. The scope can become more flexible in cases where only certain portions of a solution is automatable, or in a case where the process is automated in phases. This chapter provided analytical results on the study, which is in line with the conclusion process step of the DSR study layout.

8.3.8 Research objectives

The research objectives of this study were divided between primary and secondary objectives. The secondary objectives were identified as theoretical and empirical objectives.

8.3.8.1 Primary objective

The primary objective of this study was to identify and evaluate the parameters that influence the scope, time and cost between human and DPA solution capability in terms of a predefined process. The parameters were identified between Chapters 4, 6 and 7. It can be seen that the factors that influence time, cost and scope of both execution methods may be generalised with differing underlying factors for each method.

Different factors contribute to indirect cost between execution methods, but on a level up, indirect cost contributes to the cost of both execution methods. Indirect cost for DPA solution execution only took infrastructure costs into consideration, whereas human execution also included licensing. In the same way, direct cost for DPA solution may have included additional costs, whereas additional cost was not applicable to the human execution of a process as human execution took the rate per hour into account instead (which is excluded from the DPA solution executed).

The division of time into phases was broken up according to the scope and environment. In terms of the case study, the three phases were generalised as the preparation phase, the execution phase and the termination phase. The division of process steps into phases helped evaluate time and cost of scope on a more granular. It became easier to identify trends in the phases that have a longer duration, allowing for more understanding to be gathered to answer 'why' there was a difference.

8.3.8.2 Secondary objectives

The secondary objectives of this study were an expansion of the primary objective based upon the comparative results of the primary objective.

The focus of the theoretical objectives was to gain understanding of and insight into critical variables identified for this study.

- **To identify the influencing parameters and measure the time it takes a person to execute the process:**
 - The parameters that contributed to the time it takes a human to execute a process, in this study, were focused on the division of phases, namely the preparation phase, the execution phase and the termination phase.
 - The time it takes each employee to execute each process run was observed and recorded in Chapter 4, in terms of the phases defined in the abovementioned point.
- **To identify the influencing parameters and measure the time it takes the DPA solution to execute the process:**
 - The parameters that contribute to the time it takes the DPA solution to execute a process, in this study, were focused on the division of phases, namely the preparation phase, the execution phase and the termination phase.
 - The time it takes the DPA solution to execute each process run was observed and recorded in Chapter 6, in terms of the phases defined in the abovementioned point.
- **To identify the influencing parameters and measure the cost of a person execute the process:**
 - The parameters that influenced the cost of a person executing the process need were identified in Chapter 4 as direct and indirect cost. The direct cost took the employee's rate per hour as well as the average time it takes an employee to execute the process into consideration. Indirect cost took licensing and infrastructure cost into consideration.
 - Measuring the cost of a person executing the process was also done and recorded as part of Chapter 4.
- **To identify the influencing parameters and measure the cost of the DPA solution executing the process:**
 - The parameters that influence the cost of the DPA solution executing the process need were identified in Chapter 6 as direct and indirect cost. The direct cost for this study was 0; however, additional costs (like requirements gathering, development

and sunken cost) may be taken into consideration as part of direct cost. The additional costs were detailed and analysed in Chapter 7. The indirect cost took the infrastructure cost into consideration.

- Measuring the cost of the DPA solution executing the process was also done and recorded as part of Chapter 6. More analysis was presented in Chapter 7.
- **To identify the influencing parameters and measure the scope of a person executing the process:**
 - The scope of human execution does not have any predefined contributing factors in this study.
 - The measurement of this variable was evaluated as the number of steps in a process, in Chapter 4.
- **To identify the influencing parameters and measure the scope of a DPA solution executing the process:**
 - The scope of DPA solution execution does not have any predefined contributing factors in this study.
 - The measurement of this variable was evaluated as the number of steps in a process, in Chapter 6.

The focus of the empirical objectives was to generate the information that was needed in order to execute the study.

- **To acquire the quantitative data**
 - The quantitative data needed to measure the above six variables was successfully gathered, as seen in Chapters 4 and 6.
- **To compare the acquired data**
 - The data was compared in Chapter 7, presenting the difference between time, cost and scope of the different execution methods.
- **To identify the significant variables that have an effect on the study**
 - The variables that impact this study are detailed as part of the theoretical objectives above as well as in Chapters 4, 6 and 7.
- **To apply business rules to the comparison in order to make informed decisions based on the comparative analysis**
 - As part of the conceptual design and build chapters, a report was built to visualise and understand the state of the process. Informed decisions may be based off of these reports, in addition to the analysis presented in Chapter 7.

8.3.9 Research questions

The research questions were broken up into primary and secondary research questions. The primary research questions, in line with the primary research objectives aimed to identify factors and parameters that influenced the time, cost and scope between execution methods. The secondary research questions aimed to support the primary research questions by providing the context derived from the case study.

Primary research questions	Research answers	Reference
What parameters influence how long it takes a human to execute the process?	The duration of a human executing a process can be split up into different phases. In this study, the phases identified and measured were preparation phase, execution phase and termination phase.	Sub-section 4.2.1
What parameters influence how long it takes a DPA solution to execute the process?	The duration of the DPA solution executing a process can be split up into different phases. In this study, the phases identified and measured were preparation phase, execution phase and termination phase.	Sub-section 6.7.1
What parameters influence how much it costs having a human to execute the process?	The cost of a human executing a process was broken up into two categories, namely indirect cost and direct cost. Direct cost was influenced by parameters like the number of employees executing the solution, the time it takes each employee to execute the process as well as each employee's rate per hour. Indirect cost was influenced by the number of employees executing a process, the number of working hours per annum, infrastructure costs (hardware costs and licensing costs) as well as the execution time per employee.	Sub-section 4.2.2
What parameters influence how much it costs having a DPA solution to	The cost of the DPA solution executing the process was also broken up into direct and indirect cost. Indirect cost was influenced by infrastructure costs. In this study, the direct cost was 0 as direct labour was not needed during	Sub-section 6.7.2 and sub-section 7.3.1

execute the process?	the execution of the process when the DPA solution is executing the process. In other case studies, the direct cost may be substituted by additional costs, like requirements gathering development and sunken costs. For this study, additional costs were calculated to show how it could affect total cost.	
What parameters determine the scope of the human's execution of the process?	Scope was measured by the number of process steps that it takes to execute the process. In order to understand the variance in the time it takes each employee to execute the process, the actions that lead to the variance would need to be evaluated. In this study, the following aspects were measured to understand the scope variations of human execution: number of windows used, number of actions executed, number of mouse clicks executed, number of keyboard keys pressed, the number of text entries, the number of hotkeys used, the number of tasks executed and the number of processes executed.	Sub-section 4.2.3
What parameters determine the scope of the DPA solution's execution of the process?	Scope was measured by the number of process steps that it takes to execute the process. These steps are outlined in Figure 6-3.	Figure 6-3 and sub-section 6.7.3
Secondary research questions	Research answers	Reference
How long (on average) does it take a human to execute the process?	On average, it took approximately 23 minutes and 47 seconds (per run) to execute the process manually.	Table 4-3 and sub-section 7.5

How long (on average) does it take the DPA solution to execute the process?	On average, it took approximately 44 seconds (per run) for the DPA solution to execute the process.	Table 6-8 and sub-section 7.5
Is there an actual realised benefit of time saving by automating the process?	There was a saving of approximately 23 minutes (per run) of having the DPA solution execute the process.	Table 7-2 and sub-section 7.5
How much does it cost for a human to execute the process?	On average, it costed approximately R99.53 (per run) to execute the solution manually.	Table 4-4, Table 4-5 and sub-section 7.5
How much does it cost for the DPA solution to execute the process?	On average, it costed approximately R2.08 (per run) to have the DPA solution execute the process.	Table 6-9 and sub-section 7.5
Is there an actual realised benefit of cost saving by automating the process?	There was a short-term saving of approximately R95.74 (per run) of having the DPA solution execute the process (excluding additional costs). By including the additional costs, breakeven only occurred in roughly 969 runs. If the process was executed 52 times a year (once a week, every week), it would have taken approximately 18 years to breakeven. If the frequency was increased to 4 380 per year (once an hour, every day), the breakeven point would have occurred approximately three months after implementation. After the breakeven point occurred, an annual cost saving would have been realised. The underlying benefit of considering additional costs was adapting the frequency of process runs.	Table 7-3 and sub-section 7.5

What is the scope of the human's execution of the process?	The process consisted of nine process steps that were detailed in Figure 4-1.	Figure 4-1, Table 4-6 and sub-section 7.5
What is the scope of the DPA solution's execution of the process?	The DPA solution executed an average of eight process steps to complete the execution of the process. These steps were outlined in Figure 6-3.	Sub-section 7.5 and sub-section 6.3.1.3, with specific reference to Figure 6-3
Is there an actual realised benefit of scope saving by automating the process?	Although there was a difference of one in the number of steps executed between processes, the largest benefit of the DPA solution executing the same steps repeatedly was consistency. The benefit of having one step less would be bundled with the impact on time or cost that executing the extra step would have (which was also dependent on what the step was and whether it made sense in the process that the DPA solution executes. In this study, the extra step was already accommodated for by the rest of the process steps).	Sub-section 7.5

Table 8-2: Tabular representation of the primary and secondary research questions

Table 8-2 detailed the primary and secondary research questions alongside the answers to each question as well as references to support each answer provided in the table.

8.4 Limitations and future work

The work presented in this study may be expanded on in multiple different ways with focus placed on different aspects. Some of these aspects include:

- Expand the scope of this study to calculate the return on investment (ROI) per automation:
 - o The most important aspect of calculating ROI would be to identify the factors that would influence ROI.
 - o The validity of the ROI calculation should be measured by ensuring that all aspects identified above have been appropriately accounted for and that there are no other

factors that could dramatically impact the ROI. A challenge that could potentially arise with this is quantifying qualitative factors, for example quality.

- Focus on enhancing the power BI report with more detail:
 - o Adding more detail around exceptions and errors would allow the operational support engineers to identify what the errors and exceptions were during specific runs, without having to deep dive through log files.
 - o It would also be beneficial to add visualisations of historical data to the report in order to identify trends and possible projections.
 - o The report could also be extended to visualise the appropriate benefits of each process (or automation solution), in terms of time, scope, cost and ROI. These values could be rolled up to determine the net realised benefits and savings.
- A test framework that uses AI models to infer which packages need upgrading and which projects would work could be added to the solution.
 - o The aim of this would be to train the model to infer what the process configuration should be, based on what systems, integrations and technologies are used within a solution.
 - o For example, any process that interacts with a tier 1 system should be deemed business critical, whereas interactions with tier 2 systems that have a QA environment, should not be classified business critical.
 - o This could even become more granular depending on what functionality the integration makes use of, in terms of create, read, update and delete (CRUD). Solutions that only read data, may be considered a lower risk to execute than processes that create, update or delete records in tier 1 or tier 2 systems.
- Focus on measuring more elements of a project, for example placing more emphasis on quality assurance and quality measurement between solutions.
 - o Traditionally, the triple constraint model focuses on the restrictions placed on project cost, time (or schedule) and scope. The more modern approach has seen this transform into the project management diamond, which includes quality as an additional factor. Since the execution of automation solutions is assumed to be consistent, due to the consistency of scope, it would be interesting to evaluate the relationship between scope and quality.
 - o It would also be interesting to identify ways in which quality can be accurately measured and assured.
- Execute the process in a different environment, with different project management methodologies:

- Instead of designing and developing the solution in an agile environment, it would be interesting to see if a waterfall (or possibly even a hybrid) project management methodology would have an impact in development cost.
- Expanding the scope of the process to include other solution types, for example updating libraries used in .NET and .NET Core projects.
 - Different platforms use different mechanisms that manage package/library usage. These mechanisms will need to be catered for in different ways, but could still be plugged into the spine of the solution as modular components. Instead of rebuilding the solution for each platform, it would be interesting to identify how building specific plug-ins would impact the solution.
- Add a minimisation of cost model that shows the next-best cost-effective option and measure the impact on execution time.
 - The factors that influence cost referenced in this study are enough to start a basic cost minimisation model, although it would be beneficial to understand whether or not there are other factors that could contribute to minimising cost.
 - In a client facing environment, this concept could be expanded to maximising profit, while still minimising cost.

8.5 Conclusion

Time can be broken down into different phases, which may be specifically determined by predefined criteria. In this case, the parameters that influenced time were process preparation time, process execution time and process termination time. Cost was influenced by direct and indirect cost. There was no direct cost associated to having a DPA solution execute the process; however, direct cost for human execution was influenced by the average duration of process execution and rate per hour (per employee). Indirect cost consisted of infrastructure costs (specifically referring to hardware, support and licensing costs). The scope of the process execution was determined by the number of steps that were executed.

Based on the analytical findings of this study, there was an actual realised time and cost saving of automating the auto-package upgrade process that was previously executed manually. There was a short-term and long-term cost saving where additional costs were excluded from the comparison; however, if additional costs were included, there would only have been a long-term cost saving benefit, which would have been realised once the cost saving and additional costs break even. An underlying benefit of realising the cost benefit would be to increase process frequency from once a week to once an hour. The scope aspect of a solution tends to relate to

the ability to be adaptable. A human retains the advantage of being able to adapt when necessary, whereas the DPA solution would need any additional actions to be developed and/or added to its knowledge base. The quality of the solution would remain constant due to the predefined steps that the process will consistently execute, based on the business logic built into the solution. The consistent quality of the DPA solution is an additional, unintended benefit.

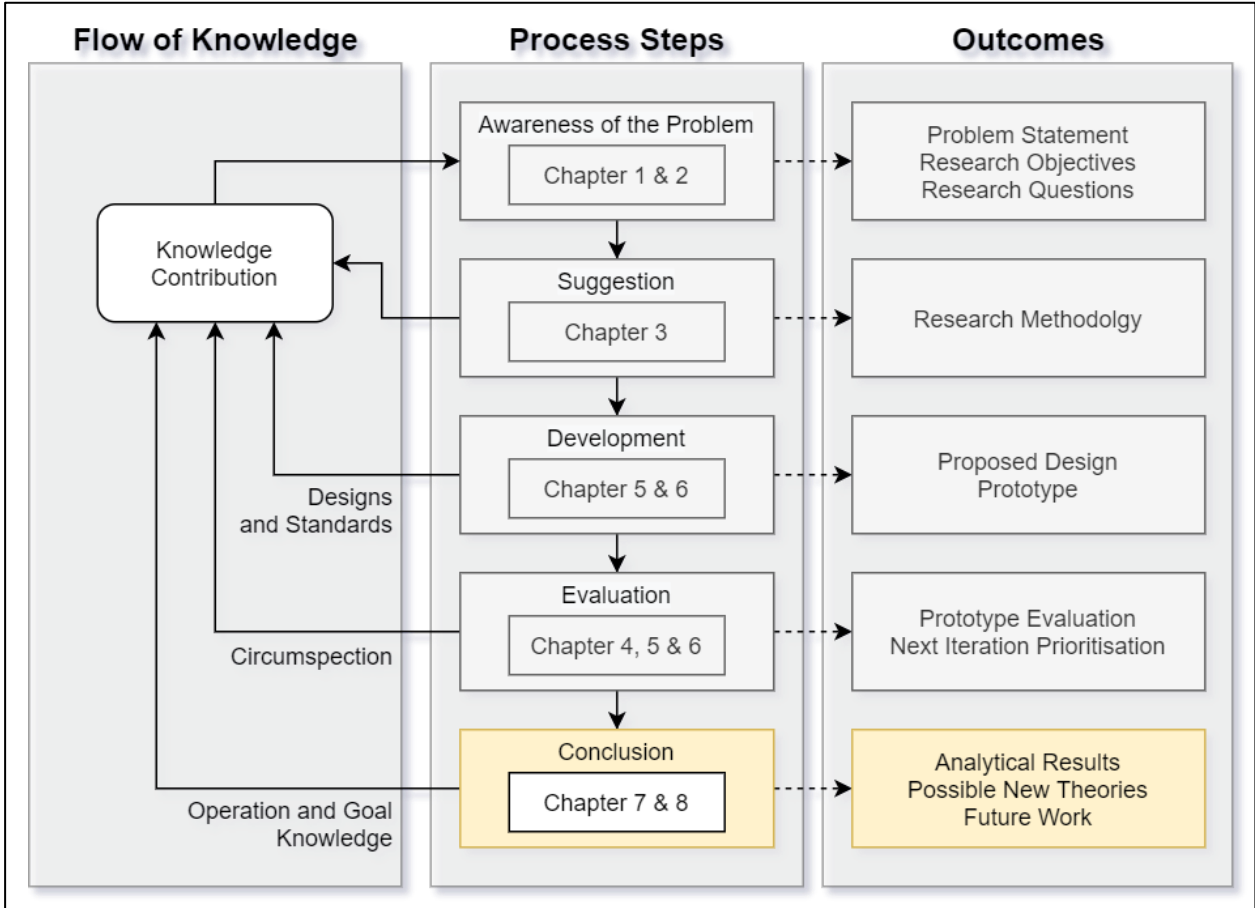


Figure 8-3: Chapter 8 DSR study layout contribution

This chapter closed off the last process step in the DSR study layout (as seen in Figure 8-3) as the analytical findings, possible new theories and future work were presented throughout the chapter.

REFERENCES

- Acemoglu, D. & Restrepo, P. 2019. Automation and new tasks: How technology displaces and reinstates labor. *Journal of Economic Perspectives*, 33(2):3-30.
- Ackerman, B., Lesko, C.R., Siddique, J., Susukida, R. & Stuart, E.A. 2021. Generalizing randomized trial findings to a target population using complex survey population data. *Statistics in Medicine*, 40(5):1101-1120.
- Ageling, W. 2019. SAFe (Scaled Agile Framework): An introduction. <https://medium.com/serious-scrum/safe-scaled-agile-framework-an-introduction-b30bae91ebf6>
Date of access: 27 May 2020.
- Ahmad, M.O., Dennehy, D., Conboy, K. & Oivo, M. 2018. Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137:96-113.
- Anguera, M.T., Portell, M., Chacón-Moscoso, S. & Sanduvete-Chaves, S. 2018. Indirect observation in everyday contexts: concepts and methodological guidelines within a mixed methods framework. *Frontiers in psychology*, 9:13.
- Arifin, S.R.M. 2018. Ethical considerations in qualitative study. *International Journal of Care Scholars*, 1(2):30-33.
- Asatiani, A., Kämäräinen, T. & Penttinen, E. 2019. Unexpected problems associated with the federated IT governance structure in robotic process automation (RPA) deployment. *Aalto University Publication Series*
- Asiamah, N., Mensah, H.K. & Oteng-Abayie, E.F. 2017. General, target, and accessible population: Demystifying the concepts for effective sampling. *The Qualitative Report*, 22(6):1607.
- Auth, G., Czarnecki, C. & Bensberg, F. 2019. Impact of robotic process automation on enterprise architectures. Paper presented at the INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik–Informatik für Gesellschaft (Workshop-Beiträge).
- Balic, T. & Ebrahimi, H. 2017. Automation and digital transformation: The ways that automation solutions can support digital transformation within ICT companies. Gothenburg University.
- Baratta, A. 2006. The triple constraint: a triple illusion. Paper presented at the PMI Global Congress Proceedings, Seattle, Washington.
- Bashir, M., Afzal, M.T. & Azeem, M. 2008. Reliability and validity of qualitative and operational research paradigm. *Pakistan journal of statistics and operation research*:35-45.

Behzadifard, S. 2020. Strategic alignment analysis of business processes based on discovering process models with a hybrid approach.

Bekemeier, B., Delaney, K., Wenzl, S., Roberts, M. & Hersh, D. 2020. The Diamond Project: A quality improvement model for adopting shared service delivery in the Washington vaccines for children program. *Frontiers in public health*, 8:272.

Blevins, T.J., Spencer, J. & Waskiewicz, F. 2004. TOGAF ADM and MDA. *The Open Group and OMG*. <https://www.opengroup.org/cio/MDA-ADM/MDA-TOGAF-R1-070904.pdf> Date of access: 15 June 2021.

Boterman, W. 2021. Quantitative data collection: A meta view. *Seeing the city*. Amsterdam University Press. p. 22-36).

Brown, S.W. & Swartz, T.A. 1989. A gap analysis of professional service quality. *Journal of marketing*, 53(2):92-98.

Business Jargons. 2017. Process analysis. <https://businessjargons.com/process-analysis.html> Date of access: 17 May 2020.

Carmel, E., Whitaker, R.D. & George, J.F. 1993. PD and joint application design: A transatlantic comparison. *Communications of the ACM*, 36(6):40-48.

Circle, V.P.C. 2018. TOGAF ADM introduction. <https://circle.visual-paradigm.com/docs/togaf-adm-guide-through/togaf-adm-introduction/> Date of access: 13 January 2021.

Corbin, J. & Strauss, A. 2014. *Basics of qualitative research: Techniques and procedures for developing grounded theory*: Sage Publications.

Creswell, J.W. & Clark, V.L.P. 2017. *Designing and conducting mixed methods research*: Sage Publications.

Croson, R., Anand, J. & Agarwal, R. 2007. Using experiments in corporate strategy research. *European Management Review*, 4(3):173-181.

De Weerd, J., De Backer, M., Vanthienen, J. & Baesens, B. 2012. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Information Systems*, 37(7):654-676.

Driscoll, D. & Brizee, A. 2012. Ethical considerations in primary research. *Retrieved September*, 29:2013.

Francis, J.J., Johnston, M., Robertson, C., Glidewell, L., Entwistle, V., Eccles, M.P. & Grimshaw, J.M. 2010. What is an adequate sample size? Operationalising data saturation for theory-based interview studies. *Psychology and health*, 25(10):1229-1245.

Frost & Sullivan. 2019. The intelligent RPA revolution with process discovery. http://kryonsystems.com/Documents/The_Intelligent_RPA_Revolution_With_Process_Discovery_WhitePaper.pdf?hsCtaTracking=6ce3d7ca-d231-453e-9271-a33ad507da22%7Ce1eda6a3-e27a-46d1-b046-f39aea27ca08 Date of access: 26 May 2020.

Gaikwad, P.K., Jayakumar, C.T., Tilve, E., Bohra, N., Yu, W. & Spichkova, M. 2019. Voice-activated solutions for agile retrospective sessions. *Procedia Computer Science*, 159:2414-2423.

Garavaglia, J.A. 2009. Full automation in live-electronics: advantages and disadvantages. Paper presented at the International conference on auditory display (ICAD)

Gerber, A., Kotze, P. & Van der Merwe, A. 2015. Design science research as research approach in doctoral studies. Paper presented at the AMCIS 2015 Proceedings, Puerto Rico.

Gonçalves, L. 2018. Scrum. *Controlling & Management Review*, 62(4):40-42.

Graves, T. 2013. On layers in enterprise-architecture. <http://weblog.tetradian.com/2013/12/11/on-layers-in-ea/> Date of access.

Groover, M.P. 2019. Automation. <https://www.britannica.com/technology/automation> Date of access.

Gunesh, R. 2016. Guidelines on research. <https://www.rajgunesh.com/resources/methodology.htm> Date of access: 22 August 2020.

Harrison, H., Birks, M., Franklin, R. & Mills, J. 2017. Case study research: Foundations and methodological orientations. *Forum: Qualitative Social Research*, 18(1).

Haruna, K., Akmar Ismail, M., Suhendroyono, S., Damiasih, D., Pierewan, A.C., Chiroma, H. & Herawan, T. 2017. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 7(12):1211.

Henreaux, E., Noutcha, M., Phan-Ngoc, T. & Suzanne, K. 2021. Design sprints integrating agile and design thinking: A case study in the automotive industry. (*In*. International Conference on Applied Human Factors and Ergonomics organised by: Springer. p. 189-195).

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

Hevner, A.R. 2007. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4.

Hibbard, K.A., Meehl, G.A., Cox, P.M. & Friedlingstein, P. 2007. A strategy for climate change stabilization experiments. *Eos, Transactions American Geophysical Union*, 88(20):217-221.

Hobbs, K. 2015. A beginner's guide to business process analysis
<https://www.excella.com/insights/a-beginners-guide-to-business-process-analysis> Date of access: 17 May 2020.

House, B.C. 2019. Process automation technologies.
<https://businessconsultinghouse.de/en/process-automation-technologies/> Date of access: 25 May 2020.

Ilyin, I.V., Grigoreva, A.A. & Zapivakhin, I.M. 2017. Architectural solution as a tool for planning and approval of changes in projects for information systems implementation and customization. *Information systems and technologies in business*(2):68.

Iyawa, G., Botha, A. & Herselman, M.E. 2016. Identifying and defining the terms and elements related to a digital health innovation ecosystem. *Strategies, Approaches and Experiences: Towards building a South African Digital Health Innovation Ecosystem*. p. 30-54).

Izzo, M., Harrap, J., Donelan, P., Chen, B. & Davidson, N. 2002. Software design document. *Event Driven DIS PDU Logger (EDDIS system)*.
[http://portal.unimap.edu.my/portal/page/portal30/Lecture%20Notes/KEJURUTERAAN_KOMPUTER/Semester%20%20Sidang%20Akademik%2020112012/EKT420%20Software%20Engineering/Example%20of%20Software%20Design%20Document\(SDD\)/EDDISS.pdf](http://portal.unimap.edu.my/portal/page/portal30/Lecture%20Notes/KEJURUTERAAN_KOMPUTER/Semester%20%20Sidang%20Akademik%2020112012/EKT420%20Software%20Engineering/Example%20of%20Software%20Design%20Document(SDD)/EDDISS.pdf) Date of access.

Jimenez-Ramirez, A., Reijers, H.A., Barba, I. & Del Valle, C. 2019. A method to improve the early stages of the robotic process automation lifecycle. (*In*. International Conference on Advanced Information Systems Engineering organised by: Springer. p. 446-461).

Johannesson, P. & Perjons, E. 2014. Research strategies and methods. *An Introduction to Design Science*. Springer. p. 39-73).

Josey, A. 2016. TOGAF® Version 9.1-A Pocket Guide: Van Haren.

Joshi, D. & Shah, R. 2015. Automation in construction industry. *Indian Journal of Advance Research in Engineering, Science and Management*, 2(3):1-5.

Kaushik, V. & Walsh, C.A. 2019. Pragmatism as a research paradigm and its implications for social work research. *Social Sciences*, 8(9):255.

Khalilzadeh, M. & Alikhani, A.M. 2020. The effects of PMBOK knowledge areas on the phases of ERP implementation. *Industrial Engineering & Management Systems*, 19(1):242-253.

Kimberlin, C.L. & Winterstein, A.G. 2008. Validity and reliability of measurement instruments used in research. *American journal of health-system pharmacy*, 65(23):2276-2284.

King, A.C., Hekler, E.B., Castro, C.M., Buman, M.P., Marcus, B.H., Friedman, R.H. & Napolitano, M.A. 2014. Exercise advice by humans versus computers: Maintenance effects at 18 months. *Health Psychology*, 33(2):192.

Kolb, S.M. 2012. Grounded theory and the constant comparative method: Valid research strategies for educators. *Journal of Emerging Trends in Educational Research and Policy Studies*, 3(1):83.

Koplowitz, R. & Rymer, J. 2019. The four faces of rapid development. <https://go.forrester.com/blogs/the-four-faces-of-rapid-development/> Date of access: 25 April 2020.

Kuechler, B. & Vaishnavi, V. 2008. On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, 17(5):489-504.

Leffingwell, D. 2018. SAFe 4.5 reference guide: Scaled agile framework for lean enterprises: Addison-Wesley Professional.

Leshob, A., Bourgouin, A. & Renard, L. 2018. Towards a process analysis approach to adopt robotic process automation. (In. 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE) organised by: IEEE. p. 46-53).

Lukka, K. 2003. The constructive research approach. *Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B*, 1(2003):83-101.

Malterud, K., Siersma, V.D. & Guassora, A.D. 2016. Sample size in qualitative interview studies: Guided by information power. *Qualitative health research*, 26(13):1753-1760.

Mason, R.O. 1986. Four ethical issues of the information age. *MIS quarterly*:5-12.

Master of Code. 2020. An intelligent approach to business automation: RPA, DPA and AI. <https://masterofcode.com/blog/an-intelligent-approach-to-business-automation-rpa-dpa-ai> Date of access: 25 May 2020.

Masur, J.S. & Posner, E.A. 2010. Against feasibility analysis. *The University of Chicago Law Review*:657-716.

Mending, J., Reijers, H.A. & van der Aalst, W.M. 2010. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2):127-136.

Merriam, S.B. & Tisdell, E.J. 2015. Qualitative research: A guide to design and implementation: John Wiley & Sons.

Mineraud, J., Mazhelis, O., Su, X. & Tarkoma, S. 2016. A gap analysis of Internet-of-Things platforms. *Computer Communications*, 89:5-16.

Mitchell, D.J. 2005. The impact of government spending on economic growth. *The Heritage Foundation*, 1813:1-18.

Mohajan, H.K. 2018. Qualitative research methodology in social sciences and related subjects. *Journal of Economic Development, Environment and People*, 7(1):23-48.

Morse, J.M. 2000. Determining sample size: Sage Publications Sage CA: Thousand Oaks, CA.

Myers, M.D. & Venable, J.R. 2014. A set of ethical principles for design science research in information systems. *Information & Management*, 51(6):801-809.

Nama, G.F. & Kurniawan, D. 2017. An enterprise architecture planning for higher education using the open group architecture framework (togaf): Case study University of Lampung. (In. 2017 Second International Conference on Informatics and Computing (ICIC) organised by: IEEE. p. 1-6).

Oliveira, W. 2015. Understand and apply the business process analysis methodology. <https://www.heflo.com/blog/process-mapping/business-process-analysis-methodology/> Date of access: 17 May 2020.

Pagallo, U. 2017. From automation to autonomous systems: A legal phenomenology with problems of accountability. (In. 26th International Joint Conference on Artificial Intelligence, IJCAI 2017 organised by: International Joint Conferences on Artificial Intelligence. p. 17-23).

Perdicoúlis, A. & Glasson, J. 2006. Causal networks in EIA. *Environmental Impact Assessment Review*, 26(6):553-569.

Qurratuaini, H. 2018. Designing enterprise architecture based on TOGAF 9.1 framework. (In. IOP Conference Series: Materials Science and Engineering organised by: IOP Publishing. p. 012065).

Ramkumar, M. & Sadashivaiah, C. 2018. Robotic process automation – Doing it right first time. <https://site.ieee.org/indiacouncil/files/2018/11/p54-p56.pdf> Date of access: 20 May 2020.

Raza, S.A. & Shah, T.Z. 2012. Work environment and its impact on triple constraint of project management. *Information Management and Business Review*, 4(10):545-552.

Rosenberger, P. & Tick, J. 2018. Suitability of PMBOK 6 th edition for agile-developed IT Projects. (In. 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI) organised by: IEEE. p. 000241-000246).

Rozinat, A. & Günther, C.W. 2019. The added value of process mining. *BPM Everywhere: Internet of Things, Process of Everything*.

Samost-Williams, A. & Nanji, K.C. 2020. A systems theoretic process analysis of the medication use process in the operating room. *Anesthesiology*, 133(2):332-341.

Saunders, M., Lewis, P. & Thornhill, A. 2012. Research philosophy in the 'research onion': Pearson Education Limited.

Shah, M. 2017. Evolving role of a business analyst. *International Journal of Business and Management*, 1(2):7-12.

Shenton, A.K. 2004. Strategies for ensuring trustworthiness in qualitative research projects. *Education for information*, 22(2):63-75.

Simon, H.A. 2019. The sciences of the artificial: MIT Press.

Skinner, C. 2020. Quantitative research. Handbook for research students in the social sciences. Routledge. p. 215-224).

Smith, A. & Fressoli, M. 2021. Post-automation. <https://www.sciencedirect.com/science/article/pii/S0016328721000872> Date of access: 20 September 2021.

Smith, M.L., Erwin, J. & Diaferio, S. 2005. Role & responsibility charting (RACI). Paper presented at the Project Management Forum (PMForum).

Snyder, H. 2019. Literature review as a research methodology: An overview and guidelines. *Journal of business research*, 104:333-339.

Stake, R.E. 2006. Multiple case study analysis. New York: Guilford, 2006.

Stray, V., Moe, N.B. & Sjoberg, D.I. 2018. Daily stand-up meetings: Start breaking the rules. *IEEE Software*, 37(3):70-77.

Swanepoel, J.W.H., Swanepoel, C., Van Graan, F., Allison, J. & Santana, L. 2021. Elementary statistical methods: Axiom Academic Publishers.

Tashakkori, A., Teddlie, C. & Teddlie, C.B. 1998. Mixed methodology: Combining qualitative and quantitative approaches. Vol. 46: Sage.

Thomas, D.R. 2003. A general inductive approach for qualitative data analysis. *The American Journal of Evaluation*, 27.

Tootian, E.S. 2018. Identification of obstacles caused by organizational silence in policies for paying medical expenses in social security organization of Alborz Province.

Ubis, F., Kirkham, T., Matthews, B., Lastra, J.M., Harrison, R., Herrera, V.V. & Chowdrey, A. 2008. The challenges along the road to the realisation of a factory automation lifecycle. (In. 2008 32nd Annual IEEE International Computer Software and Applications Conference organised by: IEEE. p. 559-562).

UiPath. 2020a. Details about the PDD. <https://docs.uipath.com/task-capture/docs/details-about-the-pdd> Date of access: 24 May 2020.

UiPath. 2020b. Employee-driven assessment. <https://docs.uipath.com/automation-hub/docs/employee-driven-assessment> Date of access: 24 May 2020.

UiPath. 2020c. Fast track support release: Built for hyperautomation. <https://www.uipath.com/blog/2020-fast-track-support-release-for-hyperautomation> Date of access: 17 May 2020.

UiPath. 2021. Process discovery tool: Task Capture. <https://www.uipath.com/product/task-capture> Date of access: 19 August 2021.

Vafadar, A., Tolouei-Rad, M., Hayward, K. & Abhary, K. 2016. Technical feasibility analysis of utilizing special purpose machine tools. *Journal of Manufacturing Systems*, 39:53-62.

Van den Broucke, S.K. & De Weerd, J. 2017. Fodina: A robust and flexible heuristic process discovery technique. *decision support systems*, 100:109-118.

Van Der Aalst, W. 2011. Process mining: discovery, conformance and enhancement of business processes. Vol. 2: Springer.

Van der Aalst, W.M., Bichler, M. & Heinzl, A. 2018. Robotic process automation: Springer.

Weijters, A., van Der Aalst, W.M. & De Medeiros, A.A. 2006. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166:1-34.

Whitten, J., Bentley, L. & Dittman, K. 2001. System analysis and design methods: NJ: McGraw-Hill.

Ziemons, E., Hubert, C. & Hubert, P. 2019. Process analysis. *Encyclopedia of Analytical Science*:1-7.

Zyphur, M.J. & Pierides, D.C. 2017. Is quantitative research ethical? Tools for ethically practicing, evaluating, and using quantitative research. *Journal of Business Ethics*, 143(1):1-16.

ANNEXURE A: PROCESS ASSESSMENT

Process Name:	Auto Package Upgrade and Rollback
Business Unit:	ITS
Main Function:	DevOps
SME Responsible:	Jacqui Muller
Process Owner:	Jacqui Muller
RPA Business Analyst:	Jacqui Muller

Type	Question	Explanation	Answers	Result Computation
INPUT TYPES	How do the <u>majority</u> of your data inputs look like?	Digital: any data that can be stored on a machine [server, laptop, IOT devices] Structured data: has a predictable format, exists in fixed fields (e.g. an xls cell or a field in a form) and is easily detectable via search algorithms. Think about your structured data as: excels, tables with fixed structure, databases where you can easily extract your information through a search function. Unstructured data: may have its own internal structure, but this is not highly predictable. Examples: web logs, multimedia content, customer service interactions and social media data.	Digital and Structured	Feasible
PROCESS STABILITY	How will your process change in the next <u>6 months</u> ?	Think about changes of the following type: - legislative changes that are upcoming and could change the rules/logic of the process. - organizational changes such as re-distributing bits of the processes between different roles/departments; - industry shifts that could change the business model and hence the rules of the process;	No change expected	0
APPLICATIONS STABILITY	How will your applications change in the next <u>6 months</u> ?	Think about not only applications but also interfaces, menus in apps and reports.	No change expected	0

OVERALL RESULT 0

BENEFITS	How many <u>FTEs</u> are currently needed to perform the process?	Optional field	3	2
	What is the <u>frequency</u> of the process?		Weekly	
	What is the <u>volume of transactions/frequency</u> (number of times the process is ran/selected frequency)?	E.g. If the process happens weekly please then we need to know the volume that happens on average during a week.	1	
	What is the <u>average time it takes for the process to be ran once</u> (average handling time/transaction)?	Please quantify the average handling time in minutes e.g. if 4 hours then insert 240	90	
	What is the average number of <u>human errors</u> ?		40%	
PROCESS COMPLEXITY	How would you characterize <u>the peaks</u> of the process?		The process does not have peaks	N/A
	How many <u>steps</u> does the process have?		<=10 steps	0,1
	How <u>difficult</u> are the decisions that you must take to complete the process?		The process involves simple decisions (yes/no type)	0,2
	What is the average number of cases where you are unable to complete the entire process? (Either because you require input from a different person or because you end up in a situation that is not covered by a clear rule)		5%	
	What is the <u>number of applications</u> that you use for the process?		4-5 applications	0,6
INPUT DATA	Are any of the applications accessed via <u>VDI</u> ?		No	1
	What % of your <u>input data</u> is digital?		100%	
	Is any of your digital input <u>scanned</u> ?		No	1
	What % of your input data is structured?	Structured data: has a predictable format, exists in fixed fields (e.g. an xls cell or a field in a form) and is easily detectable via search algorithms. Think about your structured data as: excels, tables with fixed structure, databases where you can easily extract your information through a search function. Examples: if your inputs consist of 5 excel files/or ERP reports then 100% of your data is structured, whereas if your input consists of 4 excel files/ERP reports and 1 e-mail text then 80% of your data is structured.	40%-60%	0,7

Ease Of Implementation	41%	Medium Effort
Benefit/Suitability	58%	Medium High Benefit

Bandwidth Freed (man hours/year)	2 413
Error Reduction	58%
Average Handling Time Reduction (AHT Reduction)	39%
Effort to Implement (man hours)	840

ANNEXURE B: PROCESS DEFINITION DOCUMENT

Introduction

The process definition document outlines the business process chosen for automation. The document describes the sequence of steps performed as part of the business process, the conditions and rules of the process prior to automation (AS IS), as well as the new sequence of steps that the process will follow as a result of preparation for automation (TO BE).

The PDD is a communication document between:

- The business analyst and the SME/process owner. The goal is to ensure that the business analyst has the correct understanding of the process and has represented it accurately.
- The business analyst and the development team (represented by the solution architect and development lead). The goal is to ensure that the process is documented appropriately and to a sufficient level of detail so that the solution architect can then create the solution based on the PDD content.

Objectives

The main objective of automating the updating of packages is to enable the operations team to focus on other incidents and requests that are not related to package upgrades. The benefits expected by the process owner after automation are time saving, efficiency gains and more process understanding.

Minimum prerequisites for the automation

The prerequisites for automation are as follows:

1. Filled in process definition document
2. Test data to support development
3. User access and user account creations (licenses, permissions, restrictions to create accounts for robots)
4. Credentials (user ID and password) required to logon to machines and applications

AS IS process

This section provides detailed insight into the current process that is proposed for automation.

AS IS process overview

The current process is subject to the following information:

Item	Description/answer
Process name	Auto-package upgrade
Automation area	DevOps
Short description	The automation of package upgrades and testing thereof
Roles required	Developer and/or operations engineer
Process frequency	Daily
Number of times the process is run by selected frequency	Total runs per annum = 52 Total runs per month = 4-5 (calendar dependent) Total runs per week = 1
Process scheduling and restrictions	The upgrades will be done at 21:00 during workdays and not on the weekend.
Peak period	There are no peak periods.
Peak volume approximate increase	Not Applicable
Number of persons performing the process	Three operations support engineers
Expected volume increase during next periods	This is dependent on the number of projects created. The estimated increase is 10% per year.
Percentage unhandled exceptions	Human errors are the main unhandled exceptions – this solution will reduce the likelihood of upgrades being done due to more important operational matters that need to be addressed by the team. The estimated percentage of unhandled exceptions is 75%.
Input data description	Project solution files Process configuration information Library information
Output data description	Reports on upgrade success and failures.

Table 1: AS IS process overview

AS IS process applications used

The following applications are used within the current process:

Application	Version	Application language	Thin/thick client
Command prompt (or appropriate IDE)	N/A	N/A	Thick
Microsoft Excel	Microsoft Office 365 ProPlus	N/A	Thick
File Explorer	N/A	N/A	Thick

Table 2: Applications used

High-level AS IS process flow

This section contains various process maps contributing to a better understanding of how the process is performed pre-automation.

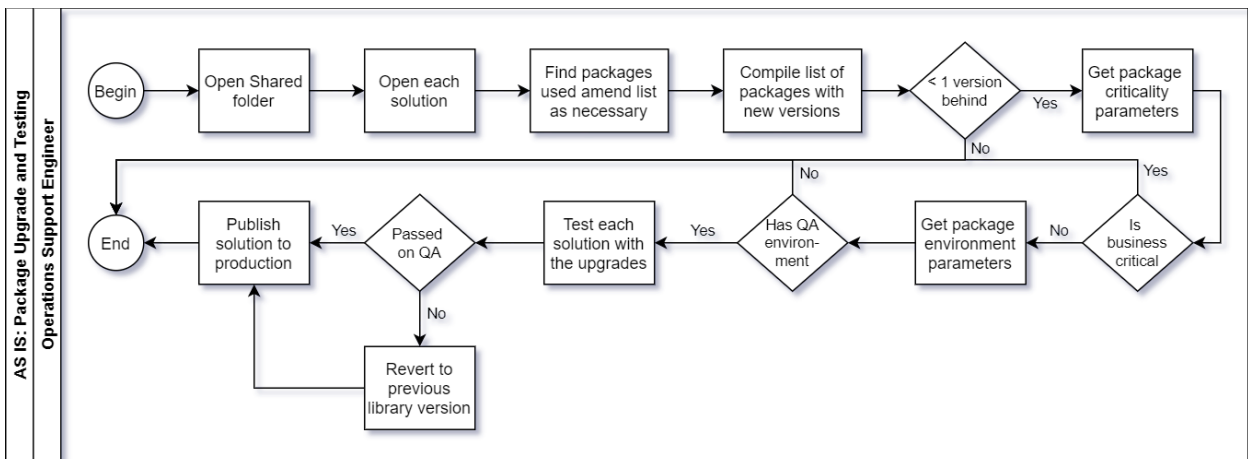


Figure 1: AS IS high-level

Detailed AS IS process steps

Table 3 shows how the AS-IS process flows with regard to input, description, exceptions, and possible actions.

# Step	Input	Description	Exceptions handling	Possible actions
1	Open Shared folder	<p>A shared folder is used to store all of the solutions that need to be accessed as the first step of the process. Sub-folders also need to be catered for as some solution folders may contain different solutions specific to different versions and environments.</p> <p>This step is done by going through the files through File Explorer.</p>	<p>3. The shared folder is inaccessible</p> <p>4. Sub-folders may be in use by other processes</p>	<p>1. Retry later</p> <p>2. Retry later</p>
2	Open each solution	<p>Each solution found in the shared folder (and subfolders) needs to be opened in order to find the package names.</p> <p>This step is done by going through the files through File Explorer.</p>	<p>2. Sub-folders may be in use by other processes</p>	<p>2. Retry later</p>
3	Find packages used amend list as necessary	<p>A list of libraries used in each solution is compiled and amended as necessary. As soon as a new library is found, it is added to the list. As soon as a new solution is found, it should</p>	N/A	N/A

		<p>be added to the list along with the libraries used in the solution and the current version.</p> <p>This step is done by going through the solutions in the relevant IDE (either notepad++ or PyCharm). The list is to be maintained in an Excel document</p>		
4	Compile list of packages with new versions	<p>The distinct list of libraries should be used in order to get the list of packages</p> <p>This step is executed by running a command in the command prompt. The list is to be recorded in an Excel document.</p>	N/A	N/A
5	Get package criticality parameters	<p>The Excel sheet is filtered to find each solution. The corresponding value from the field marked "Is Business Critical" is extracted.</p> <p>This step is executed using Excel.</p>	N/A	N/A
6	Get package environment parameters	<p>The Excel sheet is filtered to find each solution and the applicable parameters</p> <p>This step is executed through Excel.</p>	N/A	N/A

7	Test each solution with the upgrades	<p>Once the solutions feasible for testing have been identified, these solutions are used to test the success of the upgrade on the QA environment.</p> <p>The execution of the solution is done through IDLE (or any relevant IDE).</p>	N/A	N/A
8	Revert to previous library version	<p>In a case where a test on QA fails, the upgrades need to be reverted (on QA).</p> <p>This is done through the IDLE (or any relevant IDE).</p>	1. Failed to revert	2. Retry later
9	Publish solution to production	<p>Successful QA upgrades should be replicated on the production environment.</p> <p>This is done through IDLE (or any relevant IDE)</p>	2. Failed to publish	2. Raise alert for manual oversight in next attempt
10	Report on success and failure	<p>A report showing the number of successes and failures should be refreshed with the latest results.</p> <p>The report is a Power BI report and can be viewed in Power BI Desktop or on the online Power BI service.</p>	N/A	N/A

Table 3: AS IS process steps

Input data description

The following table contains details regarding the inputs that every step of the process takes.

# Step	Input type	Are inputs * <i>Native Digital?</i>	Are the Inputs * <i>Structured Data?</i>
1	File Explorer	Yes	Yes
2	File Explorer	Yes	Yes
3	Excel/ Notepad ++	Yes	Yes
4	Excel	Yes	Yes
5	Excel	Yes	Yes
6	Excel	Yes	Yes
7	IDLE	Yes	Yes
8	IDLE	Yes	Yes
9	IDLE	Yes	Yes
10	Power BI	Yes	Yes

Table 4: Input Data

* **Native Digital:** This is data that was originally created digitally e.g. Excel, database or application reports etc. The non-native digital inputs are usually scanned images.

* **Structured Data:** has a predictable format and exists in fixed fields (e.g. an excel cell or a field in a form) and is easily detectable via search algorithms.

TO BE process

This section provides detailed insight into the desired future state of the process that is proposed for automation.

Detailed TO BE process map

Figure 2 depicts the desired future state (or TO BE) process flow of steps executed as part of the process:

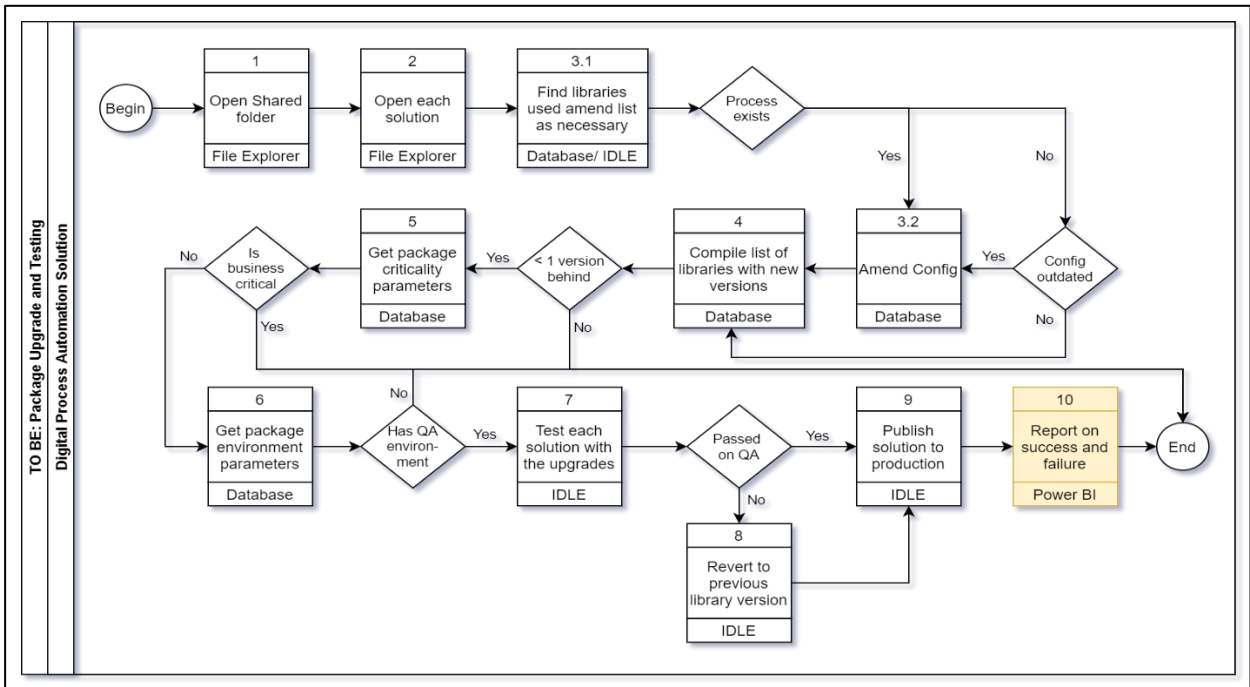


Figure 2: Detailed To-Be process

In-scope for automation

The steps in-scope for automation are be listed in Table 5.

Number	Detail
1	Retrieve a list of processes
2	Retrieve a list of packages
3	Retrieve a list of processes that need to be upgraded
4	Upgrade the packages that need to be upgraded
5	Test that the upgrade did not break any of the processes
6	Report on the successes and failures

Table 5: In-scope process steps

Out-of-scope for automation

The steps out-of-scope for automation are be listed in Table 6 together with the reasoning.

Activity/ Step	Reason for out of scope	Impact on the TO BE	Possible measures to be taken into account for future automation
N/A	Package upgrades for projects built using languages or platforms other than Python and UiPath	There is no impact on the TO BE	A future enhancement could be to cater for other languages and platforms.

Table 6: Out-of-scope process steps

Exception handling

Exceptions are of two types and both are addressed:

- **Known exceptions:** Previously encountered. A scenario is defined with clear actions and workarounds for each case.
- **Unknown exceptions:** New situation that was not encountered before. It cannot be predicted and in case it happens it needs to be flagged and communicated to an authorised person for evaluation.

For all the other unanticipated or unknown business (process) exceptions, the robot should:

- Send an email notification to the developer and the error message screenshot attached.

Exception type	Exception name	Action to be taken
Known	File not accessible	Rerun the process.
Known	Use of deprecated method	Notify the developer

Table 7: Exception handling

Reporting

In this section, all the reporting requirements of the business are detailed so that when the RPA solution is moved to production, the administrators can track the performance of the solution.

Report type	Update frequency	Details	Monitoring tool to visualise the data
Summary	Weekly	<ol style="list-style-type: none"> 1. Number of projects that required upgrades 2. Number of libraries/package upgrades 3. Number of successful upgrades 4. Number of failed upgrades 	Excel/Power BI

Table 8: Reports

Process documentation

Table 9 details the documentation elements that have been provided for the automation to be built.

Standard operating procedure	Yes/No
High-level process map	Yes
Detailed level process steps	Yes
Work instructions	Yes
Input files	Yes
Output files	Yes

Table 9: Process documentation

ANNEXURE C: SOLUTION DESIGN DOCUMENT

Introduction

Outlines the major components of the master project (the overall output of the development, containing one or multiple projects that together cover the scope of the automation process) taking into consideration all the business restrictions (scheduling, peaks, future increases in volume etc.). The focus of the solution architect will be on:

- Robustness;
- Scalability;
- Efficiency;
- Replicability;
- Reusability of component

The information herein is targeted primarily at the developers who will initially implement the solution and subsequently at the support developers in case of change requests.

Objectives

The main objective of automating the updating of packages is to enable the operations team to focus on other incidents and requests that are not related to package upgrades. The benefits expected by the process owner after automation are time saving, efficiency gains and more process understanding.

Development guide

This section details the elements used within the development process of the project.

Architectural structure

This project will include the use of a Python script that will perform a series of steps in order to upgrade packages, test the upgrades and rollback (where necessary).

The high-level technical architecture is outlined in Figure 1.

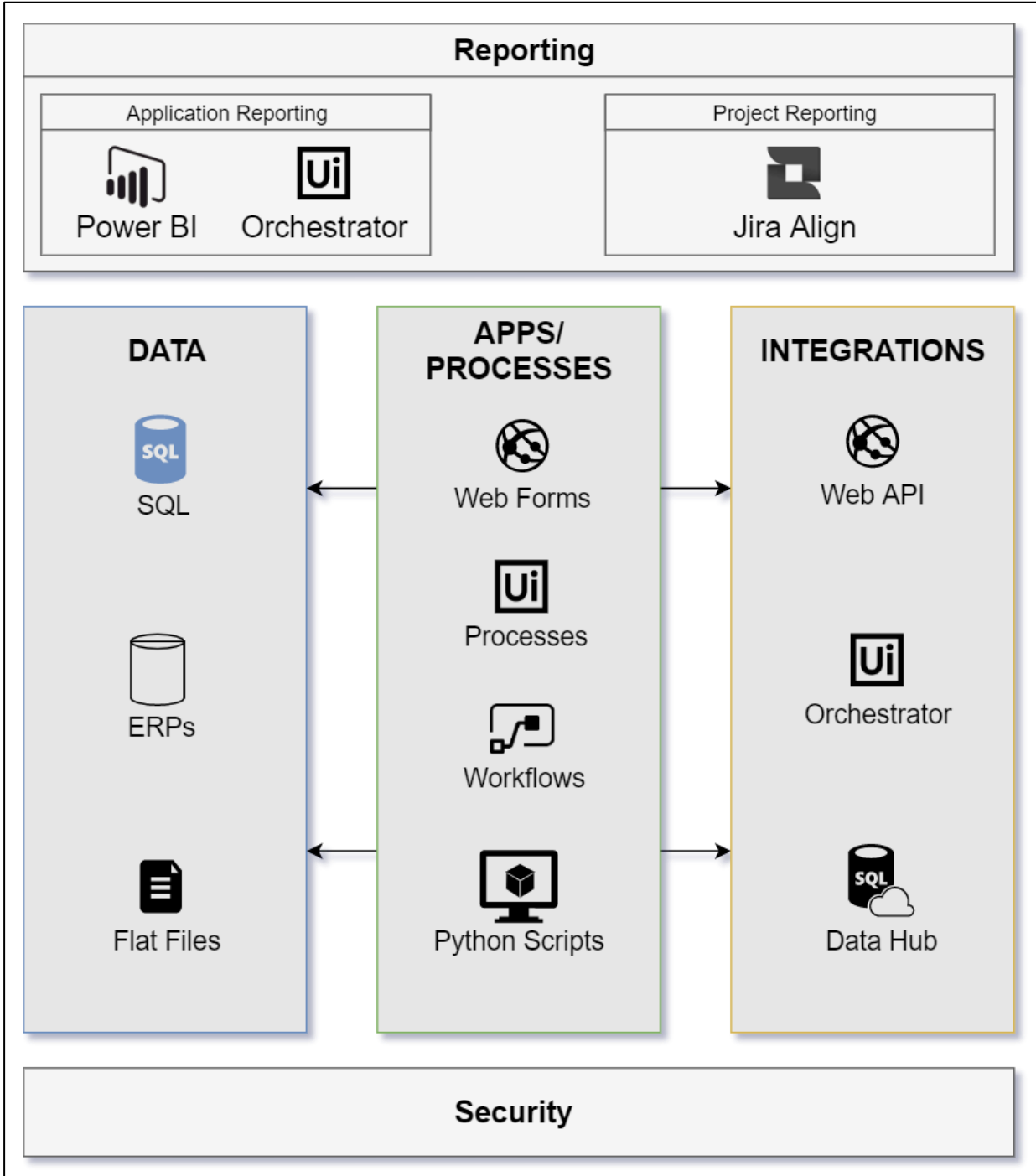


Figure 1: High-level technical architecture

The high-level application architecture of the project is as follows:

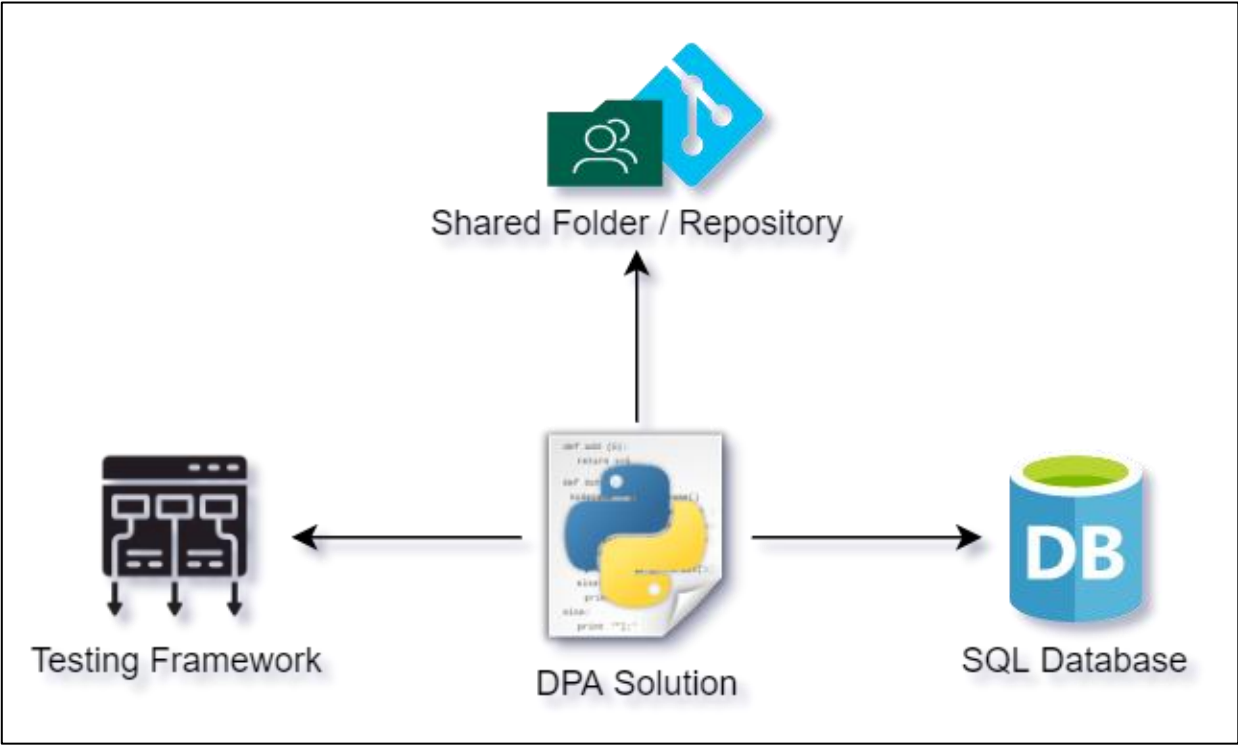


Figure 2: High-level application architecture

Based on the Figure 2, the Python DPA solution interacts with an SQL database, the code repository (or the shared folder). It also encapsulates a testing framework that can be expanded upon as the DPA solution is enhanced to cater for more solutions.

Therefore, the DPA solution is further broken down into multiple components, namely:

- Package information retrieval
- Latest package version retrieval
- Upgrade package DSS (which includes the testing framework)

Detailed process definition

The above architecture is broken down further, as stated in the TO BE section of the process definition document (PDD). The process is outlined in Figure 3.

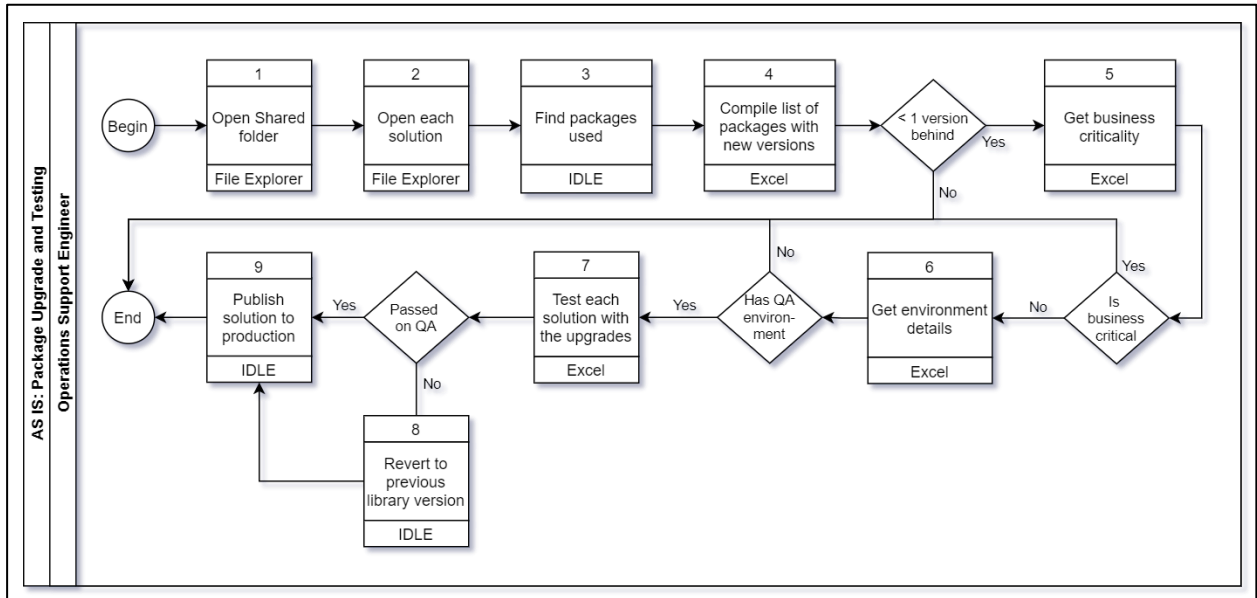


Figure 3: Process flow steps

Figure 3 encapsulates all three elements of the process that are further broken down in the subsections below.

Package information retrieval

The package information retrieval process will be the first workflow that makes up the DPA solution. It will go through all of the folders and subfolders that exist on a specified shared drive. These folders and subfolders are assessed as solution folders that make use of nuget (among other) libraries and packages.

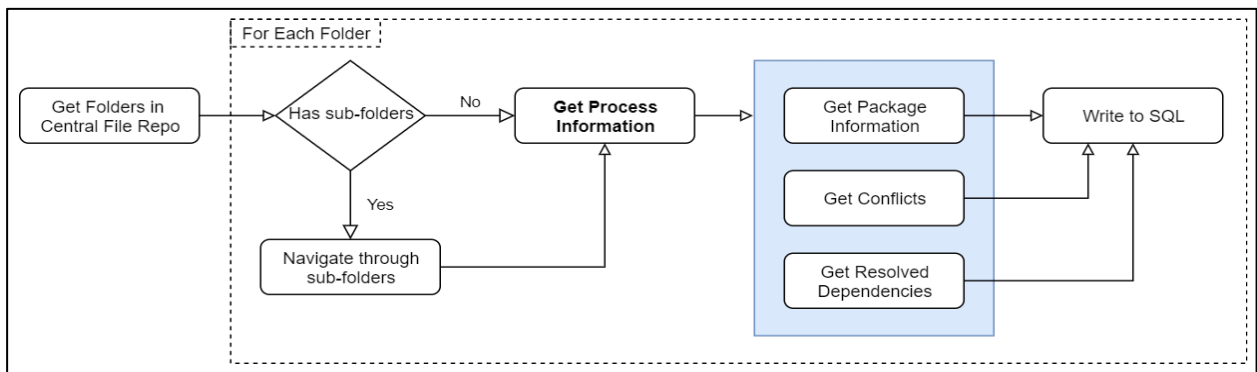


Figure 4: Package information retrieval process steps

The next workflow process that follows the package information retrieval is the retrieval of latest package versions retrieval process, which is described in the subsection below.

Latest package version retrieval

The latest package version retrieval process retrieves a complete list of all packages used by any of the solutions found on the shared drive. The process then queries the latest package version and keeps a record of new versions in an SQL database.

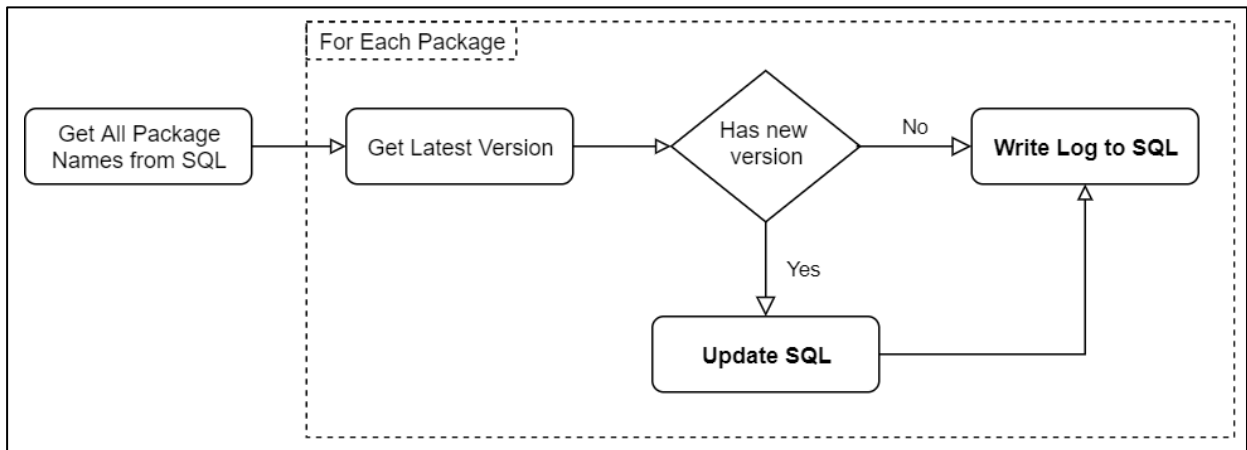


Figure 5: Latest package version retrieval process steps

The last workflow process that follows the latest package version retrieval is the package upgrade decision support system (DSS) process, which is described in the subsection below.

Upgrade package decision support system (DSS)

This is the portion of the solution where the decision support system is used as the foundation for deciding which packages to upgrade, as well as which solutions to upgrade. The upgrade package DSS process retrieves a complete list of all solutions that require one or more package upgrades. These solutions are then mapped against a mapping table that contains the following information:

- Solution
- Package
- Number of versions behind
- Has QA environment
- Test succeeded on QA
- Test succeeded on production
- Datetime stamp

Only in a case where a package is less than one full version behind, is not critical to other business operations, has a QA environment and succeeds testing in the QA environment, may the package be upgraded in production.

Figure 6 details the steps that make up the upgrade package DSS process:

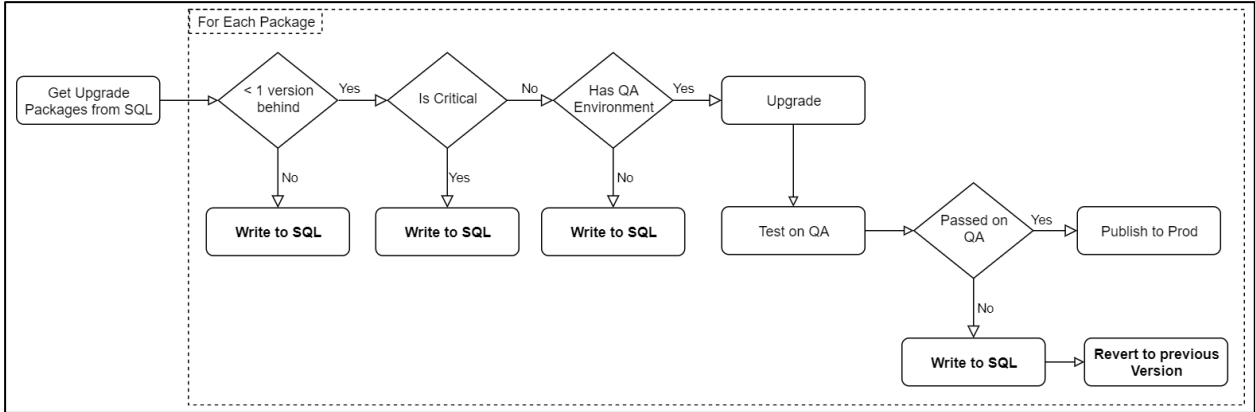


Figure 6: Upgrade package DSS process steps

The subsection below addresses the data flow diagram.

Data flow diagram

The process relies heavily on the database and the objects that it contains. Figure 7 defines the data flow.

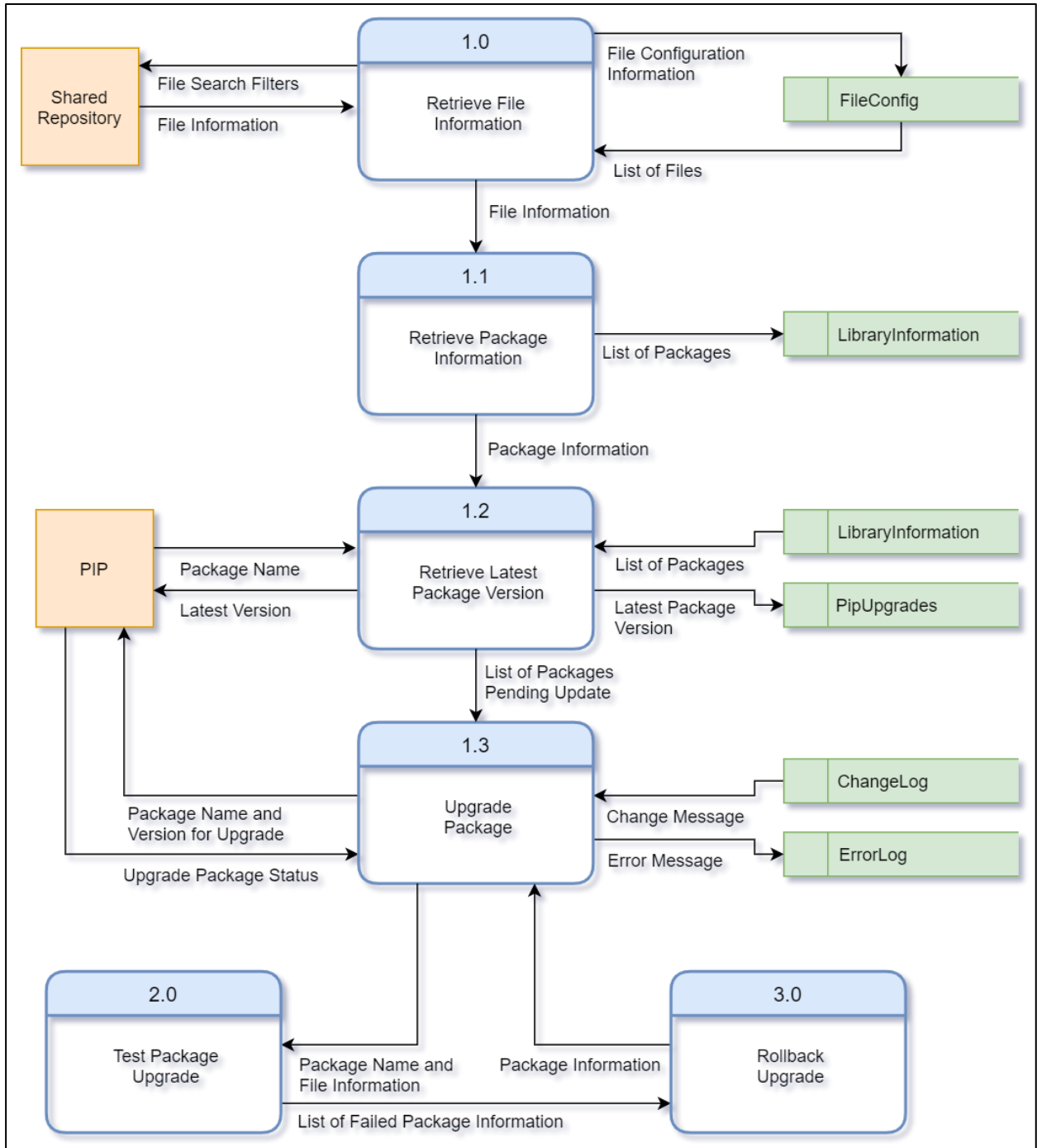


Figure 7: Data flow diagram

As seen in Figure 7, certain data stores are needed to store the data being processed. The current environment advises that the most feasible collective data store for the solution would be an SQL database hosted on the existing SQL server.

Database design

The database design is outlined as follows:

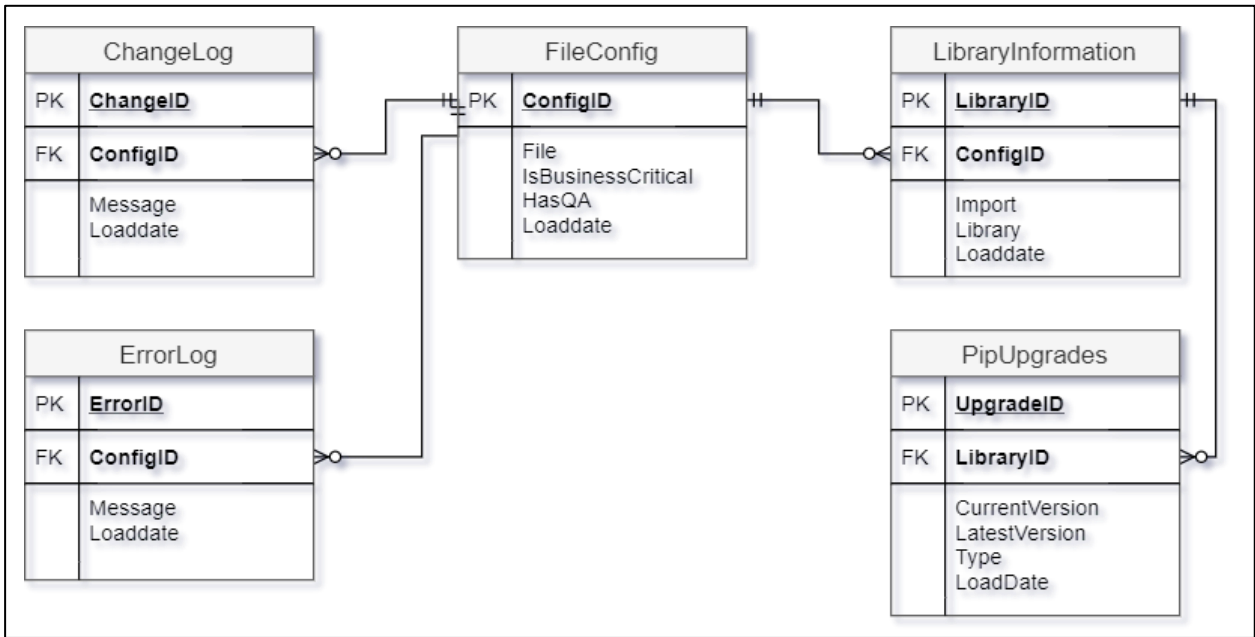


Figure 8: Entity relationship design (ERD) of the database design

Figure 8 accommodates for all of the data that needs to be captured and allows for seamless integration with the rest of the environment.

Project details

Table 1 outlines the automated process:

Item name	Description
Master project name	Auto-package upgrade
Environment(s) required	Package information retrieval Latest package version retrieval Upgrade package decision support system
Prerequisites to run	Python SQL server
Input	List of files/scripts List of libraries List of latest versions
Expected output	Updated library versions
How to start the automated process	There will be a time trigger set up. As soon as an item enters, the process will be kicked off.
Reporting	Error and change log reporting will be done
Password policies	N/A
Stored credentials	scriptsettings.json
Schedule details	Runs daily at 08:00

Table 1: Project details

ANNEXURE D: PROCESS UPGRADE INFORMATION (PROCESS CONFIGURATION)

Script	Business Critical?	Has QA Environment?	Passed QA?	Published?
Projects\Moving Average Forecast.py	No	Yes	Yes	Yes
Projects\Superstore Data Manipulation.py	No	Yes	Yes	Yes
Projects\Slack Message.py	No	Yes	Yes	Yes
Projects\PIP Libraries to SQL.py	No	Yes	Yes	Yes
Projects\Microsoft Teams Message.py	No	Yes	Yes	Yes
Projects\Linear Trend Projection Forecasting.py	No	Yes	Yes	Yes
Projects\Get Current Weather.py	No	Yes	Yes	Yes
Projects\Exponential Smoothing.py	No	Yes	Yes	Yes
02 Schedule Optimisation\Schedule Optimisation.py	Yes	No	NA	NA
02 Schedule Optimisation\SQL Schedule.py	Yes	No	NA	NA
03 UiPath Config Manager\Config Manager.py	Yes	No	NA	NA
03 UiPath Config Manager\Config Manager.py	Yes	Yes	NA	NA
03 UiPath Config Manager\Latest Library Version.py	Yes	No	NA	NA
03 UiPath Config Manager\Package Manager.py	Yes	No	NA	NA
03 UiPath Config Manager\Package Manager.py	Yes	Yes	NA	NA
03 UiPath Config Manager\Process Name Extraction (With File).py	Yes	No	NA	NA
03 UiPath Config Manager\SAP Account Numbers.py	Yes	No	NA	NA
05 Structure Compare\Comparison.py	Yes	No	NA	NA
06 Library Compare\Compare Library.py	Yes	No	NA	NA
06 Library Compare\Compare.py	Yes	No	NA	NA
06 Library Compare\JSON Schema Compare.py	Yes	No	NA	NA
07 Logging\Database Logging.py	Yes	No	NA	NA
07 Logging\File Logging.py	Yes	No	NA	NA
08 SQL Monitoring\SQL Job Monitoring.py	Yes	No	NA	NA
09 Website Scraper\ITSM Scraping.py	Yes	No	NA	NA

10 Website Scraper\Call content.py	Yes	No	NA	NA
11 Inbox Monitoring\Inbox Job Monitoring.py	Yes	No	NA	NA
11 Inbox Monitoring\Mail Scraping.py	Yes	No	NA	NA
11 Inbox Monitoring\Move Mail.py	Yes	No	NA	NA
11 Inbox Monitoring\Auth.py	Yes	No	NA	NA
11 Inbox Monitoring\Multiple Attachments.py	Yes	No	NA	NA
11 Inbox Monitoring\pyo365.py	Yes	No	NA	NA
11 Inbox Monitoring\SMTP.py	Yes	No	NA	NA

ANNEXURE E: PROCESS UPGRADE INFORMATION (SCRIPT LIBRARIES)

Project	Library	Checked	Current version	Latest version	Requires update
Config Manager	pandas	Yes	1.5.0	1.5.1	Yes
	json	Yes	0.9.5	0.10.0	Yes
	os	Yes	3.1.1	4.0.2	Yes
	io	Yes	1.5.0	4.5.0	Yes
	datetime	Yes	2.8	3.4.7	Yes
	pyodbc	Yes	2.4.4	2.8.3	Yes
	sqlalchemy	Yes	1.14.0	1.16.0	Yes
	urllib	Yes	1.4.1	1.4.3	Yes
Library Compare	pandas	Yes	1.5.0	1.5.1	Yes
	json	Yes	0.9.5	0.10.0	Yes
	os	Yes	3.1.1	4.0.2	Yes
	io	Yes	1.5.0	4.5.0	Yes
	datetime	Yes	2.8	3.4.7	Yes
	pyodbc	Yes	2.4.4	2.8.3	Yes
	sqlalchemy	Yes	1.14.0	1.16.0	Yes
	urllib	Yes	1.4.1	1.4.3	Yes
Inbox Monitoring	email	Yes	0.16	0.17.1	Yes
	imaplib	Yes	2.9	3.2	Yes

	json	Yes	0.9.5	0.10.0	Yes
Schedule Optimisation	collections	Yes	3.0.4	4.0.0	Yes
	datetime	Yes	2.8	3.4.7	Yes
	pandas	Yes	1.5.0	1.5.1	Yes
	matplotlib.pyplot	Yes	1.1.1	2.0.1	Yes
	numpy	Yes	0.14.0	0.16.2	Yes
	requests	Yes	2020.2.20	2021.4.4	Yes
	json	Yes	0.9.5	0.10.0	Yes
	os	Yes	3.1.1	4.0.2	Yes
Structure Compare	xmltodict	Yes	1.12.0	1.12.1	Yes
	json	Yes	0.9.5	0.10.0	Yes
	oauth2client.tools	Yes	0.14.0	0.16.2	Yes
	datetime	Yes	2.8	3.4.7	Yes
	pandas	Yes	1.5.0	1.5.1	Yes
	numpy	Yes	0.14.0	0.16.2	Yes
	requests	Yes	2020.2.20	2021.4.4	Yes
	json	Yes	0.9.5	0.10.0	Yes
	os	Yes	3.1.1	4.0.2	Yes
Logging	json	Yes	0.9.5	0.10.0	Yes
	requests	Yes	2020.2.20	2021.4.4	Yes
SQL Monitoring	osr	Yes	3.1.1	4.0.2	Yes
	requestsr	Yes	2020.2.20	2021.4.4	Yes
	PySimpleGUIQt	Yes	2.4.6	2.4.7	Yes
	win10toast	Yes	0.34.2	0.36.2	Yes

	timer	Yes	0.3.1	0.4.1	Yes
Exponential Smoothing	pandas	Yes	1.5.0	1.5.1	Yes
	numpy	Yes	0.14.0	0.16.2	Yes
Get Current Weather	datetime	Yes	2.8	3.4.7	Yes
	pandas	Yes	1.5.0	1.5.1	Yes
	requests	Yes	2020.2.20	2021.4.4	Yes
	json	Yes	0.9.5	0.10.0	Yes
	time	Yes	0.3.1	0.4.1	Yes
Linear Trend Projection Forecasting	pandas	Yes	1.5.0	1.5.1	Yes
	numpy	Yes	0.14.0	0.16.2	Yes
Microsoft Teams Message	pymsteams	Yes	2.4.4	2.8.3	Yes
Moving Average Forecast	pandas	Yes	1.5.0	1.5.1	Yes
	numpy	Yes	0.14.0	0.16.2	Yes
PIP Libraries to SQL	pandas	Yes	1.5.0	1.5.1	Yes
	json	Yes	0.9.5	0.10.0	Yes
	os	Yes	3.1.1	4.0.2	Yes
	sys	Yes	0.3.1	0.4.1	Yes
	io	Yes	1.5.0	4.5.0	Yes
	datetime	Yes	2.8	3.4.7	Yes
	pyodbc	Yes	2.4.4	2.8.3	Yes
	sqlalchemy	Yes	1.14.0	1.16.0	Yes
Slack Message	urllib	Yes	1.4.1	1.4.3	Yes
	urllib	Yes	1.4.1	1.4.3	Yes
Superstore Data Manipulation	pandas	Yes	1.5.0	1.5.1	Yes
	json	Yes	0.9.5	0.10.0	Yes

Website Scraper	re	Yes	5.3	5.4.1	Yes
	requests	Yes	2020.2.20	2021.4.4	Yes
	urllib.request	Yes	1.4.1	1.4.3	Yes
	time	Yes	0.3.1	0.4.1	Yes
	io	Yes	1.5.0	4.5.0	Yes
	bs4	Yes	1.15.16	1.20.88	Yes

ANNEXURE F: PROCESS UPGRADE INFORMATION (PACKAGE VERSIONS)

Package	Current version	Latest version	Environment
asn1crypto	0.24.0	1.3.0	venv_dev
astroid	2.0.4	2.3.3	venv_dev
attrs	19.1.0	19.3.0	venv_dev
azure-functions	1.0.4	1.2.0	venv_dev
boto3	1.11.9	1.12.16	venv_dev
botocore	1.14.9	1.15.16	venv_dev
certifi	2018.8.13	2019.11.28	venv_dev
cff	1.12.3	1.14.0	venv_dev
colorama	0.3.9	0.4.3	venv_dev
cryptography	2.7	2.8	venv_dev
deepdiff	3.3.0	4.2.0	venv_dev
Django	2.1.3	3.0.4	venv_dev
docker	4.0.2	4.2.0	venv_dev
docutils	0.15.2	0.16	venv_dev
idna	2.7	2.9	venv_dev
isort	4.3.4	4.3.21	venv_dev
Jinja2	2.10.1	2.11.1	venv_dev
json	0.9.4	0.9.4	venv_dev
jsonpickle	1.2	1.3	venv_dev
keyring	21.1.0	21.1.1	venv_dev
lazy-object-proxy	1.3.1	1.4.3	venv_dev
lxml	4.4.1	4.5.0	venv_dev
nipyapi	0.13.2	0.14.0	venv_dev
oauthlib	2.1.0	3.1.0	venv_dev
packaging	19.1	20.3	venv_dev
paho-mqtt	1.4.0	1.5.0	venv_dev
pandas	0.25.1	1.0.1	venv_dev
pathspec	0.6.0	0.7.0	venv_dev
psutil	5.6.7	5.7.0	venv_dev
pycparser	2.19	2.20	venv_dev
Pygments	2.5.2	2.6.1	venv_dev
pylint	2.1.1	2.4.4	venv_dev
pyodbc	4.0.27	4.0.30	venv_dev
pyOpenSSL	19.0.0	19.1.0	venv_dev
pyparsing	2.4.2	2.4.6	venv_dev
python-dateutil	2.7.3	2.8.1	venv_dev
pytz	2018.5	2019.3	venv_dev
pywin32	224	227	venv_dev
PyYAML	3.13	5.3	venv_dev
regex	2019.12.9	2020.2.20	venv_dev
requests	2.20.1	2.23.0	venv_dev

requests-oauthlib	1.0.0	1.3.0	venv_dev
ruamel.yaml	0.15.94	0.16.10	venv_dev
s3transfer	0.3.2	0.3.3	venv_dev
setuptools	45.2.0	45.3.0	venv_dev
six	1.11.0	1.14.0	venv_dev
SQLAlchemy	1.3.8	1.3.13	venv_dev
typed-ast	1.4.0	1.4.1	venv_dev
urllib3	1.23	1.25.8	venv_dev
virtualenv	16.7.9	20.0.9	venv_dev
websocket-client	0.56.0	0.57.0	venv_dev
Werkzeug	0.15.6	1.0.0	venv_dev
wrapt	1.10.11	1.12.0	venv_dev
zipp	2.1.0	3.1.0	venv_dev
appdirs	1.4.3	1.4.4	venv_qa
asgiref	3.2.3	3.3.4	venv_qa
asn1crypto	1.3.0	1.4.0	venv_qa
astroid	2.3.3	2.5.7	venv_qa
attrs	19.3.0	21.2.0	venv_qa
azure-functions	1.2.0	1.7.2	venv_qa
boto3	1.12.16	1.17.88	venv_qa
botocore	1.15.16	1.20.88	venv_qa
certifi	2019.11.28	2021.5.30	venv_qa
cffib	1.14.0	1.14.5	venv_qa
chardet	3.0.4	4.0.0	venv_qa
colorama	0.4.3	0.4.4	venv_qa
cryptography	2.8	3.4.7	venv_qa
deepdiff	4.2.0	5.5.0	venv_qa
distlib	0.3.0	0.3.2	venv_qa
Django	3.0.4	3.2.4	venv_qa
docker	4.2.0	5.0.0	venv_qa
docutils	0.16	0.17.1	venv_qa
idna	2.9	3.2	venv_qa
importlib-metadata	1.5.0	4.5.0	venv_qa
isort	4.3.21	5.8.0	venv_qa
Jinja2	2.11.1	3.0.1	venv_qa
jmespath	0.9.5	0.10.0	venv_qa
jsonpickle	1.3	2.0.0	venv_qa
keyring	21.1.1	23.0.1	venv_qa
lazy-object-proxy	1.4.3	1.6.0	venv_qa
lxml	4.5.0	4.6.3	venv_qa
MarkupSafe	1.1.1	2.0.1	venv_qa
nipyapi	0.14.0	0.16.2	venv_qa
oauthlib	3.1.0	3.1.1	venv_qa
ordered-set	3.1.1	4.0.2	venv_qa
packaging	20.3	20.9ss	venv_qa
paho-mqtt	1.5.0	1.5.1	venv_qa

pathspec	0.7.0	0.8.1	venv_qa
pip	20.0.2	21.1.2	venv_qa
psutil	5.7.0	5.8.0	venv_qa
Pygments	2.6.1	2.9.0	venv_qa
pylint	2.4.4	2.8.3	venv_qa
pyOpenSSL	19.1.0	20.0.1	venv_qa
pyparsing	2.4.6	2.4.7	venv_qa
pytz	2019.3	2021.1	venv_qa
pywin32	227	301	venv_qa
PyYAML	5.3	5.4.1	venv_qa
regex	2020.2.20	2021.4.4	venv_qa
ruamel.yaml	0.16.10	0.17.7	venv_qa
ruamel.yaml.clib	0.2.0	0.2.2	venv_qa
s3transfer	0.3.3	0.4.2	venv_qa
setuptools	45.3.0	57.0.0	venv_qa
six	1.14.0	1.16.0	venv_qa
sqlparse	0.3.1	0.4.1	venv_qa
typed-ast	1.4.1	1.4.3	venv_qa
virtualenv	20.0.9	20.4.7	venv_qa
websocket-client	0.57.0	1.0.1	venv_qa
Werkzeug	1.0.0	2.0.1	venv_qa
wheel	0.34.2	0.36.2	venv_qa
wrapt	1.12.0	1.12.1	venv_qa
zipp	3.1.0	3.4.1	venv_qa
appdirs	1.4.3	1.4.4	venv_prod
asgiref	3.2.3	3.3.4	venv_prod
asn1crypto	1.3.0	1.4.0	venv_prod
astroid	2.3.3	2.5.7	venv_prod
attrs	19.3.0	21.2.0	venv_prod
azure-functions	1.2.0	1.7.2	venv_prod
boto3	1.12.16	1.17.88	venv_prod
botocore	1.15.16	1.20.88	venv_prod
certifi	2019.11.28	2021.5.30	venv_prod
cffib	1.14.0	1.14.5	venv_prod
chardet	3.0.4	4.0.0	venv_prod
colorama	0.4.3	0.4.4	venv_prod
cryptography	2.8	3.4.7	venv_prod
deepdiff	4.2.0	5.5.0	venv_prod
distlib	0.3.0	0.3.2	venv_prod
Django	3.0.4	3.2.4	venv_prod
docker	4.2.0	5.0.0	venv_prod
docutils	0.16	0.17.1	venv_prod
idna	2.9	3.2	venv_prod
importlib-metadata	1.5.0	4.5.0	venv_prod
isort	4.3.21	5.8.0	venv_prod
Jinja2	2.11.1	3.0.1	venv_prod

jmespath	0.9.5	0.10.0	venv_prod
jsonpickle	1.3	2.0.0	venv_prod
keyring	21.1.1	23.0.1	venv_prod
lazy-object-proxy	1.4.3	1.6.0	venv_prod
lxml	4.5.0	4.6.3	venv_prod
MarkupSafe	1.1.1	2.0.1	venv_prod
nipyapi	0.14.0	0.16.2	venv_prod
oauthlib	3.1.0	3.1.1	venv_prod
ordered-set	3.1.1	4.0.2	venv_prod
packaging	20.3	20.9	venv_prod
paho-mqtt	1.5.0	1.5.1	venv_prod
pathspec	0.7.0	0.8.1	venv_prod
pip	20.0.2	21.1.2	venv_prod
psutil	5.7.0	5.8.0	venv_prod
Pygments	2.6.1	2.9.0	venv_prod
pylint	2.4.4	2.8.3	venv_prod
pyOpenSSL	19.1.0	20.0.1	venv_prod
pyparsing	2.4.6	2.4.7	venv_prod
pytz	2019.3	2021.1	venv_prod
pywin32	227	301	venv_prod
PyYAML	5.3	5.4.1	venv_prod
regex	2020.2.20	2021.4.4	venv_prod
ruamel.yaml	0.16.10	0.17.7	venv_prod
ruamel.yaml.clib	0.2.0	0.2.2	venv_prod
s3transfer	0.3.3	0.4.2	venv_prod
setuptools	45.3.0	57.0.0	venv_prod
six	1.14.0	1.16.0	venv_prod
sqlparse	0.3.1	0.4.1	venv_prod
typed-ast	1.4.1	1.4.3	venv_prod
virtualenv	20.0.9	20.4.7	venv_prod
websocket-client	0.57.0	1.0.1	venv_prod
Werkzeug	1.0.0	2.0.1	venv_prod
wheel	0.34.2	0.36.2	venv_prod
wrapt	1.12.0	1.12.1	venv_prod
zipp	3.1.0	3.4.1	venv_prod

ANNEXURE G: ETHICAL CLEARANCE



Private Bag X1290, Potchefstroom
South Africa 2520

Tel: 018 299-1111/2222

Fax: 018 299-4910

Web: <http://www.nwu.ac.za>

Senate Committee for Research Ethics

Tel: 018 299-4849

Email: nkosinathi.machine@nwu.ac.za

ETHICS APPROVAL LETTER OF STUDY

Based on approval by the **Faculty of Natural and Agricultural Sciences Ethics Committee (FNAS-REC)**, the Faculty of Natural and Agricultural Sciences Ethics Committee hereby **approves** your study as indicated below. This implies that the North-West University Senate Committee for Research Ethics (NWU-SCRE) grants its permission that, provided the special conditions specified below are met and pending any other authorisation that may be necessary, the study may be initiated, using the ethics number below.

Study title: Evaluating time, scope and cost of human labor vursus digital process automation capabilities															
Study Leader/Supervisor: Dr W van Blerk															
Student: J Muller															
Ethics number:	N	W	U	-	0	1	7	7	9	-	2	0	-	A	9
	Institution				Study Number						Year		Status		
Status: S = Submission; R = Re-Submission; P = Provisional Authorisation; A = Authorisation															
Application type: Single				Risk Category:				Minimal							
Commencement date: 01/11/2020															
Expiry date: 01/02/2022															
Approval of the study is initially provided for a year, after which continuation of the study is dependent on receipt and review of the annual (or as otherwise stipulated) monitoring report and the concomitant issuing of a letter of continuation.															

Special in process conditions of the research for approval (if applicable):

- The following documentation are archived by FNASREC and should be complete and kept up to date:
 - Research proposal
 - Signed approval from the scientific committee indicating the proposed risk category
- All researchers involved in the study should submit signed NWU code of conduct statements annually.
- All researchers of low risk studies should submit proof of relevant ethics training every two years.
- All researchers that take part in activities that pose a safety and security threat to the researchers or the environment should submit a risk assessment form annually.
- All research involving human interaction should follow best ethical practise and keep documents as proof. This includes informed consent, questionnaires, incorporation of risk-benefit, and responsible data management.
- Any research at governmental or private institutions, permission must still be obtained from relevant authorities and provided to the FNASREC. Ethics approval is required BEFORE approval can be obtained from these authorities.

General conditions:

While this ethics approval is subject to all declarations, undertakings and agreements incorporated and signed in the application form, the following general terms and conditions will apply:

- *The study leader/supervisor (principle investigator)/researcher must report in the prescribed format to the FNASREC:*
 - *annually (or as otherwise requested) on the monitoring of the study, whereby a letter of continuation will be provided, and upon completion of the study; and*
 - *without any delay in case of any adverse event or incident (or any matter that interrupts sound ethical principles) during the course of the study.*
- *The approval applies strictly to the proposal as stipulated in the application form. Should any amendments to the proposal be deemed necessary during the course of the study, the study leader/researcher must apply for approval of these amendments at the FNASREC, prior to implementation. Should there be any deviations from the study proposal without the necessary approval of such amendments, the ethics approval is immediately and automatically forfeited.*
- *Annually a number of studies may be randomly selected for an external audit.*
- *The date of approval indicates the first date that the study may be started.*
- *In the interest of ethical responsibility, the NWU-SCRE and FNASREC reserves the right to:*
 - *request access to any information or data at any time during the course or after completion of the study;*
 - *to ask further questions, seek additional information, require further modification or monitor the conduct of your research or the informed consent process;*
 - *withdraw or postpone approval if:*
 - * *any unethical principles or practices of the study are revealed or suspected;*
 - * *it becomes apparent that any relevant information was withheld from the FNASREC or that information has been false or misrepresented;*
 - * *submission of the annual (or otherwise stipulated) monitoring report, the required amendments, or reporting of adverse events or incidents was not done in a timely manner and accurately; and / or*
 - * *new institutional rules, national legislation or international conventions deem it.*
- *FNAS-REC can be contacted for further information or any report templates via Roelof.Burger@nwu.ac.za 018 299 4269*

The FNASREC would like to remain at your service as scientist and researcher, and wishes you well with your study. Please do not hesitate to contact the FNASREC or the NWU-SCRE for any further enquiries or requests for assistance.

Yours sincerely,



Prof Roelof Burger
Chairperson Faculty of Natural and Agricultural Sciences Ethics Committee (FNASREC)

ANNEXURE H: CODE OF CONDUCT



NORTH-WEST UNIVERSITY
YUNIBESITHI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

Research and Innovation

CODE OF CONDUCT FOR RESEARCHERS

This code of conduct is applicable to all NWU researchers.

As a researcher of the North-West University (NWU), I subscribe to the rules of the NWU Senate Committee for Research Ethics (SCRE), all applicable policies of the NWU as well as all national and international laws and regulations applicable to my field of study. Furthermore, I commit myself to abide by the ethical principles and responsibilities as set out in the Singapore statement on Research Integrity (22 September 2010), in any and all research endeavours that I undertake as a researcher of the NWU.

The four major principles of research integrity to which I will adhere and that will guide my research are:

- Honesty in all aspects of research;
- Accountability in the conduct of research;
- Professional courtesy and fairness in working with others;
- Good stewardship of research on behalf of others.

Consequently I will also adhere to the following ethical responsibilities:

1. I will take responsibility for the originality and trustworthiness of my research.
2. I will stay abreast of and adhere to all institutional, national, and international laws, regulations, and policies applicable and related to my research.
3. I will at all times employ appropriate research methods, base my conclusions on critical analysis of the evidence and report my findings and interpretations fully and objectively.
4. I will keep clear and accurate records of all research that I have conducted in a manner that will allow verification and replication of my work by others, if applicable.
5. I will, where applicable, share my data and findings openly and promptly, in line with external funding rules. This will be done as soon as possible after I have had an opportunity to establish priority and ownership claims.
6. I will take responsibility for my own contributions to publications, funding applications, reports and other representations of my research. I will also and only include authors who meet valid authorship criteria.
7. I will acknowledge the names and roles of those who made significant contributions to my research in publications, including writers, funders, sponsors, and others, but do not meet authorship criteria.
8. In my peer reviews, I will provide fair, prompt and rigorous evaluations and I will respect confidentiality when I review others' work.
9. I will disclose all conflicts of interest (financial and other) that could compromise the trustworthiness of my work in research proposals, publications, public communications, and in review activities.
10. When I publically address a community in the spirit of academic freedom, I will in all stages base my professional comments on research findings (if applicable) and my expertise. I will distinguish between professional comments and opinions based on personal views.
11. Should any irresponsible research practices and/or research misconduct become known to me or brought under my attention, I will report such irresponsible research activities to the appropriate authorities.
12. I will respond to irresponsible research practices or conduct, by taking prompt actions as set out in the procedures of the university. I will also protect those who report misconduct in good faith, to the best of my abilities.
13. I will endeavour to create and sustain an environment that encourage research integrity through education of students, research teams and peers, as well as abide by policies, and reasonable standards for advancement.
14. I will at all times weigh societal benefits against the risks inherent in my work.

Name: Jacqueline Muller

Signature: *J Muller*

Date: 2020-11-11

Original details: (11664754) P:\9. Research and Post-graduate Education\9.1 Implementation of the research strategy\9.1.5 Ethics\9.1.5.1.3_Code_Conduct_2017.docm
18 July 2017

File reference: 9.1.5.1.3