Technical Debt Decision-Making under Uncertainty

by

Suwichak Fungprasertkul

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

August 2023

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

The key to effective technical debt management (TDM) decision-making is to select a feasible and conclusive decision on time. Decision-making, in general, can suffer from uncertainties, obscuring a problem domain and challenging one from producing a useful decision. However, the presence of uncertainties in TDM decision-making is inadequately explored in the Technical Debt (TD) community.

This thesis investigates the presence of Information Uncertainty (uncertainty due to information deficiency), Deep Uncertainty (uncertainty due to unresolvable disagreement) and Erratic Uncertainty (uncertainty due to unpredictability of a problem domain) and their consequences in TD identification, TD measurement, TD prioritisation and TD monitoring.

We leverage goal-obstacle and the expected value of information to strategically mitigate Information Uncertainty in TD identification and measurement. By leveraging these analyses, Information Uncertainty can be managed regardless of information availability.

We leverage Many Objective Robust Decision-Making to implement a framework to mitigate Deep Uncertainty during TD prioritisation. The framework's premise is to frame TD rankings in disagreements into different objective functions and utilise a set of decisions that perform well in those disagreements.

We propose a method to mitigate the consequence of Erratic Uncertainty in TDM monitoring: ineffective responsibility sharing. The method leverages Behavioural Cloning to filter the situation when developers/maintainers are necessary and utilises Replicator Dynamics to automatically resolve the TD monitoring situation when they are unnecessary, resulting in a more balanced workload between TD analysts and developers/maintainers.

The thesis individually evaluates the above contributions through various experiments with industrial cases, aiming for the effectiveness of the contributions compared to traditional approaches without the acknowledgement of uncertainties.

Acknowledgements

I would like to express my gratitude to my main supervisor, Rami Bahsoon, for his brilliant mind, infinite compassion and unparalleled dedication. His support, both as a supervisor and a friend, always have been there from my early PhD year, during COVID periods. His invaluable guidance, experience and insights are reminding gifts forever placed in my memory. Thank you very much to my external supervisor, Rick Kazman, for his always insightful and responsive feedback on my work. His kind and understanding attitude are to be respected and idolised.

I also would like to thank my supervisory team (Leandro L.L. Minku, Shan He and Xin Yao) for helping in shaping my PhD tenure in a way I could not possibly imagine. It is my honour to receive invaluable advice from them. I would like to thank the School of Computer Science, University of Birmingham, for facilitating me with incredible resources throughout my tenure. My PhD is fully funded by Royal Thai Government. I would like to thank all staff in the Office of Education Affairs for providing numerous scholars with physical and emotional shelter throughout difficult times.

I am eternally grateful to my extended family members, either in the form of humans or animals, closest of friends and partners, for lifting me up and providing me with aspirations when I needed the most. I did not know where my life was heading without them.

Finally, I would like to pay respect to family members who passed away before I can make them proud. I appreciate every one of them for paving the way for the family. I hope one day all of you can smile when looking at yours truly, down from heaven.

Contents

List of Figures viii

List of Tables x

1 Introduction 1

1.1 Backgrounds 1

1.1.1 Information Uncertainty 2

1.1.2 Deep Uncertainty 3

1.1.3 Erratic Uncertainty 4

1.2 Research Questions 5

1.3 Thesis Contributions 7

1.3.1 Goal-Obstacles-based Approach for Managing Information Uncertainty 7

1.3.2 MORDM-based Approach to Mitigate Deep Uncertainty . . . 9

1.3.3 Skin in the Game-driven automated method to alleviate Erratic Uncertainty 10

1.4 Scope and Assumptions 12

1.5 Thesis Outline and Publications 12

2 Overview: Managing Technical Debt Under Uncertainty 15

2.1 Introduction 15

2.2 Research Questions 16

2.3 Mapping Execution 18
2.3.1 Search Scope 18
2.3.2 Study Selection 20
2.3.3 Duplication Removal 21
2.3.4 Additional Search 21
2.3.5 Classification 22
2.3.6 Data Extraction 24
2.4 Comparative Analysis 26
2.4.1 Comparison Result 30
2.5 Results 38
2.5.1 Research Landscape (RQ 1) 38
2.5.2 Manifestation in TDM activities (RQ 2 & RQ 3) 38
2.5.3 Type of Uncertainty (RQ 4) 39
2.6 Reflection on Additional Causes of Uncertainty 40
2.7 Discussions 41
2.8 Moving Forward and Addressing the Gaps in Managing TD under Uncertainty 43
2.9 Threat to Validity 45
2.9.1 Inadequate Coverage of Study Search 45
2.9.2 Bias on Study Selection 45
2.9.3 The Lack of Quality Assessment 46
2.9.4 Bias in Data Extraction 46

3 Goal-Obstacles-based Approach for Managing Information Uncertainty 47

3.1 Introduction 47
3.2 Background 51
3.2.1 Uncertainty in TD Identification and Measurement 54
3.2.2 Health Economics 56

3.2.3 Mapping of Treatment Selection to Technical Debt Management Decision-Making . . . 57
3.3 Related Work . . . 59
3.4 Methodology . . . 60
3.4.1 Goal Modelling and Obstacles Analysis for Technical Debt . . . 62
3.4.2 Probabilistic Model of Technical Debt . . . 66
3.4.3 Scenario-based Framework for Technical Debt . . . 69
3.4.4 Expected Value of Technical Debt Information . . . 72
3.5 Evaluation . . . 74
3.5.1 Case and Experimental Setup . . . 76
3.5.2 RQ1: Predictability Analysis . . . 81
3.5.3 RQ2: Stability Analysis . . . 83
3.6 Discussions and Threat to Validity . . . 84
3.7 Conclusion . . . 88

4 MORDM-based Approach to Mitigate Deep Uncertainty 89
4.1 Introduction . . . 89
4.2 Background . . . 93
4.2.1 Deep Uncertainty and Robust Decision Making . . . 93
4.2.2 Black Swan . . . 94
4.2.3 MORDM . . . 97
4.3 Related Work . . . 100
4.4 Deep Uncertainty and Black Swan in TDM Decision-making . . . 101
4.4.1 Deep Uncertainty in TDM Decision-making . . . 101
4.4.2 Black Swan in TDM Decision-making . . . 105
4.5 Deep Uncertainty in TD Rankings . . . 122
4.5.1 Definition . . . 124
4.5.2 Black Swan Blindness Mitigation . . . 127
4.5.3 Scenario Elicitation . . . 130

4.5.4 Solutions Generation . . . 132
4.5.5 Vulnerabilities Discovery . . . 135
4.6 Case Study A: Apache Accumulo . . . 136
4.6.1 Scenario Elicitation . . . 137
4.6.2 Solutions Generation . . . 138
4.7 Case Study B: Apache Cocoon . . . 141
4.7.1 Scenario Elicitation . . . 141
4.7.2 Scenario Generation . . . 142
4.8 Evaluation . . . 143
4.8.1 Evaluation Setup . . . 146
4.9 Result and Discussions . . . 151
4.9.1 Research Question 1 . . . 151
4.9.2 Research Question 2 . . . 152
4.9.3 Research Question 3 . . . 153
4.9.4 Research Question 4 . . . 155
4.10 Threat to Validity . . . 158
4.11 Conclusion . . . 160

5 Skin in the Game-driven Automated Method to Alleviate Erratic Uncertainty 162

5.1 Introduction . . . 162
5.2 Background: Skin in the Game . . . 167
5.3 Related Work . . . 169
5.4 Approach to Skin in the Game in Technical Debt Management Decision Making . . . 171
5.4.1 Asymmetries in TDM Decision-Making . . . 171
5.4.2 Guideline for managing TD decisions with Skin in the Game . 176
5.5 Applying and Formalising Skin in the Game Thinking in Technical Debt Monitoring . . . 178

5.5.1 Three Steps Method for Applying Skin In the Game Thinking to TD Monitoring . . . 178
5.5.2 Asymmetries in TDM monitoring decision-making . . . 181
5.5.3 Weighting of monitored TD . . . 184
5.5.4 Automated Monitoring Corrective Actions . . . 186
5.6 Case Studies . . . 190
5.6.1 Case A: Apache Accumulo . . . 193
5.6.2 Case B: Apache Cocoon . . . 196
5.7 Evaluation . . . 196
5.7.1 Evaluation Setting . . . 199
5.7.2 Research Question 1 . . . 203
5.7.3 Research Question 2 . . . 204
5.7.4 Research Question 3 . . . 205
5.8 Threat to Validity . . . 205
5.9 Conclusion . . . 207

6 Reflection and Appraisal 209

6.1 Summarised Answers to Research Questions . . . 209
6.2 Dealing with Information Uncertainty . . . 216
6.3 Dealing with Deep Uncertainty . . . 217
6.4 Dealing with Erratic Uncertainty . . . 219
6.5 Applicability of Contributions . . . 221
6.6 Practicality of Contributions . . . 224

7 Future Directions 227

7.1 Exploration of Our Contributions in Specific Problem Domain . . . 227
7.1.1 Consideration on Specific TD type . . . 227
7.1.2 Application to Different TDM activities . . . 228
7.2 Improvement of Existing Contributions . . . 229

7.2.1 Automated Scenario Categorisation 229
7.2.2 Automated Decision Selection 230
7.2.3 Strategies Generation 231

Bibliography **232**

A Selected Studies in SLR **247**

List of Figures

2.1 Search Procedure for This Mapping Study 18

2.2 A Systematic Map of Study In Relation Of Research Types, Contribution Types, and TDM Activities 25

2.3 A Systematic Map of Study In Relation Of Uncertainty Types, Research Types, and TDM Activities 25

3.1 A high-level extract of AND goal refinement, SG1, SG2 of SGN. . . . 77

4.1 Accumulo: Scenario trees of the system components: a) Master b) TabletServer c) Garbage Collector 137

4.2 Example of an interactive parallel coordinates plot of generated solution. Blue lines indicate different solutions and their objective values, and vertical axes represents different objective functions. 139

4.3 Cocoon: Scenario trees of processes: a) URI Authentication b) Read/Write session context c) Pipeline Processing 141

5.1 Apache Accumulo: Replicator Dynamics of Three Strategies (Strategy A - Architectural Weighting Set, B - Functional Weighting Set, C - Severity Weighting Set). Population size plateau afterwards. . . . 195

List of Tables

1.1 Table of Research Questions 6

1.2 Table of Research Objectives 7

1.3 List of Chapters and Publication Progresses 14

2.1 Comparison Matrix of Studies 27

2.2 List of Publications 28

2.3 List of Primary Studies 30

3.1 Generalised source of uncertainty in TD identification and measurement 53

3.2 A Summary of Mapping of Obstacle and Technical Debt Using an Example: Performance Goal 64

3.3 Mapping of Obstacle and Its Corresponding Technical Debt Resolution Strategies 66

3.4 Table of Notation 67

3.5 Parameters of Identified Technical Debts 80

3.6 Table of Identified Scenarios 81

3.7 Table of File Properties 81

3.8 Ranking of technical debt decisions from Probabilistic model, Cost-saving-based (Csv) and Interest-based (Intr) approaches and its entropy 82

3.9 Expected potential benefit from all scenarios. their EVTPI is 33789.69 - 23238.17 = 10551.52 82

3.10 Ranking of Technical Debt Decisions by EVPPI and Interest-based Decision-making 82

3.11 Stability Indices of different TD decision-making approaches 84

3.12 Stability Indices of between EVPPI and interest-based approach . . . 84

4.1 Mapping of Levels of Uncertainty from [66], Supplemented with Examples in TDM based on Our Observations 95

4.2 Non-Exhaustive Potential Disagreement in Selected TDM Activities which can lead to Deep Uncertainty, Complemented by Examples from Our Case Studies 104

4.3 Elements of Black Swan Observed from [112] with Their Relevancy to TDM decision-making, Illustrated by Examples in general TDM decision-making and Observation/Analysis Of Our Case Studies . . . 106

4.4 Distinction of Extremistan and Mediocristan data Observed from [112], with Examples from Our Case Studies in Selected TDM activities 110

4.5 Characteristics of Extremistan Data Observed from [112] with Examples from Our Case Studies in Selected TDM Activities 111

4.6 Black Swan Blindness Observed from [114] with Examples in Selected TD Activities based on Our Case Studies 119

4.7 Possible Black Swan Blindness Mitigation Observed from [112] with Examples in Selected TD Activities based on Our Case Studies 128

4.8 Defined Scenarios based on Components Responsibilities 138

4.9 Project constraint and MOEA parameters 139

4.10 Accumulo: Fitness values of selected solutions 140

4.11 Accumulo: Pay-off matrix of selected solutions in fitness function $U(d, S1)$ against $U(d, S2)$ (Nash equilibria in bold) 140

4.12 Accumulo: Total Nash equilibria of selected solutions in different fitness function competitions 140

4.13 Cocoon: Fitness values of selected solutions 143

4.14 Accumulo: Statistical Properties of All Samples 149

4.15 Cocoon: Statistical Properties of All Samples 149

4.16 Accumulo: Result from Shapiro-Wilk tests of all samples 150

4.17 Cocoon: Result from Shapiro-Wilk tests of all samples 150

4.18 Accumulo: Result from Spearman's Rank Correlation tests of all samples and metrics 150

4.19 Cocoon: Result from Spearman's Rank Correlation tests of all samples and metrics 151

4.20 Accumulo: Result from Mann-Whitney U tests of all samples 151

4.21 Cocoon: Result from Mann-Whitney U tests of all samples 152

4.22 Result from Cross-Project Mann-Whitney U tests of all samples . . . 155

5.1 Suggested guidelines for alleviating Exclusive Information, Unfair Consequences, Capable minority and Incapable majority with examples in TD identification, measurement, prioritisation and monitoring. . . 177

5.2 Statistical Properties of Apache Accumulo and Apache Cocoon 193

5.3 Effort Percentage Change TD Backlog of Apache Accumulo 193

5.4 Apache Accumulo: Payoff Table for Three Different Weighting Sets. Each value represents the payoff of strategies in the row completed against strategies in the column. 195

5.5 Effort Percentage Change TD Backlog of Apache Cocoon 196

5.6 Mann-Whitney U Hypothesis Test Result of Cost Analysis Samples . 203

5.7 Statistical Comparison of Our Approach and Traditional Approach . 204

5.8 Mann-Whitney U Hypothesis Test Result of TDS Score Samples . . . 204

5.9 Mann-Whitney U Hypothesis Test Result of Jaccard Index Samples . 205

A.1 A Table of Selected Studies and their SLR facets 254

A.2 A Table of Selected Studies and their SLR facets (continued) 255

Chapter 1

Introduction

1.1 Backgrounds

Defined in 1992, Technical Debt (TD) is a programming phenomenon in postponing the mitigation of "not quite right code" [32]. Since then, the definition of TD has been expanded, covering various software artefacts, different software quality attributes and TD admission behaviour. When the postponement is properly utilised, TD provides flexibility to achieve development goals in lower project constraints [61]. In contrast, overburden TD can halt the value of developing software and, in the worst case, put the existence of the developing software at stake [80]. Technical Debt Management (TDM) is an essential set of activities to ensure that TD is timely and properly mitigated, with the goal to enable decision-making regarding "how" and "when" to mitigate TD [42]. Activities in TDM include TD identification, TD measurement, TD prioritisation, TD monitoring, TD communication, and TD repayment [74].

In each TDM activity, decision-makers are required to make a choice, ensuring

an optimal decision is chosen and under project constraints. For instance, decision-makers must decide which and how to repay each TD instance, aiming that severe TD is repaid timely within the project budget and the project timeframe. However, there is a knowledge gap in the TD community regarding uncertainty during TDM decision-making. Without the acknowledgement of uncertainty in TDM decision-making, decision-makers may select a decision which is imprecise, inadequate, or myopic to the problem domain, contributing to a sub-optimal decision and deterring the quality of TDM. Based on the systematic mapping study (Chapter 2), we found that classes of uncertainties during TDM decision-making are as follows:

1.1.1 Information Uncertainty

This type of uncertainty occurs when additional information is mandatory for effective decision-making. Causes of this type of uncertainty can relate to incomplete information and/or insufficient data [127]. Information Uncertainty is similar to Code Uncertainty in [56], where more information is required to receive a clearer view of the computerised model.

Information Uncertainty can exist in any phase of TDM because information inadequacy can occur in any activities which require information, data, or experience. Problems in several TDM studies are rooted in Information Uncertainty. For instance, the lack of information on different TD types and their manifestation can lead to an improper TD model, potentially leading to hidden TD [62]. Another example is uncertainty in TD parameters, which is caused by the lack of information on reliable TD tools/measurements [50]. Information Uncertainty problems in TDM are usually alleviated by information investment, that is, allocating resources to obtain information in the area of uncertainty (e.g. the involvement of TD identification by developers in different roles [62] or utilising different TD measurements

methods and comparisons [50]).

However, the existence of Information Uncertainty in TDM and its link to information investment is not explicitly explored. The gap in knowledge renders Information Uncertainty to be managed in an ad-hoc manner, which can be ineffective and labour-intensive. Ill-managed Information Uncertainty can contribute to sub-optimal TD decisions. Mismanaged Information Uncertainty regarding TD identification tools (e.g. using tools which focus only on one TD aspect) can lead to improper TD tools utilisation. Improper use of TD identification tools can help in concealing TD, supporting its ill-timed repayment. Mistreated Information Uncertainty on the TD impact measurement (e.g. solely depending on one measurement method) can contribute to over- or underestimation. If this issue is undetected, it can contribute to inefficient or ineffective TD repayment.

1.1.2 Deep Uncertainty

As defined by [126], Deep Uncertainty exists when parties to the decision do not agree on the model representation of the problem. These conflicts stem from interpretations or opinions based on information or knowledge that come from different sources, different timeframes and/or focus on different dimensions. Disagreement in TDM decision-making is not uncommon, and Deep Uncertainty from this cause can exist throughout TDM. Effective TD decisions can relate to the problem context, the system environment, available resources, developers' competency, etc. Decision-makers can have different perceptions of those deciding factors, which potentially result in different stances on a suitable decision. Deep Uncertainty in TD decisions results in indecision, which can be mitigated when decision-makers mutually accept one of proposing decisions. This can be achieved when a disagreement resolution is utilised.

However, decision-makers still can suffer from Deep Uncertainty due to ineffective disagreement resolution. Decision-makers may be unable to borrow a recommended decision from the TD literature due to the lack of unified guidance in the TD community. For example, they may be unable to decide whether to consider the security impact of TD [51] because there is no consensus on TD factors and measurement approaches [69]. Most importantly, research on TD disagreement resolution strategy is unexplored, and decision-makers, therefore, may utilise traditional strategies (e.g. voting, compromising, anchoring) to mitigate disagreement in TD decisions.

Ineffective resolution can result in a decision which is myopic, biased, overly- or under-focused on specific dimensions. These issues can degrade the quality of decisions depending on the TDM area where the disagreement resides. For instance, a myopic TD identification approach can lead to hidden TD, a biased TD measurement approach can over- or underestimate TD parameter(s), and a consensus-based TD prioritisation approach can contribute to a sub-optimal repayment strategy.

1.1.3 Erratic Uncertainty

This category of uncertainty is caused by the unpredictable nature of the problem domain, which is irreducible within the given time and resource. Unpredictability can be inherited from the system behaviour, the interaction between system components, or agents affected by decision-making. Uncertainty from the unpredictability is based on [127], and it is similar to Residual Variability in [56]. The unpredictability in the manifested system, components, their interactions and/or TD parameters can contribute to the unpredictability in TDM decision-making. Erratic Uncertainty in TDM decision-making can relate to one or many TDM sub-activities, including the changing environment, the existence of TD, TD parameters, the effectiveness of repayment strategies, TD monitoring criteria, or the combination of several aspects.

TDM problems rooted in Erratic Uncertainty are usually mitigated by utilising assumptions. However, this technique is problem-driven, which can lead to a gap in Erratic Uncertainty mitigation. [40] applies Portfolio Theory to manage uncertainty from the growth of TD in the backlog. This study assumes that the unpredictability of TD interest follows the normal distribution. It requires further research and validation to ensure that this can be effectively applied to other TD problems of Erratic Uncertainty. In addition, this technique can have demanding prerequisites. This is evident in the application of real options theory in handling the unpredictable nature of requirement TD [1], where knowledge of real options theory is required to estimate the volatility factor, rendering this inapplicable in unknowledgeable stakeholders.

Due to these limitations, a solution for Erratic Uncertainty can be inadequately assessed and improperly utilised. For instance, a TD identification approach that emphasises code-based TD can indirectly support the manifestation of TD in other types. A TD measurement method with a high learning curve can result in misestimation of TD parameters. A TD monitoring approach without consideration of its feasibility may cause an overwhelming workload of developers/maintainers, which can influence the approach's effectiveness. These limitations are supported by a lack of study on the presence of Erratic Uncertainty in TDM decision-making.

1.2 Research Questions

To investigate uncertainty in TDM decision-making, it is essential to first explore possible classes of uncertainties in decision-making that are applicable to TDM. *The presence of uncertainties in TDM decision-making* can be investigated from observations on their behaviours, their causes, and manifestation nature. Once those characteristics are identified, it is possible to study current TDM issues which con-

Table 1.1: Table of Research Questions

Classes of Uncertainties	RQ No.	Description
Information Uncertainty	1.1	What are the causes of Information Uncertainty in TDM decision-making?
	1.2	What are TDM problems rooted in Information Uncertainty? Which TDM activities are affected by it?
	1.3	How to tackle Information Uncertainty in TDM? What are the benefits of the said mitigation?
Deep Uncertainty	2.1	What is the origin of Deep Uncertainty in TD decision making?
	2.2	How does Deep Uncertainty affect TDM decision-making? Which TDM activities can suffer from Deep Uncertainty?
	2.3	How to handle Deep Uncertainty in TDM decision-making? What is the difference between the said mitigation and without using one?
Erratic Uncertainty	3.1	What are the causes of Erratic Uncertainty? How can Erratic Uncertainty be manifested in TDM decision-making?
	3.2	Which TDM activities are vulnerable to Erratic Uncertainty? What are their implications?
	3.3	How to mitigate implications from Erratic Uncertainty in TDM decision-making? What are the advantages or disadvantages of the mitigation, comparing an approach without one?

tribute to *the manifestation of uncertainties in TDM decision-making*, including the current TD knowledge gap, limitation on existing TD literature/tools, and ongoing research challenges in the TD community. Results from this facilitate us to map those issues to relevant TDM decision-making areas, supporting discovery on *problems rooted in TDM decision-making uncertainties* in specific TDM activities. As the above problems are grounded in specific TDM activities, it is easier to observe problem contexts and their consequences in detail. Those are incorporated in search of *alleviation from TDM decision-making uncertainties*. Collectively, our core research questions are as follows:

What are the classes of uncertainties that can face TDM decision-making? How do these uncertainties affect the quality of TDM? What are the benefits of these techniques and frameworks compared to traditional approaches without acknowledgement of these uncertainties?

These questions are elaborated in Table 1.1, where the above questions are discussed in relation to Information Uncertainty, Deep Uncertainty and Erratic Uncertainty. Table 1.2 illustrates the objectives to achieve the detailed research questions in Table 1.1, providing precise, actionable research activities.

Table 1.2: Table of Research Objectives

RQ No.	Objectives
1.1	Investigate the definition of Information Uncertainty, its behaviour and its possible mitigation. These findings should indicate the relevancy of Information Uncertainty to TDM decision-making.
1.2	Explore ongoing problems in TDM activities which can be vulnerable to Information Uncertainty. The result from this exploration should identify TDM activity prevalence to Information Uncertainty and their potential implication.
1.3	Identify decision-making techniques which can mitigate Information Uncertainty. Build a mitigation approach upon these techniques. The approach should alleviate the impact of Information Uncertainty in TDM.
2.1	Study the causes of Deep Uncertainty in decision-making. Look into characteristics of Deep Uncertainty, which can be prevalent in TDM decision-making.
2.2	Investigate ongoing TD challenges and the nature of different TDM activities, which support the manifestation of Deep Uncertainty and its implications. Findings from these should indicate possible implications of Deep Uncertainty in different TDM activities.
2.3	Look into decision-making techniques utilised to mitigate Deep Uncertainty. Based on these techniques, build an approach that minimises the presence of Deep Uncertainty in TDM decision-making.
3.1	Explore studies on the presence of Erratic Uncertainty and its causes. The result of this exploration should indicate the relevancy of Erratic Uncertainty to TDM decision-making.
3.2	Mapping causes of Erratic Uncertainty to vulnerable TDM activities by observing responsibilities for TDM activities, their ongoing challenges and knowledge gap.
3.3	Investigate decision-making techniques which help in mitigating Erratic Uncertainty. Create an approach based on these techniques to minimise the implication of Erratic Uncertainty.

1.3 Thesis Contributions

This thesis aims to investigate the presence of Information Uncertainty, Deep Uncertainty and Erratic Uncertainty and their implications in TDM decision-making. The following contributions are presented to alleviate problems we discovered that are rooted in these uncertainties.

1.3.1 Goal-Obstacles-based Approach for Managing Information Uncertainty

We investigate the presence of Information Uncertainty in TDM decision-making, by observing possible types of uncertainties which can be encountered in computerised models. As there exists a lack of guidance on the TD identification models and TD measurement approaches [69], ad-hoc techniques may be utilised during their decision-making, inviting several types of uncertainties into the system. We explore different types of uncertainties which can occur in TD identification and TD

measurement. One of the fatal classes of uncertainties is Information Uncertainty, where it can reside in TD models, approaches, and parameters, and it can relate to several sub-activities in TD identification and TD measurements. Informed by software engineering studies on decision-making under uncertainty [70, 108, 24], we look for the role of information during decision-making in complex environments, mainly health economics, to draw inspiration to bridge the knowledge gap that can help in systematically mitigating the presence of Information Uncertainty in TDM decision-making. Decision-making in TDM is likened to that of health economics, as both take advantage of imperfect information and should be evidence-based. Most importantly, both subjects require a systematic management approach to mitigate information-based uncertainty, as it can exist in several areas at a given time.

We provide an information-centric TD identification and measurement approach, enabling practitioners to reduce Information Uncertainty in a systematic fashion. In the immediate term, we present a probabilistic TD model, helping practitioners consider the cruciality of TD to the system. The model is based on a novel use of goal and obstacles analysis in requirements engineering [123, 124] to produce additional information, emphasising the importance of TD in relation to the obstruction of a system goal, its impact on the system quality attributes and their likelihood to occur. To complement the above, we leverage scenario analysis to produce an intermediate tactic to reduce Information Uncertainty. Scenario analysis, commonly used in requirements engineering [110], is novelly utilised to narrow a range of TD parameters based on system scenarios, producing more precise information on a corresponding TD. For long-term mitigation, we provide a novel use of the expected value of information analysis to help in prioritising TD and its Information Uncertainty. The ranking produced from the analysis can supplement a mitigation plan, thereby mitigating Information Uncertainty in TD that yields the most potential monetary benefit.

1.3.2 MORDM-based Approach to Mitigate Deep Uncertainty

Decision-makers, with diverse experience, knowledge, and speciality, may have different scenarios of the problem, contributing to different opinions on a suitable solution. A disagreement from this, when irresolvable, can be considered as Deep Uncertainty.

We look into the presence of Deep Uncertainty in TDM activities, covering TD identification, TD measurement and TD prioritisation. Disagreement rooted in TD identification, TD measurement and decision-making related to TD rankings can propagate to TD prioritisation. Disagreements from those TDM activities can relate to resource allocation, approaches/models selection and their configurations. Due to the lack of disagreement resolution guidelines in TDM decision-making, Deep Uncertainty in the above TDM activities can be improperly mitigated. Consequently, a myopic, biased or anchored decision may be produced, influencing a Black Swan event to occur. Leveraged by the Black Swan theory [112], we present possible Black Swan blindnesses during decision-making in TD identification, TD measurement and TD prioritisation. These blindnesses are fundamentally rooted in the misuse of evidence, narration or claims during decision-making. We present a decision-making guideline for TD identification, TD measurement and TD prioritisation. The premise of this guideline is for decision-makers to mitigate and document these blindnesses when Deep Uncertainty is unavoidable. The guideline encourages the use of falsifiable, comprehensive, concrete evidence during decision-making over narration or claims and explicitly documenting the limitation of evidence, claims and narration for future reference.

To complement this guideline, MORDM (Many Objective Robust Decision Making)-based approach is presented to mitigate Deep Uncertainty from its root cause. MORDM was previously adopted to produce robust solutions in Environment Engineering [53]. This novel use of MORDM aims to systematically resolve disagreement

manifested in TD prioritisation decision-making by selecting a set of TD rankings that are acceptable in different scenarios. Our approach starts with the novel use of Formal Scenario Analysis, previously used in requirements engineering [46], to ground disagreements as scenarios and their utility score. MOEA (Many Objective Evolutionary Algorithms) is then utilised to help practitioners explore possible solutions which perform well in different scenarios. The performance of each solution is indicated by MOEA objectives, consisting of utility scores in different scenarios. Based on the concept of optionality [113], we encourage practitioners to choose a set of solutions as options to be interchangeably exercised in case there is a change outside of grounded scenarios. A mapping of a solution to a specific scenario can be visualised with payoff matrices, where practitioners observe Nash equilibria [86] of opposing solutions in different scenarios.

1.3.3 Skin in the Game-driven automated method to alleviate Erratic Uncertainty

We investigate the presence of Erratic Uncertainty in TDM by leveraging Skin in the Game practises [114]. We discovered that negative consequences from Erratic Uncertainty exist when a mitigation decision imposed by a minority of decision-makers is collectively accepted. Especially when the minority is more intolerant to the problem, has more information, and gains benefit from a decision more than the majority. When a decision with Erratic Uncertainty is untreated, it can contribute to a decision with ineffective responsibility sharing (e.g. an unfair decision/assignment). Ineffective responsibility sharing divides decision-makers by capability in assessing the decision, creating mistrust or bias towards different factions and deterring the quality of TDM. These issues can occur in various TDM activities, including TD identification, TD measurement, TD prioritisation and TD monitoring.

We present a guideline for decision-making in TD identification, TD measurement, TD prioritisation and TD monitoring. The premise of this guideline is to facilitate decision-makers in identifying situations that can lead to ineffective responsibility sharing and mitigate them before a decision is produced. The guideline fundamentally encourages decision-makers to participate in fair and transparent decision-making, and to collectively acknowledge sources of imposing decisions and their possible implications. The guideline recommends the use of documentation during decision-making for traceability.

We investigate potential problems rooted in ineffective responsibility sharing in TD monitoring. This TDM activity involves different roles of stakeholders, including TD analysts, developers, and maintainers, rendering it vulnerable to ineffective responsibility sharing. We discover that with the lack of weighting in each TD instance, and the limited actions from TD monitoring results, maintainers and developers can suffer from ineffective responsibility sharing, resulting in imbalanced TD monitoring workloads.

To mitigate this, we present an automated approach that only notifies developers and maintainers when their interventions are needed. This approach is built on our model of weighted percentage change in monitoring TD over time. This model enables practitioners to indicate the importance of each TD instance, filtering only the percentage change in significant TD to be notified. The approach introduces an automated action from TD monitoring results. This automated action is based on human demonstrations, observed by Behavioural Cloning [94]. Once the automated action is initiated, it recommends mitigation plans to developers and maintainers. These plans are archived by developers or maintainers and chosen by Replicator Dynamics [102].

1.4 Scope and Assumptions

Below we describe the scope of this thesis and assumptions:

- We scope our investigation to commonly discussed and acknowledged TD components (TD principal, TD interest amount and TD probability [98]) to maintain the consistency with the TD community and to avoid introducing overlapping terms and concepts [69].

- We take the view that TDM activities consist of TD identification, TD measurement, TD prioritisation, TD prevention, TD monitoring, TD repayment, TD representation or TD documentation, and TD communication, and inline with the TD community [75].

- We view TDM decision-making as a collective activity participated by decision-makers. TDM decision-making refers to an activity, which produces a TDM decision, residing throughout the TDM lifecycle. Each decision-maker is a stakeholder in the software system, regardless of their responsibility to the project. The rationale for the assignment of decision-makers is subject to the stakeholders and is out of the thesis scope.

1.5 Thesis Outline and Publications

The chapters of the thesis are outlined as follows:

Chapter 2 illustrates the gap of knowledge, ongoing challenges and the limitation on the study of uncertainties in the context of TDM.

Chapter 3 discusses the presence of Information Uncertainty in TD identification and measurement, compassing its behaviour, vulnerable areas in the above TDM activities, our approach to alleviating them, and the evaluation of our approach.

Chapter 4 introduces the concept of Deep Uncertainty, its manifestation in TD identification, TD measurement and TD prioritisation, its consequences in the above TDM activities, our framework to alleviate Deep Uncertainty in TD prioritisation, and the framework's evaluation.

Chapter 5 discusses the causes and consequences of the lack of effective responsibility sharing in TD identification, TD measurement, TD prioritisation and TD monitoring. This chapter indicates the ongoing TD monitoring problem rooted in the lack of effective responsibility sharing, our method to alleviate the problem and its evaluation.

Chapter 6 reflects on our contributions and addresses their limitations.

Chapter 7 explores directions for future work.

Table 1.3 summarises progress towards publishing the core contribution chapters of the thesis and the considered publication venues. In various parts of the thesis, formal equations are employed where they help clarify and represent concepts more effectively. The use of formal methods serves several purposes. First, formalisation provides a better understanding of how contributions are developed and derived from associated publications. Second, formalisation ensures that our contributions are represented as objectively as possible. This benefit facilitates the replicability of our contributions, which in turn can encourage one to investigate, evaluate or practical adaptation in the future.

Table 1.3: List of Chapters and Publication Progresses

Chapter No.	Publication Venues
Chapter 2	Partially published in IEEE Software, vol. 38, no. 6, 2021 under the title of "Managing Technical Debt Under Uncertainty" by S.Fungprasertkul and R.Bahsoon. Extended version to be submitted as a survey journal.
Chapter 3	Under submission to IEEE Transactions on Software Engineering under the title of "Managing Technical Debt with Information Uncertainty" by S.Fungprasertkul, R.Bahsoon and R.Kazman
Chapter 4	To be submitted to a journal under the title of "Managing Deep Uncertainty in Technical Debt Prioritisation Using Black Swan Theory" by S.Fungprasertkul and R.Bahsoon
Chapter 5	Accepted to ACM Transactions on Software Engineering and Methodology under the title of "Technical Debt Monitoring Decision Making with Skin in the Game" by S.Fungprasertkul, R.Bahsoon and R.Kazman

Chapter 2

Overview: Managing Technical Debt Under Uncertainty

2.1 Introduction

In 1992, Ward Cunningham defined technical debt (TD) as a phenomenon in postponing the correction of "not quite right code" [32]. Since then, the technical debt community has advanced our understanding of the topic, and this definition has been expanded to consider various facets of the software development life cycle, with a focus on maintainability and evolvability of the software product [61]. Research has looked into technical debt in relation to requirements, architectures, documentation, testing, security, and self-admission, among the many topics. Several approaches that relate to TD identification, measurement, prioritisation, repayment, and interest have also been proposed [74].

While ten years of TD research has accelerated our understanding of the topic,

the knowledge gap between different TD concepts, approaches, and their practical usage still exists [98]. Practical application of the technical debt approaches will not be meaningful without factoring in various uncertainties that confront the technical debt management decision-making process. Additionally, if uncertainty is not systematically elicited, factored in, analysed and mitigated, Technical Debt Management (TDM) can fundamentally suffer, which inevitably contributes to ineffective and imprecise processes and indecision.

The chapter presents a mapping study that is aimed at locating references, approaches, solutions and tooling that have implicitly or explicitly considered uncertainties in TDM. The mapping study provides comprehensive coverage of papers appearing in mainstream conferences (e.g. ICSE, TDM and Euromicros (SEaTeD) etc.), mainstream Software Engineering journals and tooling. Additionally, documented findings of systematic surveys, tertiary and mapping studies by the TD community (e.g., [69, 98, 74]) were particularly useful for coverage adequacy and when synthesising our findings.

2.2 Research Questions

The primary objective of this chapter is to investigate an overview of the current research on uncertainty in TD. This objective can be answered in different aspects, which we categorised as research questions below:

RQ1. What is the current research landscape in relation to uncertainty in TD?

Rationale: This RQ is related to types of research and their contributions related to uncertainty in TD, which researchers/practitioners are working on. A type of research refers to categories of research depending on the formality of their case,

evaluation, validation or etc. The answer to this question can help us investigate the maturity of the research landscape on uncertainty in TD.

RQ2: Which TDM activities can suffer from uncertainty?

Rationale: The presence of uncertainty in TD may manifest in several TD areas. One common perspective is to identify TDM areas based on TDM activities. This research question investigates which TDM activities can suffer from uncertainty, according to researchers/practitioners. By answering this RQ, we can know which TDM activities that are given the most attention in the TD community.

RQ3: In which TDM activities uncertainty is under-studied?

Rationale: As not all TDM activities might receive the same attention from the TD community, this RQ is related to TDM activities where the presence of uncertainty is under-explored or unacknowledged. The answer to this RQ can shed light on research gaps in TDM activities on uncertainty in TD, potentially leading to future research directions.

RQ4: What are the causes of uncertainty in TDM activities?

Rationale: Uncertainty can be differently defined depending on the problem domain, the environment, the research field, perceivers, etc. This RQ explores how researchers and practitioners identify the cause of uncertainty in behaviours of TD, TD attributes, TD problem context, and TD environment in different TDM activities. The answer to this problem sheds light on the current state of study on different types of uncertainty in TDM activities and informs researchers and practitioners of possible types of uncertainty in TDM activities that they might be unaware of.

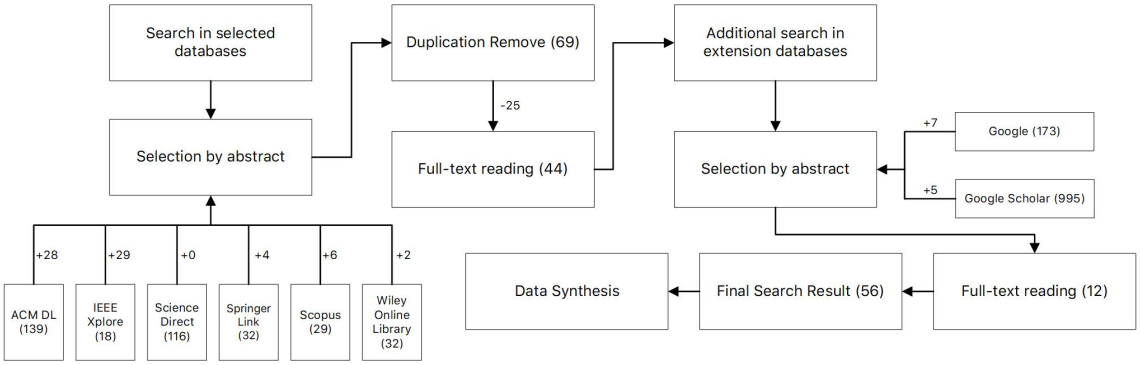


Figure 2.1: Search Procedure for This Mapping Study

2.3 Mapping Execution

In order to answer the above RQs, we initiated our mapping study procedure following the guideline by Petersen et al. [93]. Figure 2.1 illustrates our procedure, sources of electrical databases and the number of studies throughout the procedure. The detail of the procedure as addressed in the following subsections:

2.3.1 Search Scope

We utilise an automatic search strategy to look for studies which are relevant to this mapping study. Although we considered manual searches in specific relevant journals or publication venues (e.g. TechDebt) as a potential strategy, we decided not to do so as there might be studies on uncertainty in TD that exist outside those publication venues. We utilise PICO (Population, Intervention, Comparison and Outcomes) process to identify and formulate our search strings. This technique is suggested by Kitchenham and Charters [57]:

- **Population:** The population of our mapping study is "technical debt" as we consider studies that directly relate to the problem in the technical debt domain.

- **Intervention:** For research design outside of clinical settings, intervention refers to the application or theory in mind. For our mapping study, the intervention keyword is "uncertainty" as uncertainty is the TD application of our interest.
- **Comparison:** As comparison is irrelevant to RQs, we do not consider comparisons between different studies in our mapping study.
- **Outcomes:** We intend to observe outcomes from studies for answering RQ1. Thus, we do not have an outline for a specific outcome.

Based on the PICO process, our search string is as follow:

- **Technical Debt Uncertainty:** ("technical debt") AND ("uncertainty" OR "uncertainties")

We intentionally only utilise "Technical Debt" as its full term because the combination of its shortened term (e.g. "debt", "debts" or "TD") and "uncertainty" OR "uncertainties" can distort our findings, where studies outside of the technical debt or software engineering domain are included. We believe "Technical Debt" in its full term is adequate to search for studies which refer to a specific form of TD (e.g. self-admitted debt, test debt, etc.) as those studies mentioned "Technical Debt" in its full term as their reference.

We scope the search of our studies from the year 1992 (i.e. the year that technical debt is defined by Cunningham [32]) to the year 2023 (i.e. when we started this chapter).

Before we commenced our formal search, we established a preliminary search to determine the feasibility of our search string to yield acceptable search results. We

utilise the electrical databases suggested by Chen et al. [28]. From the preliminary search, EL Compendex and Web of Science are excluded as we do not have institutional access. In addition, we decided to exclude CiteSeer as its search algorithm produces the significant number of irrelevant studies with the above search string. From our preliminary search, we have received more than 5000 studies with CiteSeer, most of which are unrelated to the technical debt or software engineering domain. To ensure that we cover studies which may exclusively include in those databases, we utilise Google and Google Scholar for our additional searches, which we additionally execute after the formal search (detail in Section (2.3.4)).

2.3.2 Study Selection

After the search scope is established, we select our studies based on the following inclusion and exclusion criteria:

Inclusion Criteria:

- Paper, opinion piece, experience/technical report or grey literature that is related to technical debt.
- Paper, opinion piece, experience/technical report or grey literature that reports uncertainty in TD.

Exclusion Criteria:

- Study that is exclusively available in the form of abstract, conference/workshop summary, tutorial, presentations, audio or video.

Inclusion criteria conform to the PICO process above, whereas exclusion criteria are based on the unavailability of the study. We do not explicitly exclude studies in non-English language as our search string is efficient enough to filter non-English studies.

Our mapping study does not include selection by metadata alone. From our preliminary search, selection by metadata alone filters out a significant number of TD studies that implicitly refer to uncertainty in different terms throughout the study. Henceforth, our initial selection starts with selection based on metadata and abstract, and those inclusion and exclusion criteria are utilised during the selection, whereby the first inclusion criterion is utilised during the search by metadata, and the second inclusion and exclusion criteria are utilised during the selection by abstract.

2.3.3 Duplication Removal

We have found some studies with the same title, same authors, and same abstracts occur in different electronic databases. To resolve this, we prioritise our selection to the one which is published in the later year. If those studies are published in the same year, we prioritise the one which has more citations.

2.3.4 Additional Search

We selected Google and Google Scholar for additional searches. We specifically selected Google to investigate studies based on practitioners' experiences or opinions, which were recorded in the form of a website, blog, social platform, etc. To mitigate the limitation in the above electrical databases (mentioned in Section 2.3.1), Google Scholar is selected to search for additional studies. We utilise the same

search scope and selection criteria mentioned in the previous sections. Although we received 2,560 search queries from Google Scholar, only 995 were investigated due to the limitation of Google Scholar (i.e. only showing the results until the 100th page). We only include studies which do not have the same title, same authors and same abstracts from the previous search results. Thus, duplicated studies are pre-emptively excluded.

2.3.5 Classification

After full-text reading of the search results, we establish classification schemes for selected studies. Those classification schemes are necessary for the mapping study and in answering our RQs. Classification schemes are created following the iterative process suggested by Bailey et al. [15]. For our mapping study, four facets are created, and each helps in answering different RQs. One of the facets is the classification of research approaches, based on Wieringa et al. [129]. This facet is related to RQ1 and can be summarised as follows:

- **Validation Research:** Novel research that has not been utilised by practitioners and has been evaluated by an experiment.
- **Evaluation Research:** Research which is related to problems in industry, utilised by practitioners and evaluated with a real-world project.
- **Solution Proposal:** Research which proposes a solution to a specific problem. A solution is shown in a set of examples or a sound and valid argument.
- **Philosophical Paper:** A paper which provides a novel perspective on the existing problem.

- **Opinion Paper:** A paper expresses the existing industrial/research challenges and/or recommends the existing industrial/research challenges should be mitigated.
- **Experience Report:** Report which explains a problem in the industry and/or what and how to mitigate the problem in practice.

To answer RQ1, we created a contribution facet. This facet is based on Peterson et al.[93], where a contribution irrelevant to our search results (i.e. Metric) is removed, and contributions which are found in our search results at least two times are introduced.

- Tool
- Model
- Method
- Process
- Research Guideline (Introduced based on our search results)
- Taxonomy (Introduced based on our search results)
- Report (Introduced based on our search results)

To answer RQ2 and RQ3, we utilise TDM activities provided by Li et al. [74] as our TDM activities facet:

- TD identification
- TD measurement

- TD prioritisation
- TD representation
- TD communication
- TD repayment
- TD monitoring
- TD prevention

To answer RQ4, we created the classification of uncertainty. This classification is based on common causes of uncertainty we have discovered in search results. Each cause of uncertainty is based on the definition of uncertainty in different scientific domains:

- **Information Deficiency:** Uncertainty which can be alleviated with additional information [127].
- **Conflicting Perspectives:** Uncertainty which stems from irresolvable conflicts on optimal decisions, which is similar to Deep Uncertainty [126].
- **Unpredictability:** Uncertainty which stems from the unpredictable nature of the problem domain [127].

2.3.6 Data Extraction

After classification schemes are set, we map our findings into classification schemes. The details of selected studies are shown in Appendix A. In our mapping study, the mapping is represented by two bubble grid charts. The first one (Figure 2.2),

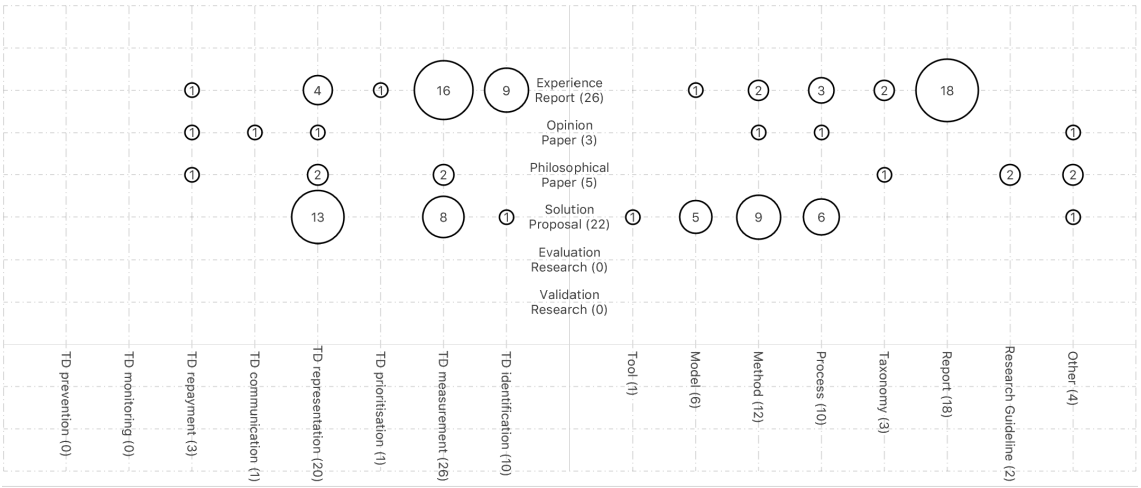


Figure 2.2: A Systematic Map of Study In Relation Of Research Types, Contribution Types, and TDM Activities

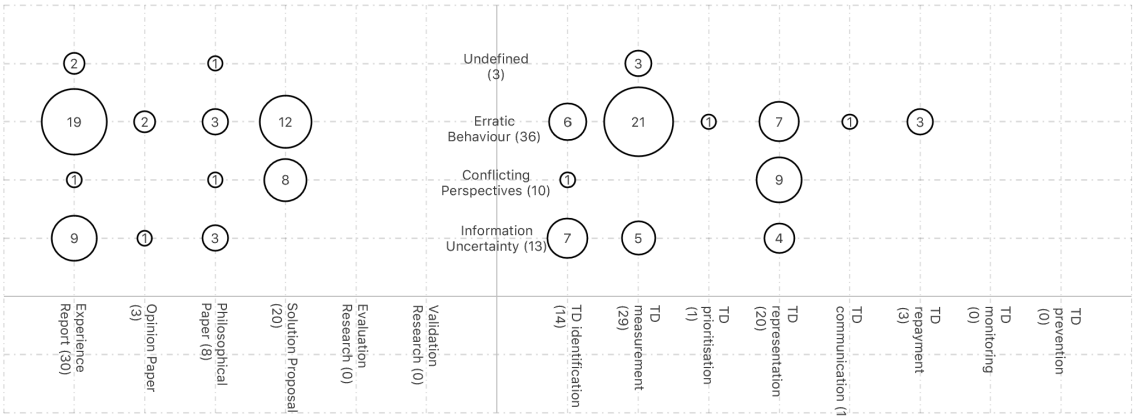


Figure 2.3: A Systematic Map of Study In Relation Of Uncertainty Types, Research Types, and TDM Activities

which helps in answering RQ1 to RQ3, is related to research types for research on uncertainty in TD, their related TD activities and their contributions. The second one (Figure 2.3), which primarily supports answering RQ4, represents types of uncertainty in relation to manifested TDM activities and research types. We do not map types of uncertainty to contribution facets as it is irrelevant to RQs.

2.4 Comparative Analysis

We compare and discuss studies considering uncertainty in TD and the diverse sources. We characterise those studies based on their maturity, their scope in investigating the problem of uncertainty in TD, and their relevance to TD types. We use Petersen et al. [93] classification scheme for completeness, relevance, and applicability, as described below:

- **Scope and maturity/completeness:** The aim is to identify studies with adequate coverage of the problem of uncertainty in TDM, covering one or more of TDM phases. We consider publications in which '**C1**- Identify a cause of uncertainty' and possibly its behaviour', and '**C2** - Provide an alleviation to a problem that is rooted in uncertainty by proposing a solution to mitigate it'. It is possible for a study to achieve C2 without C1, especially when that study proposes solution(s) for a problem identified in previous studies.

- **Relevancy:** This factor relates to the relevancy of the work to software engineering TDM. We consider publications describing '**R1** - the source of uncertainty in software development, process, or environment leading to TD' and '**R2** - uncertainty discussed in relation to TD attributes, properties, and/or TDM activities.

Table 2.1: Comparison Matrix of Studies

ID	C1	C2	R1	R2	A1	A2
S1	X	X	X			
S2			X	X	X	
S3	X		X	X	X	
S4	X		X	X	X	
S5					X	
S6	X		X	X	X	
S7	X	X	X	X	X	
S8	X	X	X	X	X	
S9	X	X	X	X	X	
S10	X		X	X	X	
S11	X		X	X	X	
S12	X		X	X	X	
S13	X		X	X	X	
S14	X		X	X	X	
S15	X		X	X	X	
S16	X		X	X	X	
S17	X	X	X			X
S18	X		X	X	X	
S19	X	X	X			X
S20	X	X	X			X
S21	X		X	X	X	
S22	X		X	X	X	
S23	X		X	X	X	
S24	X	X	X	X	X	
S25	X		X	X	X	
S26	X	X	X			X
S27	X		X	X	X	
S28	X		X	X	X	
S29	X		X	X	X	
S30	X		X	X	X	
S31	X		X	X	X	
S32	X	X	X			X
S33	X		X	X	X	
S34	X		X	X	X	
S35	X		X	X	X	
S36	X	X	X			X
S37	X		X	X	X	
S38	X		X	X	X	
S39	X		X	X	X	
S40	X		X	X	X	
S41	X		X	X	X	
S42		X	X			X
S43	X	X	X			X
S44	X	X	X			X
S45	X	X	X			X
S46	X					
S47	X		X		X	
S48	X	X	X			X
S49		X	X			X
S50	X		X	X	X	
S51	X		X	X	X	
S52	X		X	X	X	
S53	X		X	X	X	
S54	X		X		X	
S55	X		X	X	X	
S56	X		X	X	X	

Table 2.2: List of Publications

ID	Type of Literature	Found Database(s)	Title	Name	Year
S1	Grey Literature	Google	Implications of Technical Debt Uncertainty for Software Licensing Negotiations	Gat and Coté	2010
S2	Grey Literature	Google	What is Technical Debt in Agile?	Omeyer	2019
S3	Grey Literature	Google	Technical Debt - the silent villain of web development	Golofit	2020
S4	Grey Literature	Google	Managing Technical Debt	Johann and Wolff	2013
S5	Peer-reviewed	Google Scholar	Technical debt in the code	O'Neill	2013
S6	Grey Literature	Google	The Hidden Cost Of Technical Debt	Gautam	2020
S7	Peer-reviewed	IEEE, Citeseer	On the uncertainty of technical debt measurements	Izurieta et al.	2013
S8	Peer-reviewed	IEEE	Using real options to manage technical debt in requirements engineering	Abad and Ruhe	2015
S9	Peer-reviewed	ACM	A portfolio approach to technical debt management	Guo and Seaman	2011
S10	Peer-reviewed	ACM, Citeseer	Managing technical debt in software-reliant systems	Brown et al.	2010
S11	Peer-reviewed	Elsevier, ACM	Costs and obstacles encountered in technical debt management–A case study	Guo et al.	2016
S12	Peer-reviewed	IEEE, ACM	Tracking technical debt—An exploratory case study	Guo et al.	2011
S13	Peer-reviewed	IEEE, ACM	Practical considerations, challenges, and requirements of tool-support for managing technical debt	Falessi et al.	2013
S14	Peer-reviewed	Springer, ACM	Exploring the costs of technical debt management–a case study	Guo et al.	2016
S15	Peer-reviewed	IEEE, ACM	Architecture technical debt: Understanding causes and a qualitative model	Martini et al.	2014
S16	Peer-reviewed	Elsevier, ACM	An exploration of technical debt	Tom et al.	2013
S17	Peer-reviewed	IEEE, ACM	CloudMTD: Using real options to manage technical debt in cloud-based service selection	Alzaghoul and Bahsoon	2013
S18	Peer-reviewed	Google Scholar	A Cost-Benefit analysis model for technical debt management considering uncertainty and time	Fernández Sánchez et al.	2013
S19	Peer-reviewed	IEEE, ACM	Evaluating technical debt in cloud-based architectures using real options	Alzaghoul and Bahsoon	2014
S20	Peer-reviewed	ACM	To adapt or not to adapt? technical debt and learning driven self-adaptation for managing runtime performance	Chen et al.	2018
S21	Peer-reviewed	ACM, Elsevier	Investigating architectural technical debt accumulation and refactoring over time: A multiple-case study	Martini et al.	2015
S22	Peer-reviewed	IEEE	Security risk assessment and management as technical debt	Rindell and Holvitie	2019
S23	Peer-reviewed	IEEE, ACM	The danger of architectural technical debt: Contagious debt and vicious circles	Martini and Bosch	2015
S24	Peer-reviewed	IEEE, ACM	When-to-release decisions in consideration of technical debt	Ho and Ruhe	2014
S25	Peer-reviewed	Elsevier, ACM	Identifying self-admitted technical debt through code comment analysis with a contextualized vocabulary	de Freitas Farias et al.	2020
S26	Peer-reviewed	IEEE, ACM	Economics-driven approach for managing technical debt in cloud-based architectures	Alzaghoul and Bahsoon	2013
S27	Peer-reviewed	IEEE, ACM	Applicability of technical debt as a concept to understand obstacles for evolution of automated production systems	Vogel-Heuser and Rösch	2015
S28	Peer-reviewed	ACM, Springer	Measuring affective states from technical debt: A psychoempirical software engineering experiment	Olsson et al.	2021
S29	Peer-reviewed	Wiley Online Library	On the interest of architectural technical debt: Uncovering the contagious debt phenomenon	Martini and Bosch	2017

Table 2.2: List of Publications (Continued)

ID	Type of Literature	Found Database(s)	Title	Name	Year
S30	Peer-reviewed	IEEE, ACM	Towards better technical debt detection with nlp and machine learning methods	Rantala	2020
S31	Peer-reviewed	Elsevier	Building and evaluating a theory of architectural technical debt in software-intensive systems	Verdecchia et al.	2021
S32	Peer-reviewed	IEEE	Managing technical debt in database normalization	Albarak et al.	2020
S33	Peer-reviewed	IEEE, ACM	Technical debt triage in backlog management	Besker et al.	2019
S34	Peer-reviewed	Elsevier	Technical debt forecasting: an empirical study on open-source repositories	Tsoukalas et al.	2020
S35	Peer-reviewed	IEEE, ACM	Identifying scalability debt in open systems	Hanssen et al.	2019
S36	Peer-reviewed	IEEE	Identifying and estimating technical debt for service composition in SaaS cloud	Kumar et al.	2019
S37	Peer-reviewed	ACM	Supporting technical debt cataloging with TD-Tracker tool	Foganholi et al.	2015
S38	Peer-reviewed	Wiley Online Library, IEEE	Architectural technical debt in embedded systems	Martini and Bosch	2019
S39	Peer-reviewed	IEEE, ACM	Explicating, understanding, and managing technical debt from self-driving miniature car projects	Al Mamun et al.	2014
S40	Peer-reviewed	IEEE, ACM	Technical debt as an external software attribute	Lavazza et al.	2018
S41	Peer-reviewed	IEEE, ACM	From lasagna to spaghetti, a decision model to manage defect debt	Aldaeej and Seaman	2018
S42	Peer-reviewed	IEEE	Towards the definition of enterprise architecture debts	Hacks et al.	2019
S43	Peer-reviewed	ACM, Springer	Systematic elaboration of compliance requirements using compliance debt and portfolio theory	Ojameruaye and Bahsoon	2014
S44	Peer-reviewed	IEEE, ACM	Elasticity debt: a debt-aware approach to reason about elasticity decisions in the cloud	Mera-Gómez et al.	2016
S45	Peer-reviewed	IEEE	Database design debts through examining schema evolution	Al-Barak and Bahsoon	2016
S46	Peer-reviewed	Springer	Who pays for ethical debt in AI?	Petrozzino	2021
S47	Peer-reviewed	IEEE	Process debt: A first exploration	Martini et al.	2020
S48	Peer-reviewed	IEEE, ACM	Sustainability debt: A portfolio-based approach for evaluating sustainability requirements in architectures	Ojameruaye et al.	2016
S49	Peer-reviewed	Google Scholar	Prioritization of EA debts facilitating portfolio theory	Yeong et al.	2019
S50	Peer-reviewed	Google Scholar	Controlling Technical Debt Remediation in Outsourced Enterprise Systems Maintenance: An Empirical Analysis	Ramasubbu et al.	2021
S51	Peer-reviewed	IEEE	Self-Admitted Technical Debt in Ethereum Smart Contracts: A Large-Scale Exploratory Study	Ebrahimi et al.	2023
S52	Peer-reviewed	IEEE	Common Causes and Effects of Technical Debt in Costa Rica: InsighTD Survey Replication	Murillo et al.	2021
S53	Peer-reviewed	IEEE	Accumulation and Prioritization of Architectural Debt in Three Companies Migrating to Microservices	De Toledo et al.	2022
S54	Peer-reviewed	Google Scholar	Process Debt: Definition, Risks and Management	Martini et al.	2023
S55	Grey Literature	Google	Gradual Rewrite for Reducing Technical Debt	Ristimaki	2023
S56	Grey Literature	Google	How to Manage Technical Debt (and Why It Matters)	Stolyar	2021

Table 2.3: List of Primary Studies

ID	Type of Literature	Found Database(s)	Title	Name	Year
S7	Peer-reviewed	IEEE, Citeseer	On the uncertainty of technical debt measurements	Izurieta et al.	2013
S8	Peer-reviewed	IEEE	Using real options to manage technical debt in requirements engineering	Abad and Ruhe	2015
S9	Peer-reviewed	ACM	A portfolio approach to technical debt management	Guo and Seaman	2011
S24	Peer-reviewed	IEEE, ACM	When-to-release decisions in consideration of technical debt	Ho and Ruhe	2014

- **Applicability:** This factor considers '**A1** - the uncertainty-related investigations, covering the problem and/or the solution, and can be relevant or applied to different TD problems or TDM phases. A study in which '**A2** - the implementation, evaluation, the cases used are of wider applicability, and whether the evaluation is a step closer to its practical usage.' Though TDM has been widely discussed in the industrial context, our search has not found studies scoped for this context.

Table 2.1 compiles a comparison matrix of all studies using the above criteria. The list of all studies is shown in Table 2.2. In summary, only 4 out of 56 studies have adequately addressed C1 and C2 and satisfy R1, R2. Those four studies are shown in Table 2.3 and can be considered primary studies. We analyse all surveyed studies in the next section.

2.4.1 Comparison Result

Table 2.1 reports on how the identified studies are compared against completeness, relevance, applicability, and the extent to which they compare against these factors:

Difference in completeness: We found that the majority of studies tend to discuss causes of uncertainty in TD without providing a solution. This is expected as the majority of studies are experience reports, including case studies and interview

studies.

The first group of studies indicate the source of uncertainty in TDM. For example, [S11, S12, S14] explore the cost in TDM. The studies mentioned uncertainty in TD interest and repayment. [S12, S14] states that uncertainty in TD interest repayment can be considered a risk. [S11] mentioned that this type of uncertainty complicates software project management. This type of uncertainty can be considered erratic uncertainty, as repayment outcomes are context dependent and unpredictable. As [S11, S12, S14] are explorative case studies, they do not provide a solution to the above problem of TD uncertainty.

Uncertainty from challenges in estimating the impact of TD decisions is indicated in several studies [S6, S15, S21, S22, S28, S31, S38, S40, S41, S55]. [S15, S21], for example, points out a challenge to estimate ripple effects from architectural TD, partially due to the lack of awareness of the dependencies. This challenge can obscure architectural TD analysis and understanding of debt impact. This type of uncertainty is characterised by the unpredictability of architectural changes. Though these studies focus on architectural TD manifestation and accumulation over time, sources of uncertainties are not the primary focus of these studies. [S22] describes a challenge in TD interest calculation when we are uncertain about the effect of TD and security risk estimation if the context is security. [S27] adapt and apply the classification of TD to automated production systems (aPS) based on several industrial aPS projects. The study acknowledges a challenge in estimating TD impact caused by parallel software development.

Several studies acknowledge uncertainty when quantifying TD parameters or their impact. [S10], for example, is a foundational paper that raises awareness of TD and TDM in software-reliant systems. It hints at uncertainty when estimating TD impact. [S13] is an experience report on TD in the context of CMMI Maturity

Level 5. The study acknowledges uncertainties linked to the economic consequences of TD and calls for a more rigorous tool for managing TD uncertainty in this context. [S16] is an exploratory case study which utilises the study outcome and proposes a framework for understanding TD. The study notes difficulties in estimating project risks due to uncertainty around any changes to TD.

We have also found studies that discuss uncertainty in relation to unpredictable growth of interest. [S18], as an example, presents a cost-benefit analysis model to alleviate uncertainty in the growth of interest by using a binary tree. The study aims to elicit a possible range of TD interest growth within different timeframes, and the cause of uncertainty is left unmitigated. [S22] acknowledges inherent uncertainty in interest calculation and provides a risk-oriented approach to prioritise TD. However, the inherent uncertainty in interest calculation is unaddressed in the study. [S37] acknowledges uncertainty in relation to interest growth, but the study contribution (e.g. integration of TD tracker and extension tools) does not address the problem.

[S33] indicates difficulties in TD prioritisation due to the unpredictable future of the manifested components/software from the surveying results of 17 practitioners. [S34] presents a predictive TD analysis based on open-source repositories. The study utilises Machine learning methods for forecasting the uncertainty in TD evolution. However, the uncertainty of TD evolution and its parameters, which can be manifested in the data set, remain to be addressed. [S35] presents a concept of TD in scalability requirements and identifies uncertainty from the system scalability. The study demonstrates the presence of TD in the scalability requirements of a banking system. However, the raised problem of uncertainty is left unaddressed. [S46] is an opinion paper indicating the ethical uncertainty of practitioners for TD in the field of Artificial Intelligence, which is an open research topic. [S50] is an empirical analysis that indicates environmental uncertainty in the enterprise's systems can complicate TDM and curtail control balancing efforts.

[S30, S51, S53] present the cause of uncertainty when dealing with conflicting perspectives. Particularly, [S30] discovered uncertainty when using natural language processing (NLP) and machine learning to identify self-admit TD from code comments. However, the study is still a work in progress. [S51] is an exploratory study on self-admitted TD in more than 700k real-world Ethereum contacts. The study provides a taxonomy for self-admitted TD, where uncertainty in various design choices is documented. [S53] is an interview study of 47 practitioners on microservices architectural TD. The study identifies uncertainty in insufficient metadata where practitioners are unsure whether additional metadata can resolve the current issues.

The second group of studies point out uncertainty as a cause of TD. [S15, S21, S23, S29] conduct case studies in several large companies to explore the accumulation of architectural TD over time. All of those indicate uncertainty as a cause of architectural TD. A source of uncertainty is attributed to unclear specifications and use cases. This class of uncertainty can be essentially caused by information deficiency and can be mitigated with more information (e.g. clearer/revised use cases). The uncertainty that can cause TD attributed to unclear documents is briefly noted in [S25, S38, S39, S55, S56]. Notably, [S38] is a series of case studies for investigating architectural TD in embedded systems. The study points out that architectural TD can stem from the uncertainty of use cases at the beginning of the development. [S47, S54] presents a concept of Process Debt, a derivative of TD where a suboptimal activity or process that might have short-term benefits but yield a negative impact in the medium-long term. The papers point out that there can be uncertainty in Process Debt due to the lack of documentation and clarity of purposes. These are interview studies that provide a conceptual model of Process Debt. Thus, an approach or mitigation to the aforementioned cause of uncertainty is left to be researched.

We found two studies [S42, S49] that present a concept of TD in Enterprise

Architecture and discuss associated uncertainty. Enterprise Architecture Debt is a deviation of the current state of an enterprise from a hypothetical ideal state. A concept of Enterprise Architecture Debt can enable further research in alleviating a deviation from the current state of an enterprise to an ideal state. However, the cause of the deviation, including the source of uncertainty and the unavailability of resources, is not defined in the study.

There are publications that only mention the role of uncertainty without any elaboration on its causes or an explicit solution to address uncertainty. [S2] mentioned that uncertainty is one of the major factors in TDM. [S3] mentioned the impact of uncertainty in TD. These studies do not identify a cause of uncertainty or provide a solution. [S4, S5] indicates TD uncertainty without providing detail or a solution.

Differences in relevancy: In this subsection, we compare studies which adequately address C1, C2, R1 but do not meet R2. These studies indicate and address the source of uncertainty when managing TD in different system contexts.

[S1] is an opinion paper which discusses uncertainty due to the lack of TD information from software vendors in the context of software licensing negotiation. The study recommends requesting TD data for the application and TD mitigation plans. Uncertainty in this study relates to the lack of information on TD backlog.

[S17, S26] define a concept of TD for cloud-based service-oriented architectures. The study formulates TD in this environment as an option problem. [S19] utilises the Design Structure Matrix (DSM) in developing a method to address the problem. Uncertainty of the problem is related to conflicting perspectives on optimal selection or substitution decisions in the cloud-based service-oriented architecture.

[S20] presents a concept of TD in the context of Self-Adaptive Systems, where TD is related to the runtime performance. Based on this concept, the study presents a framework to determine when and whether to adapt the system in a dynamic manner and comprehensively evaluate the proposed framework in comparison to four debt-oblivious triggering approaches. Uncertainty in the study originates in different opinions on adaptation decisions in Self-Adaptive Systems.

[S36] presents a novel approach for identifying TD in service composition under Software-as-a-Service (SaaS) cloud. The paper proposes a TD model in the context of service composition and utilises a time-series forecasting method to estimate future TD. The source of uncertainty in this study originates from conflicting perspectives in service composition decisions, which are driven by time-sensitive workloads.

[S32] presents a concept of database normalisation debt, which can occur in tables below the fourth normalisation form. The study proposes a framework leveraging a novel use of the Portfolio Theory and the TOSIS method (Technique for Order of Preference by Similarity to Ideal Solution). The uncertainty in this study stems from different database normalisation decisions.

[S43] defines a concept of TD for managing compliance requirements. The study proposes a goal-oriented method and obstacle handling with portfolio-based thinking to manage and refine compliance goals. Uncertainty in the problem of the study relates to different opinions in mitigating compliance requirements.

[S44] presents a concept of elasticity debt, which originates from sub-optimal self-adaptive and elasticity management decisions. The paper demonstrates a proof of concept with CloudSim to show its effectiveness in debt-aware elasticity decision-making. Uncertainty from the problem in this study relates to conflicting perspec-

tives between short-term and long-term gain in elastic adaptation decisions.

[S45] defines database design debt, which originates from violations of key design database principles, anti-patterns, etc. The study presents a taxonomy to help in classifying various types of database design debt and illustrate its use in a MediaWiki case study. Uncertainty in database design debt problem is related to conflicting views in treating non-optimal database design decisions.

[S48] illustrates a concept of sustainability debt, which is the gap between the current state of the available resources and the environment where the sustainability goals are ideally realised. The study proposes an architectural evaluation method, extending the Cost Benefit Analysis Method and the Portfolio Theory to quantify the sustainability debt of architectural design decisions. The source of uncertainty in this study is related to conflicting views on decisions in realising sustainability goals.

Differences in applicability: We compare four studies which satisfy C1, C2, R1, R2 and A1 in this subsection. These studies indicate and tackle the source of uncertainty in TDM. However, they tend to lack implementation and/or evaluation.

[S7] is an exploratory study that seeks answers to uncertainty in using TD measurement in metric-based tools. The study presents a rigorous technique inspired by physical sciences to represent the above uncertainty. Uncertainty in this study stems from the lack of information on the exact TD measure process and methods independent of the problem context. The study provides no evaluation of the presented contributions. As the uncertainty of the problem stems from metric-based tools, the study applies only to TD measurable by tools.

[S8] proposes to apply real options thinking to establish a quantitative method for

managing requirements debts. Uncertainty in this study relates to the unpredictability of environmental changes, the problem context (e.g. requirement marketability) and the cost of TD. The study illustrates the applicability of the contributions with a hypothetical project and plans for several industrial experiments as the further research agenda. The study is highly dependent on historical data in calculating the volatility factor, which renders a project with a small volume or no historical data unsuitable for this work.

[S9] proposes a portfolio approach to determine optimal repayment decisions. Uncertainty in this study is related to the unpredictability of the growth of TD interest. The study provides no evaluation and outlines an evaluation plan. The study utilises a series of assumptions in the financial domain, which require further validation. Additionally, the study acknowledges the limitation of the reliance on human experience in the estimation of the return on TD, which can cause another type of uncertainty due to the unpredictability of TD outcomes and its ripple effects.

[S24] is a position paper proposing a TD formation in the context of software releases. The study proposes an estimation method, leveraging the Fuzzy set theory to estimate TD. Uncertainty in the problem of this study is related to TD principal and the lack of its information. The study provides a proof-of-concept evaluation, where the method is embedded as a plugin for a release planning tool, which is evaluated on a real-world project. At the time of this writing, the repository for project data and evaluation analysis is inaccessible. The study specifically aims for code-based TD and does not consider other TD attributes other than TD principal.

2.5 Results

2.5.1 Research Landscape (RQ 1)

RQ1: What is the current research landscape in relation to uncertainty in TD?

As it is shown in Figure 2.2, the current research landscape on uncertainty in TD encompasses solution proposals, philosophical papers, opinion pieces, and experience reports, where the most common research type is experience reports, followed by solution proposals. We have found zero studies on uncertainty in TD that can be considered evaluation research or validation research.

As the majority of studies are evaluated on a small set of examples or have no evaluation. Up to our knowledge, existing studies on TD and uncertainty are neither conducted nor evaluated in a real-world industrial setup. Henceforth, we conclude that the current research on uncertainty in TD is in its early phase and tend to be driven by academic efforts.

2.5.2 Manifestation in TDM activities (RQ 2 & RQ 3)

RQ2: Which TDM activities can suffer from uncertainty?

According to our findings (shown in Figure 2.2), the majority of studies are related to uncertainty in TD measurement, followed by TD representation. TD identification and TD prioritisation are the third and the fourth most common TDM activities that can suffer from uncertainty.

Based on our search results, we conclude that most TDM activities can suffer

from uncertainty, where the most documented TDM activity is TD measurement.

RQ3: In which TDM activities, uncertainty is under-studied?

We have found no study which demonstrates the practicality of its contribution in a large-scale or industrial setting. Studies in TD representation, for example, are evaluated on a set of examples. Worse, contributions to the rest of TDM activities have no evaluation.In addition, it is inconclusive whether uncertainty exists in TD monitoring or TD prevention as there is no study mentioning uncertainty in the two above TDM activities.

Henceforth, we conclude that studies on uncertainty in most TDM activities are under-researched.

2.5.3 Type of Uncertainty (RQ 4)

RQ4: What are the causes of uncertainty in TDM activities?

We found that the majority of uncertainty documented is caused by the unpredictability of the problem domain. Examples of unpredictability are ripple effects, TD impact, the complexity of the problem domain, etc. Uncertainty from the unpredictability can manifest in most TDM activities, ranging from TD identification to TD repayment.

The second most common type of uncertainty is uncertainty due to information deficiency. It is documented that information deficiency can manifest TD identification, TD measurement and TD representation. Some examples of information deficiency are unclear/incomplete use cases, the lack of relevant information to make a judgement/decision, the lack of a unified standard, etc.

The least common type of uncertainty is uncertainty due to conflicting perspectives, which we documented that only manifested in TD identification and TD representation. Examples of conflicting perspectives are whether to identify TD based on the given code comment or different dimensions on how to represent TD in a new developing environment, emerging technologies, etc.

Apart from this, we found four studies that the behaviour of uncertainty are undefined/unmentioned. It is inconclusive whether uncertainty in those four studies can be classified into one of the above three types of uncertainty.

2.6 Reflection on Additional Causes of Uncertainty

As research on uncertainty in TD is in its early phase, there is only one study which contributes to a tool that considers uncertainty in TD [S44]. We observe that this limitation, combined with the following challenges in TD tools, can contribute to uncertainty in TD:

- As observed by [13, 69], there is a lack of commercial tools and research prototypes for some specific TD types, such as test debt. The lack of a systematic approach and tooling to address specific types of debt can prompt developers to take ad-hoc TDM, relying on the developers' assumptions, expertise, knowledge and competency and introducing uncertainty in the process.
- As there is no consensus on the important TD factors and their measurement approaches [69], practitioners may face the dilemma of defining identification criteria for TD and selecting suitable approaches and metrics for analysing and measuring the debts as one tool fits all situations can be impractical. This is because TD concerns, approaches and metrics to management can be unique

to some application domains and/or TD types. Consider, for example, the case of testing: Depending on the type of testing (e.g. coverage, code-level, architecture-level etc.), the effectiveness of TD identification and management can essentially vary with the adopted techniques and uncertainties about its fit for purpose/suitability.

- As observed in [13], TD toolings can vary in their effectiveness, coverage, and analysis of debts. Practitioners may have to use several code analysis tools to widen the coverage and inclusivity of detection criteria. Analysts can face a dilemma when deciding on the set of tools to be used and how they can effectively complement each other, and a lack of knowledge about these tools can be a source of uncertainty that can miss hidden debts.

- The lack of practitioners' knowledge and experience in these tools may lead to inappropriate tool selection that is not fit for purpose, leading to project waste and uncertainty in TD decisions taking the form of misidentification of the debts; over or under-estimation of the debt; bias of the analysis to the type(s) of debt that is supported by the chosen tool's, while ignoring other hidden and critical debts.

2.7 Discussions

The fundamental of TD under uncertainty, as a subject of research, is still in its infancy and is generally underdeveloped. Existing studies often use simplified assumptions about the underlying uncertainty, which can be overly specific to certain environments and thus hamper the proposed approaches for meaningful and practical application. [S7], for example, has investigated uncertainty in TD measurement due to the imprecise TD parameters estimation. Another example is the work of [S9]

on a portfolio approach for tackling the uncertainty of interest payment. However, this approach takes a simplistic assumption in mitigating uncertainties in interest estimation by following a normal distribution, which might not be true in practice. The application of real options by [S8] is another example. The study discusses uncertainties in relation to the option to delay requirement debt repayment; the definition of uncertainty is definitive to a specific TD type and behaviour, limiting the broader application of this approach.

Researchers and practitioners need to work together to investigate how the impact of uncertainty can propagate through TDM phases and can influence several decisions across and within phases. Identifying debt dependencies and their innate and external sources of uncertainties is another area that is worthy of research; it can fundamentally influence debt admissions, debt identification, areas within the system that require careful attention to avoid interest accumulation and considerations for debt payback.

The situation looks even more glimmering for practitioners as the mainstream tools and software for TD tend to ignore and do not currently support the representation of uncertainty in TDM. Existing tools are short in providing modelling support, visualisation, and mechanisms for identifying sources of uncertainty and their mitigation when managing TD. Such limitations leave the treatment and mitigation of uncertainties in TDM opportunistic, lack systematic guidance and are biased to the analyst's skills and their personal desire. The absence of tools may drive practitioners to initiate TDM activities and manually tackle underlying uncertainty, including analysing sources, dynamics and probing for modelling the sensitivity of TD parameters to uncertainty. Consequently, practitioners may ignore the importance of uncertainty in TD or tackle it externally to the tool with techniques that might be error-prone, not ideal or fit for purpose. Practitioners might wish to employ additional techniques (e.g. Monte Carlo sampling, Delphi method) for calibrating

uncertainty from TD measurement, which can be labour-intensive, and the choice of the techniques can be biased.

2.8 Moving Forward and Addressing the Gaps in Managing TD under Uncertainty

Researchers and practitioners in TD should collectively work on setting an agenda and raising awareness of uncertainties in TDM activities and thus develop methods, techniques and tools that remediate the aforementioned issues. Research needs to develop the fundamentals of systematically managing TD under uncertainties that can be linked to different TDM activities and identify mitigation strategies. In addition, the definition of universally accepted TD uncertainty definition and properties should be discussed and should be systematically addressed to alleviate the current state of fragmented research and knowledge gaps on the topic. Research can seek inspiration from decision-making under uncertainty in economics, policy-making, health care and statistics. Notable examples that can inform research are Knightian uncertainty [60], the economics of information [109], the source of uncertainty [56] and the classification of uncertainty that is widely used in the medical application [49].

Practitioners need to embrace uncertainty in TDM and be sceptical about obtained information and/or evidence of TD throughout TDM activities. Although it can be exhaustive by employing this mindset, we can probe to identify uncertainty sources and systematically manage them.

Systematically managing uncertainty in TD requires several steps, covering elicitation of the source, monitoring, tracking, etc. The following are recommended mit-

igation activities for practitioners to mitigate uncertainty in TD along with TDM:

- Practitioners may employ scenario-based analysis and multi-perspective evaluation to improve the level of confidence about the TD query and uncertainties revolving around the query.
- Practitioners may document them similar to risk items, including their severity, origin, mitigation route, and periodically update them throughout the life cycle. The objective is to make the mitigation of uncertainties less labour-intensive, explainable and repeatable.
- In terms of TD decision-making, when it is possible, decisions must be collectively made by different stakeholders. By doing so, decisions can be verified, and possible sources of uncertainty can be collectively discussed. Additionally, decisions can be then made based on diverse evidence, neutralising bias, inexperienced opinions and/or ill-suited assumptions.
- Communication/estimation techniques, such as Planning Poker, Wideband Delphi, and Delphi method, might be employed during these activities to elicit underlying assumptions, rationale and different aspects from different stakeholders.
- As the accessibility of TD knowledge and data is essential to mitigate uncertainties, cross companies' datasets and data-sharing infrastructure can unlock the potential of more rigorous methods to alleviate uncertainty in TD via data-driven analytics. One of the potential routes is to leverage data-mining technologies and the applications of machine learning to mitigate uncertainties of human-centred predictions. Consequently, practitioners can employ knowledge and gather data to support TD decision-making under uncertainty and verify the rationale behind the produced TD decisions. Additionally, advancements in tools for modelling, visualising and mitigating uncertainties are necessary

to accelerate awareness of the topic and adoption. Continuous integration and continuous deployment methods and modern computing paradigms that are characterised by complexity, heterogeneity, and decentralisation, such as IoT, cloud, microservices architecture, etc., can particularly benefit from the vision.

2.9 Threat to Validity

We acknowledge that the quality of this mapping study may be influenced by the coverage of the study search, its study selection and the quality assessment of the extracted sources, as discussed below:

2.9.1 Inadequate Coverage of Study Search

There might be a relevant study that is not included in this mapping study, potentially deterring the overall coverage of the mapping study. Although we attempted to mitigate this by employing our additional search on Google and Google Scholar, there still might be a study which is uncovered by these databases. This may be due to the lack of the "snowballing" technique [130]. We intentionally omit this technique as it is inapplicable to grey literature, such as experience reports in blogs, technical reports, social platforms, etc.

2.9.2 Bias on Study Selection

Although we have provided clear study selection criteria, this mapping study can produce different search results due to different interpretations of inclusion criteria.

Specifically, different researchers might have different opinions on a study related to TD or related to uncertainty, which results in different studies being selected. We attempt to mitigate this by grounding precise selection criteria in different phases of selection, creating hard boundaries for different interpretations.

2.9.3 The Lack of Quality Assessment

To provide the current research landscape on uncertainty in TD, we select all types of studies, whether it is non-peer-reviewed or peer-reviewed, in our search scope. Without the quality assessment of selected studies, this decision can demote the quality of the mapping study. As research on uncertainty in TD is in its early stage, we believe it is imperative to consider all studies, regardless of their quality, as some of those can be the foundation for future research. Hence, we agree with Kitchenham et al. that the quality assessment of the primary study is not a necessary task for the mapping study [57].

2.9.4 Bias in Data Extraction

Researchers can have different judgements or opinions about classification schemes during data extraction. This issue might stem from their own gains, bias, experience, speciality, experience, etc. To mitigate this, we build our classification schemes on existing classifications in the literature, which contain a precise definition for each classification. In addition, we present the selected studies matrix with classification rationale (where it is applicable), enabling future researchers to trace back the classification schemes and their rationale in this mapping study.

Chapter 3

Goal-Obstacles-based Approach for Managing Information Uncertainty

3.1 Introduction

Information is an important input for decision-making. Information plays a major role in reducing the uncertainty of decisions, where rational decision-makers systematically utilise information in a structured strategic decision-making process [29]. With complete and exact information, the expected utility, its objective and a known probability of outcomes can be characterised, supporting decision-makers to find an optimal and satisfying solution [128].

The desire to seek complete information when making decisions about alternatives can be overly-restrictive, if not impractical; consequently, using theories like utility theory in decision-making [128] can be less revealing and ineffective in the presence of imperfect information. Imperfect information or information deficiency can result in information-based uncertainty [59]. We refer to this phenomenon

as Information Uncertainty. Information Uncertainty can be reduced by acquiring/investing in additional relevant information [59]. In a given problem, there can be more than one alternative affected by Information Uncertainty. Thus, decision-makers might face difficulties in prioritising Information Uncertainty in which alternatives should be reduced.

Technical Debt (TD) is a technical compromise from trade-offs of short-term benefits with the long-term well-being of the software system [22]. Since the admission of technical debt can be intentional or unintentional [82], it should be managed to keep its impact under control [76]. Technical debt management (TDM) activities are TD identification, TD measurement, TD prioritisation, TD prevention, TD monitoring, TD repayment, TD representation and TD communication [75]. Information Uncertainty can be present when managing technical debt and in taking decisions that relate to one or more of the above TDM activities. For TD identification and measurement, as an example, the effectiveness of these two activities is subject to approach selection and configuration by practitioners. However, the completeness of TD identification and measurement information is often subject to practitioners' satisfaction. Practitioners can have different perspectives on the selection, configuration, and utilisation of a suitable TD identification and measurement approach, those which are based on one's background, preference, experience, speciality, etc. Without similar competency, it can be difficult to determine whether existing TD identification and measurement information is sufficient. This issue can introduce Information Uncertainty in TD identification, and without careful consideration, its outcome can consequently lead to misidentified and hidden TD in the TD backlog. Decisions with Information Uncertainty can also lead to the under or over-estimation of TD parameters in TD measurement. As the information deficiency can apply to any TD, Information Uncertainty can affect several TD in the backlog. Without any prioritisation technique, practitioners are left with an Infor-

mation Uncertainty-ridden TD backlog, which can result in ineffective TD ranking and ineffective repayment strategies.

Despite the problematic consequences of Information Uncertainty in TDM and its activities during decision-making, TD research has not looked into systematic approaches and mitigation strategies for tackling Information Uncertainty, which we address in this chapter. This contribution focuses on Information Uncertainty in TD identification and TD measurement; nevertheless, the treatment can still be applied to and benefit other TDM activities.

We present a novel probabilistic goal modelling approach leveraging sound and well-established goal-obstacles and scenario analysis in requirements engineering [26, 124, 111], and value of information (VOI) analysis in health economics [10, 30] to mitigate subjectivity and incompleteness of information, as sources of uncertainty, when identifying and measuring TD. The modelling can be utilised to elicit more information on the system context to better contextualise TD identification and estimation. Our model can also help in supplementing and verifying the ability of existing TD identification and estimation models to encapsulate TD impact on the said context. To prioritise Information Uncertainty, the contribution makes a novel use of the VOI analysis to determine the worthiness of reducing Information Uncertainty in TD, valued by possible gains, when additional information is introduced. The VOI analysis and ranking can be utilised to inform an information investment strategy, looking to invest those with the greatest net benefit [134]. The approach builds on cost-effectiveness and decision analysis widely used in health economics and can be traced to a study by a Nobel Prize winner Kenneth Arrow [10]. In health economics, the expected value of perfect information can be employed to estimate the VOI of a decision on a promising treatment and/or to adopt new technology for the said treatment.

This chapter's main contribution is a systematic approach for reducing Information Uncertainty when managing TD; the focus is on TD identification and TD measurement. Below, we list the specific contributions:

- We illustrate the presence and ill-consequences of Information Uncertainty in TD identification and measurement sub-activities.
- We develop a novel probabilistic TD goal-obstacles modelling approach, complemented with scenario-based analysis, to assist in contextualising information and reducing its uncertainty when identifying and measuring TD.
- We present a new concept of decision analysis in TD by illustrating similarities of decision-making on TDM and health economics. We describe a mapping that draws on the combination of similarities with decision analysis in health economics to enable TD decision-makers to better conceptualise and formalise the potential benefits of TD for an information investment strategy. This concept and corresponding mapping are later utilised for the application of the VOI when handling the problem of Information Uncertainty in TD and investing in information investment strategies to mitigate uncertainty.

We demonstrate our approach using an open-source e-commerce microservices architecture of an industrial scale (provided by Microsoft). We evaluate the effectiveness of the approach in mitigating Information Uncertainty when managing TD. We compare our approach to two baseline approaches, namely cost-saving and interest-based. We report on entropy and stability analysis. The results show that our approach can produce a more robust and stable TD decision under situations exhibiting Information Uncertainty.

The chapter is organised as follows: Section 3.2 presents background on Information Uncertainty, health economics and its mapping to TDM and decision-making.

Section 3.4 describes the methodology. Section 3.5 reports on the evaluation. Section 3.6 reports on the discussions and threats to validity. Closely related works are discussed in section 3.3, and this chapter concludes in Section 3.7.

3.2 Background

Research on uncertainty in TDM decision-making is limited. TD studies discussing uncertainty tend to focus on specific TD measurement and estimation problems (e.g., measurement imprecision [50], the net benefit of TD [40]) or only available for a specific type of TD (e.g. requirement debt [1]). There is a general absence of a comprehensive classification of uncertainty in TDM decision-making. We look at [56] for the categorisation of uncertainty in computation modelling as the foundation for potential types of uncertainty in TDM decision-making.

1. Parameter Uncertainty is the situation where input parameters of the model, which characterise features of an application context, are unknown.

2. Model Inadequacy is the situation where the outcomes of the selected model differ from the true values of the expected outcomes.

3. Residual Variability is the situation where the selected model yields variations of the outcomes from the same inputs. This source of uncertainty can result from an unpredictable process or model conditions that are overly generalised.

4. Parametric Variability is the situation where the inputs of the model are varied and require more detail in order to be precisely utilised.

5. Observation Error is the situation where there is the possibility of an error during measurement observation.

6. Information Uncertainty: This is known as "Code uncertainty" in the study; this is the situation where more information is required in order to receive a clearer view of input-output pairs. Although, in theory, those might not be considered as unknown since they can be eliminated when information is acquired, such activity can be impossible without a trade-off between real-world constraints (e.g. cost, effort, time, etc.) and certainty.

Uncertainties in the above categorisation are fundamentally manifested in the computerised model components, and thus they can reside in TDM activities that utilise TD representation. We specifically investigate TD identification and TD measurement. As decision-making in these two TDM activities is informed by TD representation and the estimation of TD attributes, these two activities are vulnerable to the above uncertainties. Particularly, uncertainties can propagate in TD models or TD parameters, which are utilised throughout the TDM lifecycle.

The above uncertainties can simultaneously exist as the combination of TD approaches and/or methods can be utilised. During TD identification, practitioners may, for example, select more than one TD model/approach. Each model can then produce results on aspects of TD, diversifying the analysis of TD (e.g. possible code smells or architectural degradation [34]). Different results may be produced when TD identification is used in conjunction with TD measurement methods (e.g. code-based TD model from a Commercial-Of-The-Shelf (COTS) tool [107], or models focusing on specific software quality and violations to standards [13]).

During TD measurement, practitioners can employ different TD measurement methods, focusing on specific quality attributes of interest, types of TD, etc. Approaches may, for example, be centred on consulting experts, motivated by cost-benefit analysis[104], and/or informed by specific standards and scale (e.g., SQALE [71]). In addition, the effectiveness of TD identification and measurement is influ-

enced by practitioners' configurations (e.g. Identification threshold/measurement profile in a COTS tool [107]).

Table 3.1: Generalised source of uncertainty in TD identification and measurement

Process	Possible source of uncertainty	Technical Debt Identification	Technical Debt Measurement
Model selection	2) Model Inadequacy 3) Residual Variability 6) Information Uncertainty	- Is there a suitable identification approach for the type of debt? - Is the identification approach suitable for the developing environment? - Is the identification approach suitable with the TD type? - Is the identification approach sound and reasonable? - Are more than one identification approaches required for the developing environment?	- Is there a suitable measurement approach for the type of identified TD? - Is the measurement approach compatible with the identified approach? - Is the measurement approach appropriate for the identified debt? - Is the measurement approach sound and reasonable? - Is more than one measurement approach required for the identified debt?
Model Configuration	1) Parameter Uncertainty 4) Parametric Variability 5) Observation Error 6) Information Uncertainty	- Are the identification criteria properly adjusted? - Is the identified debt verifiable and falsifiable? - Does the identified debt differ from a bug or defect? - In case more than one identification approach is required, how to prioritise those approaches?	- Are the parameters of the selected measurement approaches sound and reasonable? - Are the measured values verifiable and falsifiable? - Does the measurement approach encapsulate the debt severity/characteristics? - In case more than one measurement approach is required, how to balance those measurement approaches?

Table 3.1 lists possible classes of uncertainty based on [56], where TD identification and measurement may essentially suffer from one or more of the listed classes. We provide a non-exhaustive checklist of the possible source of uncertainty in TDM activities. We briefly summarise them in the following sub-activities:

- Selection of the appropriate TD model: Since there is no consensus on which TD model and TD measurement to use [13][50], practitioners may have to pick suitable TD models and measurement at their own discretion. When more than one model or measurement is utilised, practitioners may need to balance the importance of different models or measurements on their own. TD modelling and measurements can be inadequate, limited or biased, resulting

in incorrect TD representation and sub-optimal decisions. Additionally, practitioners might utilise an inaccurate, partial TD model or partial TD measurement method, contributing to an unreliable TD model or measurement (i.e. Residual Variability).

- Configuration of the selected TD model: Practitioners may encounter Parametric Uncertainty when they do not know a suitable value and attempt to use a default profile or default parameters. This situation can happen when practitioners have no prior experience with the system, no experience with the selected TD model or do not have historical data. A default profile can contribute to Parametric Variability, where a TD model or measurement requires more detail to be reliably utilised. Worse, a default profile can cause Observation Error, where an error in observation, embedded in the default profile, is utilised in a TD model or measurement.

3.2.1 Uncertainty in TD Identification and Measurement

External or domain experts have often utilised information in the search for TD identification approaches and TD measurement methods. This information is used to justify, determine or reason about possible TD, suitable formulas, metrics and/or procedures for quantifying TD parameters. However, these sources of information are often based on common knowledge, best practices, or published studies. They can be subjective, biased or imperfect, as there may exist hidden information outside of a given scope, or in the application comes with assumptions or threats to validity. These issues challenge practitioners in determining "what" and "how many" TD identification approaches and TD measurement methods to utilise for encapsulating the presence and the severity of TD. Practitioners may attempt to mitigate this by diversifying TD identification approaches and measurement methods, including

selecting and configuring different TD identification approaches and measurement methods to cover different TD types and cater for perspectives, etc. [62] propose the participation of different stakeholders to identify and measure TD in a more comprehensive manner. However, the effectiveness of the selection and configuration of TD identification approaches and measurement methods can be difficult to verify or falsify. Practitioners might have to obtain relevant information, experience, speciality, etc., to the problem context in order to verify or falsify them. These can be infeasible to obtain or substitute. Difficulties in falsifying the proposed selection and configuration of identification approaches or measurement methods may influence practitioners to utilise them without adequate inspection, opening an opportunity for Information Uncertainty to be manifested in the TD model or its parameters.

Information Uncertainty can result in Parameter Uncertainty (i.e. the lack of information on suitable input parameters), Model Inadequacy (i.e. the lack of information on the appropriate TD model), and Observation Error (the lack of information for the correct measurement methods). It can result in ill-selection or misconfigured TD identification approaches that are not fit for purpose, ignoring hidden TD. Information Uncertainty can also result in partial or incorrect TD measurements, resulting in TD misestimation.

Although Information Uncertainty can be a root cause of several TD misidentification and misestimation problems and lead to several classes of uncertainty in TDM decision-making, its systematic mitigation approach in TDM decision-making is unexplored. In the next section, we investigate the field of health economics, where several techniques exist to mitigate information-based uncertainty. Though a set of Software Engineering studies [108, 70, 24] have explored the use of health economics techniques to mitigate software decision-making problems under uncertainty, health economics techniques have not been considered in TDM decision-making.

3.2.2 Health Economics

Health economics is the application of economic theory to health and health care. Whereas "health" is a state of complete physical, mental and social well-being of an individual, "health care" is the treatment, diagnosis, and prevention of disease as delivered by medical and other practitioners [16]. The production of health and health care are considered as "economic goods" [85], as good health care can contribute to better well-being.

In a study that consider as the birth of health economics[100], A Noble Prize winner Kenneth J. Arrow demonstrates the implications of the uncertainty of the medical-care industry towards its market equilibria by showing that uncertainty in "the incidence of disease and the efficacy of treatments" can contribute to an inefficient allocation of resources and may eventually lead to market failure [100, 10], health epidemic and death. However, uncertainty and risks underlying the decision process on treatment selection are inevitable (due to investments' irreversibility, etc.). Thus, information or knowledge becomes a commodity, which physicians and most professionals invest in the VOI for dealing with uncertainty [10].

Health economic evaluations should be subject to some form of sensitivity analysis, which can be utilised to "quantify and qualify the uncertainty underlying the decision process."[16]. As such, sensitivity analyses can determine whether it is worth gathering more data before making the final decision. One promising probability sensitivity analysis is a decision-theoretic approach based on the VOI [45]. The VOI is the value of an option before knowing its real value. This approach represents the difference between the expected utility of the optimal action after and the expected utility of the optimal action before additional information is collected. As such, the best information strategy is the one that yields the greatest net benefit [134]. This concept is called the expected value of total perfect information or the

expected value of total perfect information (EVTPI). Additionally, Howard introduces the concept of the expected value of partial perfect information (EVPPI). In general, EVPPI illustrates the difference between the optimal action after the information of an underlying parameter is obtained and the expected utility of the optimal action prior to information [45].

3.2.3 Mapping of Treatment Selection to Technical Debt Management Decision-Making

We have reviewed a study on health economics that relates to real-world health technology adoption decision-making in the UK health care system [30], and we have identified similar characteristics to that of TDM decision-making, making health economics a sensible theory to ground our techniques on through analogy and mapping:

• **Evidence-based decision-making**: Evidence-based decision-making was the main focus of the study. Specifically, a problem in Health Technology Assessment is mitigated by utilising existing clinical evidence to approximate the negative implication of incorrect health technology adoption [30]. Similar evidence-based practice is adopted in several areas of TDM. Notable examples are the utilisation of software code and documentation to identify and measure potential TD [13] and the use of the COTS TD tool to estimate TD effort and severity [107].

• **Role of imperfect information**: The study acknowledges decision-making under uncertainty caused by imperfect information or information deficiency. The study explicitly considers uncertainty surrounding the implication of a new technology to adopt. Information on new technology can be considered imperfect information as it is impossible to precisely indicate its implication without actually adopting it.

The study utilises additional information to reduce uncertainty related to imperfect information. Notably, the process of clinical evidence acquisition is required, and a need to access clinical experts is recommended to tackle uncertainty in a problem of Health Technology Assessment [30]. For TD, imperfect information is implicitly utilised in practice throughout TD decision-making. Notably, TD decision-makers make TD decisions based on an approximation of TD properties (e.g. TD interest, TD effort, etc.) without knowing their exact values. Another example is the decision-makers utilise available information during TD decision-making without the need to acknowledge the completeness of existing information. The concept of reducing Information Uncertainty from imperfect information also is used in TDM. When the practitioners utilise COTS tools or ask TD experts for their opinion, for example, the practitioners acquire more information on TD, reducing Information Uncertainty from imperfect information on TD and/or its attributes.

- **Need for research prioritisation**: The study notices the need for research prioritisation to reduce uncertainty from imperfect information. More precisely, the study utilises a framework that values additional information that may be generated by further research. The estimation of the value of additional information in different areas is consistent with objectives and resource constraints of a health care system [30]. Similarly, TDM decision-making is resource-constrained and context-dependent. The importance of each TD is valued based on its manifestation and properties (e.g. its location, impact, profitability, reusability), TD ranking, choice of the TD prioritisation approach, and resources/time constraints for mitigating the TD. Notable TD prioritisation approaches, for example, include SIG's quality model-[88], cost-benefit-[104], and SQALE-based TD prioritisation [71].

3.3 Related Work

Although TD decision-making and management are highly prone to Information Uncertainty, research into this area has been very limited and not explicitly addressed. One of the very first studies on uncertainty in TD is [50], where the authors propose to represent uncertainty from TD measurement based on methods in the physical sciences. Since their work's focus has been on TD measurement, this study refers to the source of uncertainty as systematic errors only from the poor calibration of tools, not variability of a parameter or insufficient information. This also applies to several studies into TD, which are grounded in concepts and theories from economics. For example, the use of portfolio theory in TD [40]. In the context of TD, this study treats TD as assets, and uncertainty is defined as risk, which the principle of diversification can reduce. However, uncertainty is limited and tied to the net benefit of TDM. [1] uses real options theory, which inherently considers uncertainty revolving around decisions in requirement debts. In the study context, uncertainty can relate to the option of delay in paying back the debt. However, the VOI and uncertainty that relate to information have not been considered, which we address in this chapter. Our cost-saving-based approach is based on [101], one of the few prioritisation approaches in [5] that is compatible with our evaluation. This study values the identified debts by considering the worthiness of mitigating the debts from their potential cost-saving. The study can be used to reduce uncertainty by introducing additional information in terms of the evolution steps of the system, its probability, and the cost-saving of each debt in each evolution step. By doing so, it is possible to prioritise the debts to mitigate that even the debt's components (e.g. principal, interest amount, etc.) are suffering from Information Uncertainty. However, the absence of the classical TD components (i.e. interest rate, interest amount) during decision-making by itself renders decisions from this study lacking the consideration of the debts' consequences. Our experiment case utilises the prin-

cipal estimation of TD from SonarQube, which is studied individually in [54] and along with other analytic tools in [13]. Letier et al. [70], and [24] utilise the VOI analysis for handling uncertainty in architectural decisions based on its direct monetary benefits. The main difference between this and the above studies is that our analysis is based on the consequences of decisions translated into a monetary value rather than the direct monetary benefit of decisions.

3.4 Methodology

We present three steps to mitigate Information Uncertainty in TDM decision-making, whereas each step is inspired by different findings in Section 3.2.3.

The overview and steps of our systematic approach for alleviating Information Uncertainty in TD are as follows:

1) Practitioners identify potential TD as obstacles in corresponding system goals. The premise of this step is to reduce Information Uncertainty in TD by adding evidence on the cruciality of TD to the system. We extend the relationship of system goals and obstacles [124] for systematically eliciting information for managing TD from the system context. Since the severity of TD depends on the system context and can obstruct system goals if they are mistreated, we posit that TD is a special type of obstacle; it is a shortcut, resulting in a short-term gain in exchange for an eventual non-satisfiable of a system goal. Not all obstacles are necessarily rooted in taking a debt. Therefore, it is possible to systematically locate TD by firstly identifying system goals and possible obstacles.

2) Practitioners utilise a probabilistic model of TD (Section 3.4.2) complemented with a scenario-based analysis (Section 3.4.3). The aim of this step is to reduce In-

formation Uncertainty by introducing information related to TD and its parameters. The proposed model builds on the probabilistic framework for goals and obstacles analysis [26] but scoped to the new context of application - TD identification and analysis. The objective is to facilitate obstacle analysis that can be rooted in debts, each with different severity and priority. The presented model considers parents' system goals and the probability of debts obstructing their corresponding goals. We use a scenario-based framework, which was initially proposed to elicit and verify system goals and their refinements in requirements engineering [111]; we use it to systematically elicit, falsify and verify TD information using context information along with the goals and their obstacles analysis. The objective is to reduce the subjectivity of supporting information for TD measurement and to ensure better adequacy/completeness. The ranking from the model is used to verify whether the existing TD model and ranking encapsulate information from the system context.

3) To reduce Information Uncertainty in the TD backlog, practitioners utilise the VOI analysis presented in this chapter to establish an information investment strategy. The premise of this step is to manage Information Uncertainty in TD for future mitigation. Information Uncertainty can be reduced from new information inferred from the system context; however, existing information can be inadequate to reduce uncertainty, and uncertainty might persist, rendering meaningful TD measurement impossible. We argue that one approach to mitigate this issue is to plan for a strategic investment under uncertainty. We leverage the concept of the VOI analysis to establish an information strategy of TD under uncertainty (Section 3.4.4). Our approach values the identified TD by considering its potential gain obtained when additional information is introduced.

Collectively, these three steps use various tactics to strategically mitigate Information Uncertainty in TD. As an immediate tactic, Step 1 promotes the use of existing information to reduce Information Uncertainty in TD. For an intermediate

tactic, Step 2 helps in eliciting information and using it to reduce Information Uncertainty in TD. For a long-term tactic, Step 3 prioritises Information Uncertainty in TD for future information investment. These steps can be iteratively utilised when more information is available to practitioners.

3.4.1 Goal Modelling and Obstacles Analysis for Technical Debt

We utilise goal modelling (KAOS) and obstacle techniques as the foundation of our model. As the goal modelling approach provides a precise criterion for sufficient completeness of a requirement specification [64], it is theoretically feasible to produce and probe a requirement specification for the perfect information (i.e. *domain-complete* [26]). By utilising the goal modelling, we postulate that our TD model can be traceable to its corresponding system goal and goal refinements, which facilitate practitioners in falsifying identified and measured TD and supporting them in the search for missing/hidden TD.

For a given system, there can be many goals. A goal is an objective the system should meet, and it captures the desired behaviours of the system. Each goal can have many sub-goals via *AND-* or *OR-refinements* [124]. An assertion to a goal G in the domain is considered to be an *obstacle* if it is consistent in the domain and it renders a goal G non-satisfiable [124].

We view TD as a shortcut that yields a short-term gain in exchange for an eventual obstruction of a system goal. An obstruction of TD can relate to a functional or a non-functional system goal, where the degree of obstruction relies on the system context, the manifested location of TD, the maturity of the system, the system usage, etc. For example, a bottleneck resulting from a hard-coded server address can degrade the maintenance of a performance goal in a system with a light workload

(e.g. a server with a fixed number of clients) or can render a goal on the availability of the system completely unsatisfiable in a system with highly-scalable traffic (e.g. an API). By likening TD to obstacles, decision-makers can utilise goals and obstacles analysis to investigate the relationship/rationale of TD to the parent/leaf goal and possible siblings they are likely to obstruct, along with measurable consequences. These can motivate decision-makers to explore the design space for potential TD that can obstruct goals, better understand the severity of obstructions and their propagation, and the urgency for debt refactoring. It can also help in evaluating the appropriateness of proposed TD identification approaches or measurement methods for a given case. We take inspiration to view TD as a special type of obstacle from their similar characteristics, which are addressed below and briefly summarised in Table 3.2:

• **Causes of obstacles:** Obstacles capture "undesirable conditions" that can affect goal satisfaction[124]. The causes of obstacles can be numerous and can be linked to external, internal or managerial factors; they can be rooted in misinformation and inadequate knowledge about the domain, undesirable states, threats, hazards or non-optimal methods for operationalising goals. Additionally, as an obstruction condition can be satisfied by an assertion within the domain problem, an obstacle can apply to issues at the development level. This is similar to TD, which can stem from various factors in any development level [75]. The resulting issues are sometimes absorbed (e.g., obstacle tolerance) to balance for other trade-offs, as is the case for taking or absorbing a debt in favour of other trade-offs.

• **Context-dependent :** The system context is an indicative factor of the significance of an obstacle. The frequency of a system goal to be utilised can indicate the likely severity of the obstacle, should an obstacle be manifested. In the case of TD, a similar argument can apply. The analysis can particularly help in predicting the interest on the debt, severity and likely debt accumulation over time.

Table 3.2: A Summary of Mapping of Obstacle and Technical Debt Using an Example: Performance Goal

	Obstacle (e.g. An obstacle to a performance goal)	TD (e.g. A debt observed on performance degradation)
Characteristics	- Can result from issues at any development level and rooted in misinformation (e.g. Inadequate knowledge about the said domain's performance requirements, inflated or idealised assumptions about the performance requirements, threats or hazard obstructing slowing down the performance). - Context-dependent (e.g. Its severity relates to the importance of this goal and the related goal refinements (i.e. performance and throughput) to the system, as well as the probability that the obstacle will obstruct the goal).	- Can result from issues at any development level (e.g. Managerial decisions leading to taking shortcuts in performance testing to meet time to market; shortcuts in requirements engineering and domain analysis for performance; sub-optimal design for performance; performance-related code; managerial decisions in delegating performance engineering tasks to less experienced engineers for cost savings; and pressure to accelerate the velocity of development and meeting the time to market, etc). - Context-dependent. For example, its severity relates to the manifested location(s) and where the debt is manifested and can be observed (e.g. Service Level Agreements violation of performance; end-user performance dissatisfaction; impact on other qualities such as meeting the availability requirements and subsequent business implications leading to loss of customers, loss of reputation, customer complaints and disputes leading to compensation or legal actions).
Outcome	- Obstruction of goal (e.g. The increasing service time of the system or at the specific process(es), which violates the corresponding goal, and, in the worst case, can render that goal non-satisfiable.)	- When the debt is mistreated, the interest can grow over time and its severity can be implied from the growth rate of the interest, yielding additional costs to mitigate. If the debt on performance is mistreated, it can also lead to degradation to system attribute(s) that relates to the said goal(s) and violations/obstructions of corresponding goal(s). For instance, higher cost to mitigate; longer processing time in manifested component(s) rendering the inaccessibility of manifested components within a specific time and thus the possible failure to meet the system goal(s).
Outcome characteristics	- Relative to its parent goal (e.g. The performance related-obstacle can violate a goal on the overall operation time and/or obstruct the goal to complete process(es) in time.)	- Can affect one or several goal(s) (e.g. The inaccessibility of manifested component(s) within a period of time can result in the obstruction of performance goal(s), as well as the obstruction of reliability-related goal(s) as it renders the overall system less reliable.)
Resolution strategies	- Elimination: A class of strategies which aim to render the obstacle infeasible. - Reduction: Strategies in this class similar to Obstacle Elimination strategies but with the aim to reduce the occurrence of the obstacle. - Tolerance: A class of strategies which focus on handling the obstacle consequences.	- Total Repayment: A class of strategies which aim to make the debt irrelevant (e.g., Restructuring the system adhering to performance design patterns.) - Partial Repayment: Strategies in this class similar to Total Repayment strategies but with the aim to reduce the debt impact (e.g, Performance tuning.) - Tolerance: A class of strategies which tolerate the debt, but focus instead on handling the debt impact (e.g. Offering the discount for service subscription to end-user as it costs lower than repaying the debt.)

• **Resolution strategies:** Several obstacle resolution strategies can be briefly categorised as obstacle elimination/reduction/mitigation strategies and obstacle acceptance/tolerance strategies [124]. TD resolution strategies can be categorised in a similar manner: TD mitigation/reduction by repayment and revisiting the debt acceptance/tolerance [76]. Table 3.3 presents a detailed mapping of obstacle and TD resolution strategies that are documented to assist in TDM and its derivatives, including self-admitted TD.

• **Outcome of TD:** TD can contribute to the system's degradation and eventual obstruction if the debt is mistreated or the interest on the debt significantly accumulates. Similar to obstacles, the likelihood and severe consequences of TD can be difficult to predict due to uncertainty from the system's internal and/or external factors.

Apart from those similarities, several differences between TD and obstacles are found as follows:

• **Dynamic and multifaceted impact of TD:** In goals and obstacles analysis, the impact of obstacles is associated with its parent system goal. However, TD can correspond to different system goals dynamically, and one TD can impact several aspects of the system (e.g. maintainability, evolvability, etc.) at the time.

• **Growth of TD impact:** In goals and obstacles analysis, the severity of obstacles is indicated by the obstacle's probability of obstructing its parent goal at present. In contrast, the severity of TD is indicated by TD interest, which relates to the situation when TD is mistreated in the near future.

Based on these observations, we view TD as a special type of obstacle; it can cause by a trade-off, be self-admitted, and be mitigated by resolution strategies;

Table 3.3: Mapping of Obstacle and Its Corresponding Technical Debt Resolution Strategies

	Obstacle Resolution Strategies	Technical Debt Resolution Strategies
Obstacle Elimination/Total Debt Repayment	Goal Substitution: Identifying an alternative goal which nullify the obstructed goal and its obstacle (e.g. Replacing an obstructed goal by an outdated document with a goal to ask for updates).	Component Replacement: Replacing/revising a manifested component to nullify or lessen the manifested TD (e.g. Refactoring/rewriting a debt-infested component [133]).
	Agent Substitution: Considering an alternative assignment which makes the obstacle irrelevant (e.g. Assigning an obstructed goal from an outdated document by users to the automated system).	Component Relocation: As shown in [135], relocating a manifested component to nullify or lessen its TD (e.g. Redirecting API calls/return statements to avoid a debt-infested component).
	Obstacle Prevention: Adding a new goal to avoid the obstacle. (e.g. Adding a goal for reminding the user to avoid an obstructed goal due to the forgotten task).	Workaround Development: As shown in [135], developing an additional component/an extra process to avoid the debt (e.g. Adding a pre-condition to ensure that a call does not trigger a debt-infested component).
	Goal De-idealisation: Transforming the goal that renders the obstacle disappears (e.g. Relaxing an obstructed goal from an unsent message by increasing the waiting time from 2 to 5 mins).	Debt Criteria Re-adjustment: As mentioned in [38], adjusting the TD criteria/threshold to only be relevant to the current TDM needs (e.g. Lowering an identification criterion by cyclomatic complexity when identifying TD).
	Domain Transformation: Transforming the domain which renders the obstacle disappears. (e.g. Recording in a timestamp automatically to eliminate an obstacle from a missing timestamp).	Outsourcing: Similar to Debt Conversion in [25], delegate the scope of responsibility to lessen or eliminate the debt (e.g. Utilising a 3rd party component).
Obstacle/Debt Tolerance	Goal Restoration: Adding a new goal to ensure the existing one to be satisfied again in the near future (e.g. Re-sending a message when an obstructed goal from the unsent message occur).	Future Debt Repayment: As in [133], establishing a plan for repaying the debt in the near future (e.g. Planning for repaying debt in the next iteration).
	Obstacle Mitigation: Adding a new goal to alleviate obstacle consequences. (e.g. Enabling a load balancer when an obstructed goal occurs due to the overwhelmed workload).	Debt Consequences Alleviation: As shown in [135], initialising a plan for easing the debt impact (e.g. Adding a method to sanitise a file produced by a debt-infested component).
	Do-nothing: Accepting the consequences of the obstacle as there is no significant impact (e.g. Accepting an obstructed goal due to the system bottleneck since its consequences are insignificant).	Do-nothing: Accepting the debt consequences as there is no significant impact (e.g. Accepting the debt since its expected benefit does not cover its re-engineering cost [25]).

it can render a goal non-satisfiable and can degrade software qualities attribute(s). However, not all obstacles are debts or are stemmed from shortcuts and false acceleration of development activities, but they can be a result of inadequate understanding of the context domain (i.e. rooted in the inadequacy of information).

3.4.2 Probabilistic Model of Technical Debt

In this section, we presented an extension of the TD model, which establishes a connection to goal-obstacle modelling analysis. Table 3.4 provides detailed notation for the contributions, including parameters and calculations in relation to our goal-obstacle-based TD model. We introduced a set of TDs denoted as T, where each TD

Table 3.4: Table of Notation

Variable	Definition
T	A set of TD. T = $\{TD_1, ..., TD_n\}$.
Q	A set of Quality Attributes. Q = $\{Q_1, ..., Q_o\}$.
S	A set of scenarios. S = $\{S_1, ..., S_p\}$.
$Pr(TD_j)$	Probability in which jth TD obstructs its related goal. Between 0 and 1.
$Pn(TD_j$	Principal of jth TD.
$R(TD_j, Q_k)$	Interest of jth TD, related to Quality Attributes k. Between 0 and 1.
$W(TD_j)$	Weight of jth TD. Between 0 and 1. $\sum_{j=1}^{n} TDj(t)^w = 1$
$Bn(TD_j, S_l)$	The range of minimum and maximum potential benefit from jth TD in Scenario l.
$W(Q_k)$	Weight of Quality Attribute k. Between 0 and 1. $\sum_{k=1}^{o} Q_k(t)^w = 1$
$Pr(S_l)$	Probability in which lth Scenario to occurs. Between 0 and 1.
$W(S_l)$	Weight of lth Scenario. Between 0 and 1. $\sum_{l=1}^{m} S_l(t)^w = 1$
$R(S_l, Q_k)$	Accumulated interest of lth Scenario in kth Quality Attribute aspect.
$Bn(S_l)$	The range of minimum and maximum potential benefit from lth Scenario.
$SS(S_l)$	Score of lth Scenario.
$\overline{CMS}(TD_j)$	Commonality Score of jth TD.
$\overline{CPS}(TD_j)$	Average Completion Score of jth TD.
$CPS(S_l, TD_j)$	Completion Score of TD j in Scenario l.
$\overline{IPS}(TD_j)$	Average Impact Score of jth TD.
$IPS(TD_j, Q_k)$	Impact Score of TD j on Q_k.
$TDS(TD_j)$	Technical Debt Score for jth TD.
$CSS(TD_j)$	Cost-Saving-based Score for jth TD.
$ITS(TD_j)$	Interest-based Score for jth TD.
$E[x]$	Expected value of x.

in the set consists of four parameters: its principal ($Pn(TD_j)$), obstruction probability ($Pr(TD_j)$), interest to quality attribute ($R(TD_j, Q_k)$), and weight ($W(TD_j)$). The specific details for these parameters are provided below.

Our modelling shares similarities with the traditional TD model in terms of its principal and interest factors [103]. However, our approach introduces a new parameter called obstruction probability ($Pr(TD_j)$), which is borrowed from obstacle analysis [26]. This parameter represents varying degrees of obstructions to the system goal based on the specific domain, system context, manifested location, and other relevant factors. It encourages decision-makers to consider the potential behaviours of the debt in connection with the system goal, its impact on different components and their interactions, and how many possibilities these factors could lead to an obstruction of the system goal. Essentially, our modelling helps to conceptualise the likely behaviour of the debts in relation to the system, indicating the need to mitigate TD in order to achieve the corresponding goals.

Our model explicitly maps the interest of each TD to specific quality attributes (Q_k), going beyond just maintainability and evolvability. This allows practitioners to consider other qualities, such as security or performance, in relation to their associated non-functional goals. Additionally, we introduce a weight parameter for each quality attribute ($W(Q_k)$), allowing decision-makers to rank the relative importance of these attributes based on their relevance to the system. The consideration and prioritisation of TD impact outside of maintainability and evolvability is not entirely new. A study by [51] has shown that decision-makers may make different choices when considering the security implications of TD, potentially prioritising security higher than maintainability and evolvability.

Nevertheless, we recognise that the obstruction of system goals can also stem from functional aspects, and so we introduce a weight parameter for each TD ($W(TD_j)$). This weight represents the relative importance of the TD, independent of its future impact on quality attributes or refactoring effort. The weight can reflect the severity of related system goals, customer requirements, and profitability of manifested components, among other factors.

As our model separates the consideration of TD's impacts on various quality attributes and its association with goals, it can be challenging to quantify the overall impact in a single metric. To address this issue, we propose the computation of a TD impact score for each TD. The score is calculated as a weighted summation of the obstruction probability to the corresponding goal and its interest to quality attribute k:

$$IPS(TD_j, Q_k) = (Pr(TD_j) \times W(IPS, a)) + (R(TD_j, Q_k) \times W(IPS, b)) \quad (3.1)$$

Whereas $W(IPS, a)$ and $W(IPS, b)$ are the weight (both are 0.5 as default, indicating that TD impact on various quality attributes and its association with goals are equally important). All parameters are represented as a single value, which can be obtained from its expected value, experts or a combination of both. We define the average TD impact score as the summation of the individual impact scores for each quality attribute k:

$$\overline{IPS}(TD_j) = \sum_{k=1}^{o} IPS(TD_j, Q_k) \tag{3.2}$$

The probability of obstructing the system is refined with the *AND*/*OR* refinement tree and propagation rules from [26]. There can be leaf and root TD in the system, and the severity of the violation of goal G (i.e. $SV(G)$) as $SV(G) = RDS(G) - EPS(G)$ where $RDS(G)$ is the required degree of satisfaction of goal G and $EPS(G)$ is the estimated probability of satisfaction for a goal G [26].

3.4.3 Scenario-based Framework for Technical Debt

Complementing goal-modelling [110], scenario requirements engineering provides a way to make the intention of the system-to-be clearer by reasoning on specific details from a set of scenarios. Each scenario can be considered as a sequence of behaviours in the system-to-be in a specific environment.

We utilise scenario-based analysis to help practitioners in falsifying and verifying TD parameters. More precisely, by concretising different groups of assumptions, perspectives, situations, etc., into scenarios, it is possible to reason about the appropriateness of TD parameters within the given scenario context, helping practitioners to falsify or verify the existing parameters and calibrating them when it is necessary.

As each system scenario might not be equally important, each scenario has unique properties (e.g. $Pr(S_l)$, $W(S_l)$, $R(S_l, Q_k)$) to indicate its significance. Scenarios with commonalities can be combined as "normal" scenarios, and the number of alternative scenarios depends on system complexity and safety criticality [110]. Each scenario can have the net interest of all quality attributes. For S_l, its net interest can be calculated from the summation of each quality attribute and its weight:

$$R(S_l) = \sum_{k=1}^{o} R(S_l, Q_k) \times W(Q_k) \tag{3.3}$$

The significance of each scenario can be calculated via the Scenario Score, which is the weighted summation of its net interest, its obstruction probability and its weight. For S_l, the Scenario Score can be calculated by:

$$SS(S_l) = (R(S_l) \times W(SS, a)) + (Pr(S_l) \times W(SS, b)) + (W(S_l) \times W(SS, c)) \tag{3.4}$$

Where $W(SS, a), W(SS, b), W(SS, c)$ is the weight of attributes (equal to 0.4, 0.5, 0.1 respectively by default, indicating the obstruction probability of the scenario is more important than its net interest and its weight).

In this thesis, we propose a calculation for determining the significance of TD based on its importance and frequency of occurrence in various scenarios. The importance of TD is represented by its impact score ($\overline{IPS}(TD_j)$) and its weight ($W(TD_j)$), as discussed previously. In the following paragraphs, we elaborate on the relationship between TD and scenarios, as well as the calculation process for TD occurrence in different scenarios, ultimately leading to the establishment of the TD

score.

A scenario consists of one or more TD, and each TD is considered a unique element. Thus, one TD can only exist once in a scenario. Here, we define a *common* TD as a TD, which is included in more than one scenario, and an *exclusive* TD as a TD, which is included only in one scenario. By doing so, there can be at least two TDM tactics: *Common debts-driven* (i.e. emphasising tackling common TD among scenarios) and *Scenario-driven* (i.e. focusing on mitigating exclusive debts in a scenario). The commonality score can be calculated to determine a suitable TD for the first tactic. The commonality score considers the coverage of one TD overall TD:

$$CMS(TD_j) = \frac{\sum_{l=1}^{p} \sum_{TD_j \in S_l} Pn(TD_j)}{\sum_{l=1}^{p} \sum_{TD_{all} \in S_l} Pn(TD_{all})} \tag{3.5}$$

Whereby $\sum_{l=1}^{p} \sum_{TD_j \in S_l} Pn(TD_j)$ is the summation of principal jth TD in each scenario which contains it, $\sum_{l=1}^{p} \sum_{TD_{all} \in S_l} Pn(TD_{all})$ is the summary of principal for all TD in all scenarios. For Scenario-driven, the average completion score for all scenarios and the completion score for lth scenario of jth TD (i.e. $\overline{CPS}(TD_j)$ and $CPS(S_l, TD_j)$ denote as:

$$\overline{CPS}(TD_j) = \sum_{l=1}^{p} CPS(S_l, TD_j) \times W(S_l) \tag{3.6}$$

$$CPS(S_l, TD_j) = \frac{\sum_{TD_j \in S_l} Pn(TD_j)}{\sum_{TD_{all} \in S_l} Pn(TD_{all})} \tag{3.7}$$

Whereby $\sum_{TD_j \in S_l} Pn(TD_j)$ is the summation of the principal of jth TD which

includes in lth scenario and $\sum_{TD_{all} \in S_l} Pn(TD_{all})$ is the principal summation from all TD which includes in lth scenario. A completion score for each scenario and each debt is determined by the coverage of one TD over the selected scenario.

Collectively, TD Score (TDS) for jth TD is the weight summation of its impact score, its weight, how common and how exclusive it is to defined scenarios:

$$\begin{aligned} TDS(TD_j) = (\overline{IPS}(TD_j) \times W(TDS, a)) + (CMS(TD_j) \times W(TDS, b)) + \\ (\overline{CPS}(TD_j) \times W(TDS, c)) + (W(TD_j) \times W(TDS, d)) \end{aligned} \tag{3.8}$$

Whereas $W(TDS, a), W(TDS, b), W(TDS, c), W(TDS, d)$ are the weight (by default, equal to 0.7, 0.1, 0.1 and 0.1, respectively).

3.4.4 Expected Value of Technical Debt Information

As mentioned in 3.2.3, the decision-making process in TDM and treatment selection in Health Economics is heavily influenced by evidence, in which decision-makers in both fields are assigned to prioritise issues that need to be mitigated. Furthermore, they can suffer from imperfect information, potentially obfuscating accurate assessment of the impact of the problem and proposed solutions.

Considering the above similarities, it is reasonable to postulate that techniques used in treatment selection to mitigate Information Uncertainty can be beneficial to those in TDM decision-making. In this section, we propose a technique based on information worthiness to manage Information Uncertainty in TD.

Inspired by the cost-effectiveness analysis in health economics, which considers the worthiness of information related to each decision, we propose an information-worthiness-based technique to manage Information Uncertainty in TD. This technique is based on the potential benefit of information to reduce Information Uncertainty in TD. This parameter is represented in a range as it can differ in different scenarios. Consider website traffic data, as an example, which can be of a high value during a peak scenario as it can help in verifying the impact of performance-related TD. However, it might be irrelevant and does not add value to non-workload-sensitive scenarios (e.g. the night shift).

Based on the range of potential benefits from the debt, it is possible to compute its worthiness via the expected value of information. This chapter presents an approach for indicating the maximum amount to gather information on all scenarios with the expected value of total perfect information:

$$EVTPI = E[max_{S_l \in S} Bn(S_l)] - max_{S_l \in S} E[Bn(S_l)] \tag{3.9}$$

Where $E[max_{S_l \in S} Bn(S_l)]$ is the expected value of the maximum benefit from Scenario lth. $max_{S_l \in S} E[Bn(S_l)$ indicates the maximum value from the expected benefit from Scenario lth. The value from EVTPI always is greater or equal to zero. The output of EVTPI is a monetary value that indicates the maximum amount to invest in reducing uncertainty for each scenario. Additionally, it is possible to prioritise the importance of Information Uncertainty in each TD in a scenario via the expected value of partial perfect information:

$$EVPPI = E[max(Bn(S_l, TD'_j)|Bn(S_l)]) - max_{S_l \in S} E[Bn(S_l)] \tag{3.10}$$

Where $E[max(Bn(S_l, TD'_j)|Bn(S_l)])$ is the expected value of the maximum benefit of jth from all TD included in lth scenario. Intuitively, it would be reasonable to prefer reducing the uncertainty from TD in the selected scenario, which has the most expected benefit. Like EVTPI, EVPPI of TD can indicate the maximum amount to invest in reducing the uncertainty of that debt.

3.5 Evaluation

The evaluation of uncertainty-related approaches, in general, can be difficult. This is because the presence of uncertainty is context and environment dependent. Additionally, statistical evaluation of uncertainty requires the presence of uncertainty-free values and/or ground truth. Approximation error or relative error, for example, requires an uncertainty-free value which can be difficult and impractical without knowing the ground truth or establishing it from scratch (e.g. one's own data set, backlog, environment, etc.). Establishing ground truth can potentially suffer from infinite regression (e.g. whether an identified/measured value is actually uncertainty/error-free). To address these challenges, the evaluation uses a controlled experiment to evaluate the effectiveness of our approach against two criteria, predictability and robustness in the presence of uncertainty:

- Predictability: An effective approach that caters for uncertainty should be predictable. The premise is that the more certain the approach is, the more predictable of its results. The predictability of the approach can be in relation to uncertainty from Residual Variability, Parametric Variability, and Information Uncertainty. Predictability can also imply fewer outcome variations, and as there are fewer input-output pairs in the search space, it can decrease the cost to reduce Information Uncertainty. Predictability can also help to deter-

mine uncertainty from Model Inadequacies, as there are fewer outputs to be compared.

- Robustness: As Observation Error is one of the sources of uncertainty. An ideal approach must adequately tolerate the possible error during observation. This requirement can also apply to uncertainty from Parameter Uncertainty and Parametric Variability as tolerance towards input variations can reduce the chance of uncertainty from those two sources to be propagated.

The evaluation aims to answer the below questions:

RQ1: In the presence of uncertainty, can our approach produce more predictable outcomes compared with other approaches? If so, why is it the case?
RQ2: Can our approach produce more robust outcomes when compared to other approaches? If so, why is it the case?

For the controlled environment, we use an open-source microservice system of an industrial scale, described in 3.5.1, to evaluate our approach. To ease reproducibility and adoption, the evaluation firstly examines how the *mitigation decisions* of our approach are compared against the baseline of cost-savings [101] and hypothetical interest-based approaches in the presence of Information Uncertainty. The hypothetical interest-based approach prioritises TD primarily based on its interest. The purpose of this hypothetical interest-based is to mimic several COTS TD models [13], without the reliance on any specific COTS profile and human judgement. Both predictability and stability analyses of the mitigation decisions are presented.

Secondly, we report on the predictability and stability analyses of the *investment decisions* once informed by our approach, comparing it against that of the interest-based method. The cost-saving approach was not used as it essentially assumes that

the principal is uncertainty-free, making the comparison less meaningful. Evaluation material can be accessed via [119].

For meaningful comparison, we consolidate TD metrics as follows: for the cost-saving-based, the principal of all TD is translated to be its refactoring cost (i.e. $C_{rest} = Pn(TD_j)$), all scenarios and its probability are translated to evolution steps and its probability (i.e. $Pr(S_l) = p'(e)$). The relative TD is calculated by:

$$RTD(S, S', e) = Pn(TD_j) + W(S_l) \times Pn(TD_j) \times (W(TD_j) \times \overline{IPS}(TD_j)) \quad (3.11)$$

Additionally, we present the cost-saving ratio for each debt to rank each debt to be repaid as:

$$CSS(TD_j) = \frac{\sum_{e \in E'} RTD(S, S', e) \times p'(e)}{C_{rest}} \quad (3.12)$$

When $C_{rest} < \sum_{e \in E'} RTD(S, S', e)$, otherwise. $CSS(TD_j) = 0$. The ranking of this approach is calculated as:

$$ITS(TD_j) = (R(TD_j, Q_{main}) \times 0.5) + (R(TD_j, Q_{evo}) \times 0.5) \quad (3.13)$$

3.5.1 Case and Experimental Setup

Since the microservices architecture is often governed in a decentralised manner [73], the decentralised nature of microservices development can make some of the

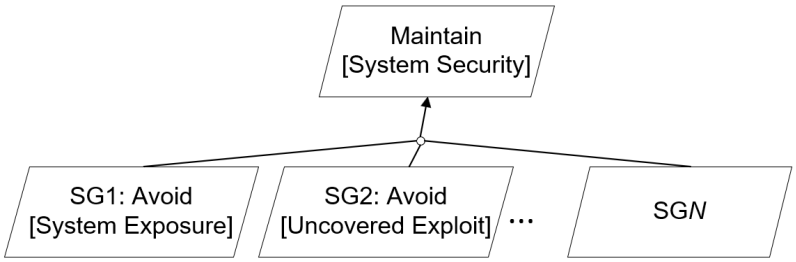


Figure 3.1: A high-level extract of AND goal refinement, SG1, SG2 of SGN.

TD information invisible, where a TD on one microservice and its root causes can be invisible to other developers. Though the intuition microservice design can confine and localise the debt in given services, the reality is that the debt can still propagate to affect the rest of the architecture. Consequently, TDM decision-making can suffer because of Information Uncertainty; for example, when the information flow that relates to the root causes of the debt, its nature, confinements and resolution strategies are invisible but may still obstruct the system (e.g. if we would think of debt as potential obstacles that can affect the satisfaction of the goal of the system, where goals might not necessarily link to a specific service). Henceforth, the decision-making can be incomplete and inaccurate. We use the eShopContainers project (commit 3573), which is a microservices system powered by Microsoft, to evaluate our approach. This project is the largest microservices system in [48]. To promote the reproducibility of the evaluation, we selected SonarQube as it supports all the programming languages used in our investigation (i.e. JavaScript, TypeScript, C#) without heavy modification and refuge to different profiles and tools. The principal of each TD is gathered by SonarQube CE v. 8.5.1.38104 with default Sonar-Way profiles for all languages. 14 TD from files with the highest principal are gathered. 9 TD are code debts, two of them (TD10 and TD13) are architectural TD from hardcoded paths, and three of them (TD2, TD5 and TD14) are test debt due to uncovered test cases. As SonarQube is unable to estimate the refactoring effort of test debt based on available information, test debt requires different treatment and is excluded from TDS calculation and TD backlog in Table 3.5. We analyse test

debt VOI for establishing an information investment strategy (Table 3.10), planning towards their effort estimation in the future. Quality attributes are maintainability, evolvability and security, whereas their $W(Q_k)$ are the normalisation of the converted 3-point Likert scale (i.e. 0.167 (1/6), 0.333 (2/6) & 0.5 (3/6) respectively). The security impact of TD is considered in this evaluation, along with the classical TD impact on system attributes (maintainability and evolvability). Table 3.7 shows file properties that are required to calculate TD interest in this case experiment. Maintainability Rating (i.e. Rating) is converted from SonarQube with the setting above to the decimal format. The weight of each file is converted from the curated 3-point Likert scale. For reproducibility, equations for calculating certain parameters are provided. *EvoPot* is an evolvability potential, and $Crit$ is the criticality of each file. All parameters in this case experiment, including the interest of each TD, are gathered via the expected value obtained from Monte Carlo Simulation ($n = 10^5$) with its corresponding range in the bracket. The interest of jth TD in Maintainability, Evolvability and Security is calculated as:

$$R(TD_j, Q_{main}) = Fi(W(TD_j)) \times Fi(Rating(TD_j)) \tag{3.14}$$

$$R(TD_j, Q_{evo}) = Fi(W(TD_j)) \times E[Fi(EvoPot(TD_j))] \tag{3.15}$$

$$R(TD_j, Q_{sec}) = Fi(W(TD_j)) \times E[Fi(Crit(TD_j))] \tag{3.16}$$

Whereas $Q_{main}, Q_{evo}, Q_{sec}$ is maintainability, evolvability and security and $Fi(x)$ is the file location of x. The partial goal and two sub-goals (Figure 3.1) of the case experiment are refined with "divide-and-conquer" refinement patterns [33]. Table

3.5 shows the parameters of TD, and Table 3.6 shows the parameters of identified scenarios. It is assumed that all TD are related to its parent goal with $OR - refinement$, and the severity of the violation of the goal is negligible. "Weight" column in the table represents the normalised weight of each debt, its minimum, maximum and its expected value before normalisation. "Pr" indicates $E[Pr(TD_j)]$, and the range of minimum and maximum $Pr(TD_j)$. "Rationale" column represents the rationale behind the range of probability and mapped scenario(s). "Pn" column represents the principal of each debt. Table 3.6 shows that Scenario 1 and residing TD should be firstly mitigated. Based on TDS and an interest-based approach, Table 3.8 illustrates that for Scenario 1, TD7 should be firstly mitigated. However, the three approaches reported vastly different TD. Practitioners must be mindful of the ability of the cost-saving-based and interest-based approaches to encapsulate TD impact. Table 3.9 illustrates the number of samples from three scenarios, their range, the mean, and the maximum potential benefit across scenarios. Table 3.10 shows the range of potential benefits of each debt and its EVPPI across scenarios. The potential benefit ranges are based on various degrees of saving from the loss of business opportunity and extended downtime [84, 39, 131]. EVPPIs from this table are obtained by a technique in [99]. From Table 3.10, the uncertainty in parameters of TD2 should be firstly reduced since it yields the highest EVPPI value.

Table 3.5: Parameters of Identified Technical Debts

TD_j	Source	Pn	Goal	Description	Pr	Rationale	Weight	$\overline{TPS}$	CMS	$\overline{CPS}$	TDS
1	F1	180	SG1	Variable visibility issues	0.6 (0.5,0.7)	Medium data relevancy	0.055 (0.1,0.3,0.228)	0.354	0.343	0.192	0.307
3	F2	30	SG2	Unused variable	0.4 (0.35,0.45)	Workload sensitive	0.032 (0.1,0.15,0.133)	0.213	0.019	0.007	0.155
4	F2	260	SG2	Exception in user-side code	0.7 (0.6,0.8)	Workload & Downtime sensitive	0.115 (0.4,0.6,0.481)	0.363	0.33	0.067	0.306
6	F3	10	SG2	Unnecessary Libraries	0.14 (0.1,0.2)	Well-tested libraries	0.078 (0.2,0.45,0.324)	0.174	0.006	0	0.013
7	F3	20	SG1	Unstandardised Coding Pattern	0.76 (0.7,0.8)	Can be error-prone	0.13 (0.5,0.6,0.541)	0.484	0.038	0.021	0.358
8	F4	90	SG2	Variable visibility issues	0.37 (0.2,0.5)	Low impact from exploit	0.019 (0.05,0.1,0.079)	0.205	0.172	0.096	0.172
9	F5	4	SG2	Unnecessary Libraries	0.75 (0.7,0.8)	Downtime sensitive	0.042 (0.1,0.25,0.174)	0.395	0.003	0	0.281
10	F5	40	SG2	Hardcoded Paths	0.65 (0.6,0.7)	Workload sensitive	0.107 (0.4,0.5,0.448)	0.345	0.025	0.009	0.256
11	F5	10	SG2	Variable visibility issues	0.47 (0.4,0.55)	Downtime sensitive	0.064 (0.2,0.35,0.268)	0.255	0.019	0.011	0.188
12	F6	10	SG2	Variable visibility issues	0.13 (0.1,0.15)	Low impact from exploit	0.049 (0.15,0.25,0.203)	0.079	0.019	0.011	0.063
13	F6	40	SG2	Hardcoded Paths	0.28 (0.25,0.3)	Low impact from exploit	0.054 (0.2,0.25,0.224)	0.154	0.025	0.009	0.117

Table 3.6: Table of Identified Scenarios

S_l	Description	$W(S_l)$	E$[Pr(S_l)]$	$TD \in S_l$	$SS(S_l)$	Ranking
1	Casual	0.333 (2/6)	0.752 (0.7,0.9)	TD1, TD7, TD8, TD11, TD12, TD14	0.623	1
2	Seasonal	0.5 (3/6)	0.301 (0.2,0.4)	TD1, TD2,TD3, TD4, TD7, TD8, TD10, TD11, TD12, TD13, TD14	0.476	2
3	Nighttime	0.167 (1/6)	0.124 (0.1,0.15)	TD1, TD4, TD5, TD6, TD7, TD8, TD9, TD11, TD12	0.161	3

Table 3.7: Table of File Properties

ID	Filename	Weight	Rating	$E(EvoPot)(min, max)$	$E(Crit)(min, max)$
F1	OrderStatus.cs	0.2 (Medium)	0.6 (C)	0.163 (0.1,0.3)	0.774 (0.75,0.8)
F2	CatalogContextSeed.cs	0.1 (Low)	0.2 (A)	0.316 (0.2,0.4)	0.252 (0.2,0.3)
F3	WebhooksDomainExceptions.cs	0.3 (High)	0.4 (B)	0.773 (0.75.08)	0.734 (0.7,0.8)
F4	CardType.cs	0.2 (Medium)	0.2 (A)	0.074 (0.05,0.1)	0.292 (0.6,0.4)
F5	Locations.API/Program.cs	0.1 (Low)	0.2 (A)	0.18 (0.1,0.25)	0.616 (0.5,07)
F6	Marketing.API	0.1 (Low)	0.2 (A)	0.514 (0.4,0.6)	0.154 (0.1,0.2)

3.5.2 RQ1: Predictability Analysis

Decisions from all approaches are evaluated with Shannon's entropy [105] as

$$H(X) = -\sum_{j=1}^{m} p(TD_j) log_2 \times p(TD_j) \tag{3.17}$$

Where $H(X)$ is the entropy of X, and the lesser value means, the more certain of the overall outcomes, $p(TD_j(t))$ is the probability that TD_j will be chosen based on its ranking. For example, if TD_2 is ranked as the number 1 among all 3 debts, $p(TD_2)$ is $(3/(1+2+3)) = 0.5$ whereas the dividend (i.e. 3) is its descending order. For mitigation decisions, the result of Table 3.8 shows that our approach outperforms those two in all scenarios, which is roughly around 18.6 per cent and 26.3 per cent less entropy compared with our average (2.634). **To answer RQ1,** our approach performs better for producing predictable outcomes with lower entropy. This is because our approach utilises information from the system context and scenarios; decision-makers can thus make mitigation decisions with more confidence and omit irrelevant TD from the scenario of interest. For investment decisions, Table 3.10 shows that decisions of both approaches achieve an entropy of 1.459.

Table 3.8: Ranking of technical debt decisions from Probabilistic model, Cost-saving-based (Csv) and Interest-based (Intr) approaches and its entropy

TD	Csv	Intr	Probabilistic Model (TDS)					
			S_1	$p(S_1)$	S_2	$p(S_2)$	S_3	$p(S_3)$
1	1	4	2	0.267	2	0.178	2	0.194
3	5	8			7	0.067		
4	11	3			3	0.156	3	0.167
6	4	10					7	0.056
7	2	1	1	0.333	1	0.2	1	0.222
8	6	9	4	0.133	6	0.089	6	0.083
9	8	2					4	0.139
10	9	5			4	0.133		
11	3	6	3	0.2	5	0.111	5	0.111
12	7	12	5	0.067	9	0.022	8	0.028
13	10	11			8	0.044		
$H(X)$	**3.236**	**3.574**		**2.149**		**2.957**		**2.794**

Table 3.9: Expected potential benefit from all scenarios. their EVTPI is 33789.69 - 23238.17 = 10551.52

S	S1 (20659,280605)	S2 (18691,46450)	S3 (13778,32695)	Max
1	23985.04	22627.3	28672.12	28672.12
2	21218.47	36046.71	19947.62	36046.71
		...		
X	24623.9	32504.2	23238.17	33789.69

Table 3.10: Ranking of Technical Debt Decisions by EVPPI and Interest-based Decision-making

TD	EVPPI				Intr
	S1	S2	S3	$\bar{X}$	
2	0.00 (9279,11191)	32.73 (6881,21724)	0.00 (4499,14193)	**10.91**	0.238
5	0.00 (6881,10533)	2.18 (4499,14193)	0.09 (4780,11191)	0.76	**0.073**
14	0.27 (4499,6881)	0.44 (7311,10533)	0.00 (4499,7311)	0.24	0.078

3.5.3 RQ2: Stability Analysis

Decisions from different approaches are evaluated via sensitivity analysis with the quadratic-mean decision function [37]:

$$(\frac{\sum_{i=1}^{n} w_i a_i^{\alpha}}{\sum_{i=1}^{n} w_i})^{\frac{1}{\alpha}} \tag{3.18}$$

Whereas $\alpha = 2$, $w_i = W(S_l)$, $a_i = CSS(TD_j)$ for the cost-saving-based approach, $a_i = ITS(TD_j)$ for the interest-based approach and $a_i = TDS(TD_j)$ for the probabilistic model approach. The stability index with ϵ perturbation can be calculated as [37]: $S(\epsilon) = (\prod_{j=1}^{n-1} \delta_j)^{\frac{1}{n-1}}$ whereby $S(\epsilon)$ range between 0 (i.e. the unstable solution) and 1 (i.e. the stable solution). δ_j is the pair-wise stability value between j and $j+1$. Since different approaches yield different numbers of debts and orders, they are analysed individually. $\epsilon = 0.1, 0.3, 0.5, 0.8$ and 1 (i.e. there are parametric changes up to ϵ in each decision) are utilised to evaluate all decisions. **To answer RQ2,** we conclude that our approach is more robust than other approaches because it produces decisions with higher sensitivity indices in all degrees of parametric perturbation (shown in Table 3.11). In the extreme perturbation ($\epsilon = 1$), decisions from our model achieve the stability index at least two times more than the baseline approaches. This is because our model is based on more information (i.e. parameters of debts) to prioritise TDS. Thus, it is more tolerant to perturbation. For investment decisions, Table 3.12 indicates that the decisions from EVPPI are also more stable than the ones from the interest-based approach. This likely is because the inputs of the analysis are the expected values, and therefore they are calibrated and more tolerant to sensitivity.

Table 3.11: Stability Indices of different TD decision-making approaches

ϵ	Cost-saving-based [101]	Interest-based [13]	Probabilistic Model		
			S1	S2	S3
0.1	0.273	0.33	**1**	**0.618**	**0.75**
0.3	0.114	0.145	**0.63**	**0.256**	**0.356**
0.5	0.073	0.094	**0.446**	**0.163**	**0.229**
0.8	0.049	0.063	**0.306**	**0.107**	**0.152**
1	0.041	0.04	**0.256**	**0.088**	**0.125**

Table 3.12: Stability Indices of between EVPPI and interest-based approach

ϵ	EVPPI	Interest Based
0.1	**1**	0.81
0.3	**1**	0.561
0.5	**1**	0.474
0.8	**0.866**	0.348
1	**0.768**	0.3

3.6 Discussions and Threat to Validity

As an RQ1 answer, Table 3.8 illustrates that our model can produce at least 18 per cent more predictable TD decision ranking compared to outputs from the classical interest-based and cost-saving-based models. This is because the ranking of TD decisions produced by our approach is categorised by scenarios, which can probe for and reveal additional cases and information that analysts may miss without concretising the case. This advantage enables practitioners to prioritise the criticality of TD by observing the number of manifested scenarios for each TD and its related system goal. These abilities can support practitioners in locating and mitigating critical TD before their negative consequences occur.

To answer RQ2, Table 3.11 and 3.12 showed that our model is capable of producing TD decisions that are significantly more preserved results to parametric perturbation when compared to classical interest-based and cost-saving-based models. In the worst case, the stability index from our approach is one time higher than the rest. This demonstrates that a more stable TD decision ranking can be produced when considering additional system contextual information. The information utilised in our model can also provide practitioners with a clearer picture of TD

impact when better associated with system goals and manifested scenario(s). Based on the number of manifested TD, practitioners can see the location which deserves attention to maintain its quality when using our approach.

Additionally, our model encourages practitioners to input additional system contextual information as the model's parameters and thus make some information related to system context more explicit. This enables better traceability between the TD model and TD decision when practitioners use our approach. This advantage might not be possible with the classical TD models, where TD decisions might base on illicit information and might not be traceable to utilised models.

Table 3.9 and 3.10 illustrate that the VOI analysis utilised by our approach can give an insight into investment for reducing Information Uncertainty in the monetary unit. This advantage can help practitioners in better TD communication between and within technical and managerial stakeholders. Transparency of the VOI analysis can also facilitate practitioners in revising the estimation of TD's potential benefit. Ultimately, our approach can facilitate decision-makers to decide whether it is worth investing in reducing Information Uncertainty for each TD when weighing in their potential benefit (i.e. from the VOI analysis) and their potential impact (i.e. considering TD parameters: interest, probability and its related goal). These advantages can be entirely infeasible with classical TD approaches.

For a threat to validity related to **applicability**, the identified TD in our case contains diverse types of debts (e.g. architectural debt, code debt, test debt, etc.); however, TD analysts may often focus on specific types in practice. We believe that our chosen microservice case demonstrates ample opportunities to test the practical potential of our contributions. Additionally, though our approach requires calculations for numerous parameters, the approach is flexible considering generic/widely used, customised, bespoke methods of calculation. This flexibility improves the

method's precision and applicability to various domains and contexts and beyond the microservices case. Our case demonstrates that various techniques can be employed for calibrating the precision of debts' parameters from different sources (e.g. a static code analysis, Monte Carlo Simulation, and other experts' elicitation techniques [14]). It also demonstrates that it is possible to prioritise debts in the presence of Information Uncertainty using the expected VOI analysis, even when all parameters cannot be estimated precisely.

The cost of seeking more information is not considered in this work as it varies with applications and contexts, making it hard to generalise. For example, the cost for introducing additional information is subject to utilised models (e.g. goal modelling, cost-effectiveness TD model), software documentation (e.g. software requirements, software scenarios) and methods to gather information (e.g. experts' opinion, COTS). Nevertheless, the cost of probing for more information can be traded against the added value of factoring in information in decision-making, which can be scenario and context-specific.

Following the common aphorism in statistics that "all models are wrong" [21], we do not claim absolute **correctness** of the model, but we provide sound models, metrics and systematic guidelines that facilitate in eliciting information for a given system context.

Our model is based on several TD publications on the impact of TD on the system context felicitated by systematic goal-obstacles modelling and VOI analysis. The combination provides decision-makers with transparent and evidence-based decision-making that can be traced back to the rationale of the goal-obstacles refinements and its context. Additionally, the evaluation metrics supporting our model are based on widely-used techniques in information theory, along with sensitivity and stability analysis, providing confidence and a basis for falsifying arguments linked

to the analysis. The analysis reveals that our model achieves higher stability indices across all degrees of perturbation when compared against other baseline approaches (cost-saving- and interest-based).

Though our model hopes to mitigate Information Uncertainty and to probe for information, **completeness** of information is difficult to achieve, and it is a widely acknowledged challenge. Since the completeness of the information can be highly subjective to perceivers and can be cost-intensive, it might not be feasible to obtain complete information in practice. Hence, the completeness of the information cannot be guaranteed. This chapter provides an approach to probe for *additional information* as more information can yield a more precise identification or estimate of TD. This, in turn, can reduce Information Uncertainty of TD during TD identification and measurement.

The aim of this chapter is to demonstrate that our approach has the potential to handle Information Uncertainty in extreme conditions where information and knowledge are scarce (e.g. greenfield projects). For this case, the steps of our approach can be iteratively applied, moving from simplified assumptions and best-effort estimates to more advanced analysis, precise estimates, and further tuning as we advance our knowledge of the case and are consistent with engineering management practices. However, we see that further research and quantification approaches of the parameters might be required, specifically for TDM of systems, experiencing Information Uncertainty which is a subject of our ongoing research. Nevertheless, our presented approach can serve as a solid foundation that can inform further studies on the topic of managing TD under Information Uncertainty in a systematic and unified manner.

Since the source of uncertainty illustrated in Section 3.2 can manifest in computerised models, we believe that Information Uncertainty of TD can manifest in different TDM activities. Considering Information Uncertainty during TD monitoring,

as an example, practitioners might suffer from uncertainty in regard to appropriate monitoring tools, monitoring sensitivity, etc., as sources of information related to those TD monitoring subjects can be unreliable, biased, myopic, etc. Our concept might be applicable to Information Uncertainty in different TDM activities. To illustrate, the location of system components and their related system goals might be capable of supporting practitioners in selecting suitable monitoring sensitivity, selecting appropriate monitoring tools, etc. Additionally, the practitioner might be capable of minimising negative consequences from improperly monitored TD when considering its monetary impact via the VOI analysis and utilising the analysis in support of TD monitoring decision-making. These further subjects can be related to the exploration of Information Uncertainty in different TDM activities and their possible alleviation, which are worth researching.

3.7 Conclusion

We present a systematic approach to reduce Information Uncertainty in TD. The approach adopts widely-used techniques in requirements engineering to establish a probabilistic goal and obstacles models of TD, complemented by a scenario-based analysis and information investment techniques from health economics. The objective is to reduce Information Uncertainty in TD through investment in information. Cost-benefit and the value of information models of TD are presented to establish an information investment strategy in information to reduce uncertainty. With the benefit-oriented information investment strategy, it is possible to prioritise which debts to consider and how much information should be invested in for a more effective TDM. The evaluation shows that our approach is capable of producing significantly less uncertain and more stable decisions under extreme parametric perturbations.

Chapter 4

MORDM-based Approach to Mitigate Deep Uncertainty

4.1 Introduction

Technical Debt (TD) is a consequence of taking the short-term benefit at the cost of the long-term well-being of the software system [22], which relates to costly or impossible future changes [12]. TD can cause both intentionally and unintentionally by developers [82]. Technical debt management (TDM) consists of a set of activities which aim to prevent potential TD from being incurred, keeping it visible and controllable [75, 76]. One of the crucial TDM activities is TD prioritisation. This TDM activity provides practitioners with a window of opportunity to review and verify the identified and measured debts and consequently decide on the priority for mitigating and re-paying the debts, constrained by time, resources and/or urgency. Existing prioritising approaches may utilise several ranking methods, which can be informed by costs, benefits, urgency, project constraints and/or experts' judgements

etc.

However, the diverse deciding factors, stakeholders' opinions, expertise and capabilities in decision-making can lead to Deep Uncertainty. Deep Uncertainty can be observed in scenarios where the parties of a decision do not know or do not agree on the system model(s), decisions and their consequences [67]. Consequently, the probability distribution to represent uncertainty about its key parameters and/or preference of desired outcomes [67] can result in Black Swan events - unexpected events which can yield significant impact on system [112]. This phenomenon can be observed in TD prioritisation: consider, for example, disagreements on TD prioritisation deciding factors; incapability in resolving the difference; indecision because of bias, myopic, ability to make decisions, and/or overly-/under-focused consideration of the TD issue.

In TD prioritisation, the presence of one or more of these issues can contribute to fragile TD prioritisation decisions and in-erratic TD rankings with potentially undesirable consequences. These issues can be observed in TD prioritisation of projects of any size, and their outcomes (i.e. sub-optimal decisions or indecision) can be fatal. The state-of-art and -practice lacks approaches for uncovering and systematically mitigating these issues. Worse, very few studies acknowledge potential disagreement among practitioners or the incapability of practitioners. As current shortcomings in the community leave practitioners to resolve their issues in decision-making on their own; the resolution can be highly vulnerable to Black Swan in TDM as they do not acknowledge Black Swan blindness (i.e. employing decision-making shortcuts/practices which can lead to Black Swan events) during TD prioritisation decision-making.

To systematically address those issues, the chapter explores the problem of Deep Uncertainty [67] and Black Swan events manifestation [112] in TD. We demonstrate

that it is inevitable for Deep Uncertainty to occur during TD prioritisation. Additionally, since TD prioritisation is related to decision-making on data with a specific type of deviation (i.e. Extremistan), as well as practitioners can utilise several shortcuts during decision-making without acknowledging their trade-offs, the probability for Black Swan events to occur can be exemplified in this phase.

For mitigating Deep Uncertainty and potential Black Swans that stemmed from TD prioritisation decisions systematically, we propose an iterative decision-analytic framework derived from Many-Objective Robust Decision Making (MORDM) [53], and Antifragility [113]. MORDM is an iterative Robust Decision-making analytic framework that uses Many-Objective Optimisation for generating Pareto-optimal solutions to be verified and discussed by practitioners. MORDM can also inform practitioners of potential vulnerabilities in selected solutions during the Scenario Discovery step. Antifragility is an ability to thrive when there are changes in the unknown environment [113]. Small stressors and optionality in the system are among the two causes of antifragility[113]. Additionally, unknown changes in the environment can help in mitigating Black Swan events[113].

In this chapter, TD rankings can be generated via the Many-Objective Optimisation Algorithm (MOEA), where the objective functions of the various scenarios under consideration have utility scores; these scores can be derived from the TD prioritisation approaches that practitioners select to inform the analysis.

In data-sensitive systems, it can be challenging to learn, imply and gather scenarios. To address this challenge, we take a Scenario Discovery approach, and we leverage the use of the "Formal Approach to Scenario Analysis"[46]. The discovery of the scenarios can help analysts to probe for vulnerabilities in TD rankings and would make it possible to initiate Scenario Discovery in an early phase of MORDM. This improvement can aid the generation of more focused utility and objective func-

tions and, consequently, more robust TD rankings that are scenario-based. The identified scenarios in our framework can act as small stressors and can help benchmark the performance of objective functions in relation to all scenarios. To promote optionality in our framework, we leverage the concept of Nash Equilibrium for the visualisation of objective values in different TD rankings. We use pay-off matrices to determine the TD rankings of different objective functions and their Nash Equilibria. In the case of unknown environments and uncertainty, TD rankings can be utilised for enacting optionality in decisions to mitigate likely Black Swan in TD prioritisation.

Collectively, our contributions in this chapter are as follows:

- We discuss Deep Uncertainty in TD prioritisation. We provide a novel mapping of the Black Swan theory into TD prioritisation. The mapping coins Black Swan elements and its blindness with selected TD activities, covering identification, measurement and prioritisation. The mapping discusses the inevitability of Deep Uncertainty and Black Swan in TD in general and prioritisation in specific.
- We formulate a systematic approach to mitigate Deep Uncertainty, leveraging an iterative decision-analytic framework driven by Many-Objective Optimisation.
- We present an approach to mitigate Black Swan in TD prioritisation decisions. We leverage the role of "stressors" and "optionality" analysis for mitigating Black Swan and converging into antifragility. Formal Scenario Analysis is used to generate scenarios that can serve as stressors. For optionality, Nash Equilibria-aided and pay-off matrices visualisation and analysis are used to determine the TD rankings that can mitigate for Black Swan and promote antifragility in TDM decision-making.

As the existence of Deep Uncertainty is context-dependent, a number of controlled experiments of non-trivial scale are used to analyse our approach and evaluate our framework. Apache Accumulo and Apache Cocoon have been selected as a project for the experiments; those are among the projects with the highest numbers of identified TD in the Curated Technical Debt Dataset [48]. We conduct statistical tests against the baseline approach with a focus on the ability to produce high-quality decisions, the similarity across decision space and improvability when changing from and to our approach. One-tailed Mann-Whitney U tests are evaluated across different approaches to ensure statistically significant evidence is found. The results show that outcomes from our approach yield 40 per cent higher quality than the baseline and approximately 70 per cent more similar decision space, which indicates the potential to alleviate Deep Uncertainty during TD prioritisation decision-making. The results also statistically indicate a potential to improve the existing TD prioritisation approach with our framework.

4.2 Background

4.2.1 Deep Uncertainty and Robust Decision Making

According to Lampert et al. [66], Deep Uncertainty is present in situations where parties involved in a decision do not know or disagree upon 1) the appropriate representation of interactions among the system's variables, 2) the probability distributions to represent uncertainty about key parameters in the computerised models, and/or 3) how to value the desirability of alternative outcomes. "Do not know" and "Disagree upon" can be further classified as types of uncertainty, where the former stems from the absolute unknown and the latter originates from potential partially unknown decisions. These types of uncertainty are more difficult to deal with as

they cannot be represented via probabilities and cannot be reduced by gathering more information.

Robust Decision Making[126] is among the most promising approaches for tackling Deep Uncertainty in decision- and policy-making. Robust Decision Making's premise includes:

1. Utilising or implementing an option that will perform adequately well in practically all conceivable scenarios. It often utilises Scenarios Planning approaches for handling underlying assumptions and different perspectives in the environment, elicits plausible scenarios, and models their possible outcomes. Additionally, Exploratory Modelling and Analysis (EMA) can make the scenarios under question concrete, comparable and narrow the search space.

2. Prepare to change an option when there is a change in the environment: The system goals must be defined, the "monitoring and correction" loop must be established, and options must be chosen for their adaptability in the unknown environment and designed to be changed when new information is available.

It is important to note that a robust option might not necessarily be the optimal option since an optimal option cannot be guaranteed to have the best performance beyond a set of circumstances, underlying assumptions, or a specific scenario.

4.2.2 Black Swan

Defined by Taleb [112], a Black Swan event is an unforeseen event that yields major consequences and can only be rationalised in hindsight. Such major consequences can both be positive and negative. In contrast to Black Swan, a White Swan event

Table 4.1: Mapping of Levels of Uncertainty from [66], Supplemented with Examples in TDM based on Our Observations

Uncertainty Level	Description	System Model	System outcomes	Examples in TDM
Level 1	A clear enough future, which one is not willing or able to measure the degree of uncertainty in any explicit way	A single system model	Point estimates with sensitivity	A single point estimates from most of COTS TD tools and mainstream TD identification/measurement approaches
Level 2	Alternate futures (with probabilities)	A single system model with a probabilistic parameterisation	Several sets of point estimates with confidence intervals, which probability attached to each set	Propagation of Error in TD measurement uncertainty [50]
Level 3	Alternate futures (with rankings)	Several system models, one of which is most likely	Several sets of point estimates, ranked according to their perceived likelihood	TD ranking based on TD prioritisation approaches
Level 4	A multiplicity of plausible futures	Several system models, with different structures	A known range of outcomes	Disagreement among practitioners during TDM on appropriate approach, value or utilisation
Level 5	Unknown future	Unknown system model	Unknown outcomes	Effectiveness of chosen TD identification, measurement or prioritisation approach(es)

is an event in which its major consequence is easily estimated, and Grey Swan is a highly probable event in which its impact can easily cascade. Errors from White and Grey Swan events shift focus on human-related errors or errors in judgment. According to Walker et al. (Table 4.1), Level 5 Uncertainty roots from *Recognised Ignorant*, resulting in Black Swan events[126]. Examples of Black Swan events are the 2008 Global financial crisis, the mega 9 Richter scale magnitude 2011 earthquake in Japan, and social disruption from an internet-based software company [126].

Black Swan events are susceptible in Extremistan data. Extremistan data is data with long-tailed and skewed distribution. This type of data is vulnerable to extreme and unpredictable values. An extreme value in this data can impact the interpretation/aggregation of the data population. Examples of Extremistan data include the market share of a specific product or the net worth of individuals.

Extremistan contrasts with Mediocristan. Mediocristan data is in line with a

normal distribution. Unlike Extremistan, extreme or unpredictable values do not impact the interpretation or do not impact the landscape of the data. Examples of Mediocristan data include the population's height or age.

Black Swan events can often trigger by one or several decision-making shortcuts. We refer to those as Black Swan Blindness, as these shortcuts can obscure the existence of Black Swan to a decision-maker. The description of Black Swan Blindness is as follows:

- **Confirmation Bias:** This is a situation where a decision-maker selectively utilises evidence or specific interpretation in supporting his decision. A decision produced with confirmation bias is vulnerable to a change outside of their evidence/interpretation. This issue can make the decision fragile to change in assumptions/conditions.

- **Silent Evidence:** This refers to a situation when a decision is chosen without considering evidence/interpretation that one fails to recognise. A decision without consideration of silent evidence can be impartial, rendering any change outside of the decision scope to be unidentified.

- **Narrative Fallacy:** This situation exists when a decision-maker utilises a narration as evidence/rationale for producing a decision. A decision based on narration can be inappropriate to the problem as it can be biased or based on distorted assumptions. Biased or distorted assumptions can alter one's perceptions, obscuring one in notice of change in the problem landscape.

- **Ludic Fallacy:** This refers to a situation when a decision-maker utilises overly simplistic assumptions in making a decision. Overly simplistic assumptions may not adequately consider the possibilities of the problem space, rendering one vulnerable to situations outside the utilised assumptions.

- **Tunnelling Bias:** In contrast to Ludic Fallacy, a decision with tunnelling bias refers to a decision under overly specific assumptions. A decision with overly specific assumptions can be sensitive to a change in the problem context. When a decision with tunnelling bias is utilised, one can be blinded from timely recognising a change in the problem context/the environment.

- **Epistemic Arrogance:** This refers to a situation when a decision-maker produces a decision based on generalisation, interpolation or extrapolation. These techniques may exclude certain aspects/details of the problem, making changes related to those to be untimely recognised.

4.2.3 MORDM

As we point out in the previous sub-sections, TD prioritisation can suffer from Deep Uncertainty due to potential disagreements among decision-makers, the incapability to resolve their differences and/or the incapability of producing a potential decision. Therefore, practitioners are faced with a colossal combination space of decisions to be taken. Additionally, decision-making processes in TD can promote the existence of catastrophic Black Swan events since these processes rely on Extremistan data, vulnerable to Black Swan blindness and fragile decisions. Due to the limited resource, practitioners are constrained to embrace the existence of Deep Uncertainty, and a systematic approach is essential to facilitate practitioners for mitigating Black Swan events, specifically, the one which can alleviate the search for a combination space of potential decisions.

One of the potential alleviations is Many Objective Robust Decision Making (MORDM) by Kasprzyk et al. [53]; this framework leverages the use of Many-Objective Evolutionary Algorithms (MOEAs) to find feasible robust solutions for a

problem in Environmental Engineering. This framework combines the strength of Many-Objective Evolutionary Algorithm optimisation with Robust Decision Making. The core of the four iterative steps of MORDM, which are driven by interactive visual analytics, are as follows:

1. Problem Formulation: In this step, practitioners define XLRM: Uncertainties of interest (marked as X), Levers of decisions, which are the parameters of decisions, can be expressed as real-valued, integer or binary, Relationship between them and Measures of a decision's performance.

2. Generating potential solutions via MOEAs: Pareto-front of potential solutions are generated via MOEAs in this step. Desirable MOEAs characteristics for this step are MOEAs that maintain both convergent and diverse search of the problem.

3. Uncertainty Analysis: In this step, alternative states of the world are raised to benchmark generated solutions. Practitioners select a solution that performs well across several states of the world into a set of robust solutions.

4. Scenario Discovery: Statistical cluster analysis on databases is utilised to identify uncertainty model inputs, later used to probe robust solutions for potential vulnerabilities. During this step, practitioners are informed of potential vulnerabilities in each solution in the set.

This framework leverages the use of MOEAs, interactive visualisation and statistical data analysis, emphasising managing conflicting perceptions of future changes in complex environmental systems, which involve deep uncertainty in future conditions. The above issue applies to TDM as decision-makers are tasked to envision the future impact of TD and associated mitigation routes. During TD identification and measurement, for instance, decision-makers with varying levels of competence can

have different trajectories on the interaction between system components and have differing focuses on the severity of TD. These differences support the presence of deep uncertainty in TD, manifesting in different TD models/representations, different sets of TD parameters, and/or different TD rankings during TDM decision-making. This identical objective of resolving deep uncertainty stemming from future trajectories makes the MODRM framework a prime candidate for adaptation in TDM decision-making. For TD prioritisation decision-making, we surmise that:

1. The use of MOEAs coupled with performance matrices of TD rankings (as objective functions) can eliminate a ranking of trivial TD and raise the importance of a TD ranking that includes significant TD. Therefore, MOEAs can potentially facilitate practitioners during TD prioritisation in narrowing the search space.

2. The interactive visualisation is widely used in several commercial off-the-shelf tools and state-of-the-art TD approaches. Thus, interactive visualisation, by nature, is compatible with facilitating practitioners during TD prioritisation.

3. The use of statistical data analysis can, however, hamper TD practitioners. This is because the location and severity of each TD are context- and practitioner-dependent. As there can be more than one TD in a system, the simulation of a specific TD can be cost-ineffective to establish. Furthermore, there is no consensus in the TD community for the universal and generalised TD model to represent its repayment cost and severity. This shortcoming directs practitioners to potential disagreement on the appropriate representation of TD and its measurement. Hence, these issues render statistical data analysis for TD prioritisation decision-making cost-intensive, if not entirely infeasible. Due to this necessity, a partial modification of the MORDM framework is essential for gaining the advantage of MOEAs and RDM, which we present in

Section 4.5.

4.3 Related Work

The application of MOEAs has been widely applied to solve several Software Engineering problems. Among the notable examples, Search-based software engineering[44], (SBSE) proposes reformulating software engineering problems as optimisation problems in order to seek a nearly optimal solution [95]. The Search-based software engineering repository features more than 1700 relevant publications on search-based software engineering from 1976 [138]. Although there is no study for alleviating Deep Uncertainty via MOEAs in TD, there are two studies that utilise MOEAs for relieving problems in the TD landscape: a study by Sri Harsha and Kari Systä [125] takes a multi-objective optimisation approach, specifically, genetic algorithms, to alleviate decision-making process where the focus is on trade-offs on short-term and long-term objectives (defined in the study). A study by Mohan, Greer and McMullan [83] demonstrates the effectiveness of metaheuristic search algorithms with metrics provided by an automated refactoring tool to reduce TD in code.

The concept of Deep Uncertainty is previously stated and mentioned in an interdisciplinary study related to computational science. Rigo [35] reports that Deep Uncertainty in large spatial-scale problems can prevent the usual statistical analysis of modelled quantities as well as suggest that semantic transparency and free software can be possible alleviations.

Scenario analysis can be found in several studies in Software Engineering. For example, ATAM (Architecture trade-off analysis method) [55] is a scenario-centric

approach used for evaluating architecture design decisions and choices. Scenario-based approaches have also been used in requirements engineering [111] for eliciting, concretising and validating requirements. Nevertheless, Deep Uncertainty has not been explored as part of existing software engineering efforts on scenario specification and/or analysis.

Nash equilibrium or game-theoretic-inspired approaches have been discussed in Software Engineering. Among several studies, Zeephongsekul and Chiera propose an optimal software release policy based on a game of timing and Nash Equilibrium of mixed strategies [137]. The theory of static games with complete information is used for producing a 2-player competitive software crowd-sourcing model [47]. Tong et al. present a Nash equilibrium-based game theoretical approach for producing effective strategies in software requirement change management, which facilitate green computing in Software Engineering [120]. [2] point out the application of Nash-equilibrium as a promising route in Social Software Engineering. For TDM, Madhivanan and Godandapani illustrate the application of epsilon Nash-equilibrium to seek an optimal strategy for alleviating a refactoring problem [77].

4.4 Deep Uncertainty and Black Swan in TDM Decision-making

4.4.1 Deep Uncertainty in TDM Decision-making

We define Deep Uncertainty in TDM decision-making as a situation when TD decision-makers cannot reach a consensus on TD decisions due to conflicts from different TD perspectives. Although Deep Uncertainty in TDM decision-making is

never explored prior, its definition bears a close resemblance to two definitions in community smells. The inability to achieve consensus on a TD decision is similar to Dissensus [90], a code-related community smell where team members cannot achieve a consensus on a solution. Dissensus is derived from social conflict in the organisational literature [18]. Conflicts from different TD perspectives are rooted in factions with different TDM responsibilities, TDM experience, TDM specialities, technical backgrounds, etc. In the context of community smells, this issue is considered as Solution Defiance [79]. Factions during TDM decision-making can stem one or more community smells, including physical, technical, social or cultural differences (e.g. Cognitive distance [116], Organisational skirmish [118]), differences in role (e.g. DevOps clash [116]), differences in development responsibility (e.g. Code Red [90], Lone Wolf [117, 27], Power Distance [116]), issues related to task coordination (e.g. Organisational Silo [118, 117], Prima Donnas [118], Obfuscated architecting [115]) or consequences from a previous decision (e.g. Dispersion [90]).

Since Deep Uncertainty in TDM decision-making relies on a conflict from different perspectives to TD decisions, it can manifest in any TDM activities, as long as TDM decision-making relates to that TDM activity can spark a potential disagreement. Table 4.2 illustrates the non-exhaustive potential disagreement that decision-makers might have up to TD prioritisation.

This table aims to illustrate that Deep Uncertainty can result in a conflict on a TD issue when managing TD along one or more of its management phases. Deep Uncertainty in TDM decision-making can inevitably exist, regardless of the system context (e.g. software architecture, type of TD, competency levels of decision-makers, etc.). Issues in Table 4.2 are categorised based on the following TDM sub-activities:

- Approach Selection: This sub-activity is related to the selection of TD ap-

proaches, their appropriateness and compatibility to the type of debts to be analysed, the different approaches that need to be employed etc.

- Approach Configuration: This sub-activity is related to the application of the selected approaches for identification, measurement, prioritisation and management of TD, including the configuration of the weighting system(s).
- Approach Utilisation: This sub-activity is related to human aspects and resource facilitation of selected approaches.

Deep Uncertainty in TDM decision-making contributes to TDM indecision, which can be resource-intensive, if not infeasible, to rationally resolve. Disagreement on a suitable TD ranking, for example, can be mitigated when all conflicting parties know the precise outcomes of decisions. This knowledge can be expensive to acquire before making a final decision (e.g. via conducting research, simulation, experiment, sandboxing, etc.).

As Deep Uncertainty in TDM decision-making has not been previously explored, decision-makers do not recognise Deep Uncertainty and utilise conflict resolution shortcuts such as voting, suppressing the differences, consulting experts, etc., to resolve any conflict during TDM decision-making. Though these shortcuts can produce a decision with fewer project constraints to resolve Deep Uncertainty, there is a lack of quality assessment of imposing decisions, and a finalised decision only depends on the voting result, decision-makers' preference, experts' competency, etc. Without consideration of different opinions, the finalised decision can be myopic, biased, or restricted to a specific assumption, condition or scenario.

Decision-makers can unknowingly introduce Black Swan Blindness when resolving TDM Deep Uncertainty with conflict resolution shortcuts. Black Swan Blindness is the list of Black Swan-induced symptoms, illustrated in Table 4.6. Blindness oc-

Table 4.2: Non-Exhaustive Potential Disagreement in Selected TDM Activities which can lead to Deep Uncertainty, Complemented by Examples from Our Case Studies

Sub-activities	Description	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Approaches Selection	Disagreement centres around the choice of TD approaches, its compatibility with TD types, any complement/different approaches that need to be employed, etc.	As the discovered TD from Accumulo and Cocoon are related to coding-related problems within a specific file, the decision-maker might object to using this as the sole source of TD and suggest alternative methods for identifying TD due to isolated or combined issues in various files (such as architectural smells, test coverage, vulnerabilities, etc.).	Under the existing method for measuring TD, it is assumed that all instances of leftover variables necessitate an equivalent amount of effort to resolve. However, some developers might contend that this approach is not sufficient for making a TD decision and demand a manual estimation of TD parameters (such as planning-poker or the Delphi method) as a substitute or supplementary option.	As COTS static code analysis tools assess factors related to code quality, such as maintainability rating, security rating, and reliability rating, for each file in addition to TD backlog, decision-makers might have differences of opinion regarding whether to incorporate the entire set or a portion of these parameters in TD prioritisation.
Approaches Configuration	Disagreement on the application of selected TD approaches and their settings.	The predetermined number of identification criteria for Cyclomatic Complexity (CC) is fixed throughout the project (10 for Cocoon and 200 for Accumulo). However, a developer might challenge this criteria and argue that it is either too few or too many, prompting them to suggest a refinement in the criteria.	According to the current estimation approach for TD, issues arising from lengthy methods, anti-singleton, and complex classes are assigned an equivalent amount of fixing effort in Accumulo. A decision-maker might take issue with this methodology and demand the TD parameter measurement profile be adjusted before using its outcome to make a decision.	The current approach to measuring TD considers issues related to low maintainability and low security ratings as equally significant. A decision-maker might contest this methodology if they consider maintainability to be either less or more essential than project security.
Approaches Utilisation	Disagreement involves over the necessary resources for implementing the chosen approaches, including roles of stakeholders, the assignment of developers, etc.	An expert could question the existing TD identification criteria from code duplication since this incident might come about due to library/code reuses, templates, or other similar scenarios	Given the wide range of skill levels among developers, some individuals might feel that the assigned TD necessitates either less or more effort than what has been estimated.	A security specialist might contest the TD prioritisation process if it disregards the security implications associated with certain TD, such as hard-coded addresses or the usage of deprecated libraries.

curs when one suffers from community smells or intentionally takes a shortcut in decision-making. We point out that conflict resolution shortcuts can manifest into Black Swan Blindness in TDM decision-making. In the next section, we discuss the feasibility of Black Swan events in TDM, based on Extremistan-driven data and the unpredictability of TD problems combined with Black Swan Blindness.

4.4.2 Black Swan in TDM Decision-making

We point out that TDM decision-making is an activity that deals with the unpredictable nature of the TD landscape and is driven by Extremistan data, resulting in TDM decisions that are sensitive/fragile to changes in the problem context or the system conditions. These decision drawbacks can be worsened when decision-makers utilise conflict resolution shortcuts, contributing to the manifestation of Black Swan Blindness in a final decision. These combining factors (are summarised in Table 4.3 and are detailed below) inevitably introduce a Black Swan event in TDM.

Table 4.3: Elements of Black Swan Observed from [112] with Their Relevancy to TDM decision-making, Illustrated by Examples in general TDM decision-making and Observation/Analysis Of Our Case Studies

Elements for Black Swan Event	Description	Relevancy to TDM Decision-making	Examples in TDM decision-making	Examples from our Case Studies (Apache Accumulo and Cocoon)
Extremistan Data-Driven	A Black Swan event could occur in data that follows a heavy-tailed distribution, where the distribution decreases more slowly than that of a typical distribution like the normal distribution. The statistical phenomenon known as a heavy-tailed distribution can be further categorised into two types: long-tailed and fat-tailed distributions. Due to the absence of a uniform standard for distinguishing between these categories, we collectively refer to both of them as heavy tails. (More detail in Table 4.4 and 4.5).	TDM decision-making relies on the TD backlog/data, which can be long-tailed or fat-tailed.	The distribution of efforts required for resolving TD can be heavy-tailed when the effort required is not bound by an upper limit, and the assessment can be highly subjective due to the technical aspects involved in addressing TD (such as the size of the affected component, the degree of difficulty in treating it, library/component reuses, etc.), as well as the expertise and time estimates of the developer.	The leptokurtic nature of the frequency distribution of TD efforts in Accumulo (28.90611) and Cocoon (3.81942) implies that the distribution has a wider, flatter shape with fatter tails than the normal distribution, suggesting the presence of Extremistan data. This serves as an indicator of the likelihood of Black Swan events in these datasets, thereby alerting TD decision-makers to exercise caution while assessing the root causes of Extremistan data and/or being careful when utilising their assessment of the TD and its repayment.
Black Swan blindness	A Black Swan event could occur when utilising one or several decision-making shortcuts that may lead to overlooking unusual behaviour. (More detail in Tables 4.6 and 4.7)	TDM decision-makers might opt for one or several decision-making shortcuts due to factors such as time constraints and resource limitations, potentially preventing them from recognizing the presence of a potential Black Swan event.	Black Swan blindness can occur when the TD decision maker may choose one or multiple TD identification and repayment strategies that might not be necessarily conclusive. As an example, the decision may only assess repayment of the TD through simple patches or tactics instead of considering the value-added of refactoring or re-writing because of a lack of knowledge, time pressure or resource constraints. The same could apply if the decision-makers make a TD repayment decision informed by only one source of TD analysis, where multiple techniques can be available (e.g. reliance on one tool to identify TD security issues instead of multiple tools, when available for more conclusive analysis for security TD).	One of the TD identification criteria used in both Accumulo and Cocoon involves Cyclomatic Complexity (CC) threshold. If this criterion is applied incorrectly, it can result in an overly simplistic approach (known as Ludic Fallacy), limit itself to specific types of TD (i.e. Tunnelling Bias), and/or reflect the beliefs or claims of a particular stakeholder group (i.e. Confirmation Bias). As an example, the reliance on CC to analyse TD may not be conclusive to the TD issues. CC is heavily focused on control-related issues (e.g. testing independent paths affected by these controls), yet it may miss data or semantics-related issues and their criticality along these paths (i.e. the use of CC may blind the decision-makers in tracking the other TD issues along these paths. The number of test cases that are required to test a specific path to test and debug the TD may exceed the number of ones suggested by CC.

Table 4.3: Elements of Black Swan Observed from [112] with Their Relevancy to TDM decision-making, Illustrated by Examples in general TDM decision-making and Observation/Analysis Of Our Case Studies (Continued)

Elements for Black Swan Event	Description	Relevancy to TDM Decision-making	Examples in TDM decision-making	Examples from our Case Studies (Apache Accumulo and Cocoon)
Sensitive/Fragile Strategies	A Black Swan event can occur when a fragile or sensitive decision is exercised.	A TDM decision can be sensitive to a specific assumption, time windows, environment, practitioners' inputs, etc. Without adequate consideration, a TDM decision can be challenged, falsified or become infeasible to exercise when one of these factors/inputs is changed.	A TDM decision that relies on an effort estimation derived from COTS (Commercial Off-the-Shelf) codes could become impractical if the developers utilise a different TD identification or measurement profile.	TD in Accumulo and Cocoon are identified/measured by a COTS static code analysis. As the method is limited to code-based TD, associated decisions can be fragile as COTS issues may go beyond the code to cover interface mismatch, incompatibilities of integration that can be difficult to solely identify on the code-level, and may challenge the validity of TD identification and measurement. The identification, admission, or repayment of TD in Accumulo and Cocoon could be highly sensitive to self-admission of TD. As Cyclomatic Complexity is employed for TD analysis in both projects, this approach can be fragile because additional techniques for TD analysis might still be required. Moreover, the analysis alone may not be conclusive due to the fact that control-based assessment and analysis is not exhaustive.

Extremistan Driven-data

Table 4.4 illustrates the descriptions of Extremistan-driven and Mediacristan-driven data and characteristics of TDM data in TD identification, TD measurement and TD prioritisation. The mapping and characteristics are further discussed in (Table 4.5).

TDM decision-making is often influenced by Extremistan data/input. Specifically, TD decision-makers normally utilise TD extrema to support their decision-making for technical debt prioritisation, resolution and repayment. In TDM, extrema could mean that analysts often use the highest reward (e.g., focusing on the highest severity) with the lowest effort when resolving TD. This observation is confirmed by several studies by the technical debt community, though not discussed using Taleb's term. For notable examples, a cost-benefit TD prioritisation by Zazworka, Seaman and Shull [136] focuses on TD with a high negative impact and low repayment efforts. Schmid [101] presents a TD prioritisation approach which considers the evolution and refactoring cost for repaying TD. The approach aims to mitigate TD with the highest positive cost-saving values. Seaman et al. demonstrate the application of the Analytic Hierarchy Process (AHP) to build a criteria hierarchy, its weights, and scales [104]. This approach aims to mitigate TD items with the highest AHP scores. Avgeriou et al. investigate TD indices featured in several commercial-of-the-shelf TD tools [13]. From the study, most of those tools utilise the skewness or the density of their TD index and population, where the index calculates the severity of the debt in the TD item population under investigation in the TD backlog.

Unlike data from physical science (e.g. age, weight, height, the density of population), TD data and its upper and lower bounds are context-dependent and unpredictable beforehand. The system complexity, for example, can vary with different

systems and is unknown without its analysis, where TD extrema can be present and can affect TDM. Apart from the unpredictability in Extremistan TD data, decision-makers can encounter the unpredictability attributed to different areas of TD problems.

Table 4.4: Distinction of Extremistan and Mediocristan data Observed from [112], with Examples from Our Case Studies in Selected TDM activities

Characteristics of Data	Description	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Extremistan	Data with heavy-tailed distributions may be susceptible to the influence of outliers, which can significantly alter the interpretation of the data and potentially impact decision-making. Note that a positive outlier mentioned in the examples is a mathematical term that indicates a member with a value that is higher than the average.	The frequency distribution of TD per component in Accumulo and Cocoon exhibits signs of heavy-tailed distribution, where a file component with a very high number of TD stands as a positive outlier of the TD backlog. This phenomenon can create cognitive pressure on TD analysts and decision-makers, as it may divert their focus and attention to this particular area while potentially neglecting other pending, deserving, or critical areas that require intervention.	The frequency distribution of TD effort in both Accumulo and Cocoon exhibits heavy tails, with positive outliers (such as TD requiring high repayment efforts) indicating an elevated need of resources to mitigate TD in the backlog.	The frequency distribution of cost in generated TD rankings for Accumulo is characterised by heavy tails, which indicates that positive outliers (such as a TD ranking with the lowest cost) could reveal ideal performance across all rankings. As the generation of TD rankings via Many Objective Optimisation yields multiple rankings, decision-makers can leverage the presence of positive outliers to benchmark different scenarios and strive for a "non-fragile" set of rankings that are resilient against potential disruptions or vulnerabilities (as demonstrated in Section 5).
Mediocristan	A data with thin-tailed distribution, where an outlier of the data does not influence the overall interpretation. (i.e. decision-makers can't decide by sole reliance on the thin-tailed distribution and more information is needed)	The severity of TD in both Accumulo and Cocoon generally follows a normal distribution, with all identified TDs classified within a severity range between 0.1 and 1. While an outlier (such as a TD with a severity level of 0.1 or 1) could potentially indicate significant issues, it should be noted that the severity level alone does not necessarily signify the criticality of the TD. TD with high severity levels may not always be crucial if the repayment effort required is small. Decision makers need more information beyond the thin-tailed distribution to determine whether a specific TD requires mitigation and what degree of repayment it can yield.	The maintainability rating of Accumulo and Cocoon projects is limited between 0.1 and 1, and any outlier (such as a TD with a rating of 1 or 0.1) indicates that TD has manifested in either a highly-maintained or lowly-maintained project, respectively. However, this information alone does not offer sufficient insights into the quality of the TD backlog. Extra effort for measuring the TD may be required for decision-making.	The number of TD present in generated TD rankings is limited by the total number of TD in the backlog. A TD ranking with a particularly high number of TD (i.e., a positive outlier) indicates that this particular ranking has a comprehensive coverage of TD. However, it should be noted that while the coverage of a TD ranking is an important factor to consider, it alone does not provide sufficient information for TD analysts to make informed judgments or comparisons between different rankings. For instance, a TD ranking with high coverage may reveal only partial repayments of various TD, leaving significant TD unresolved. Consequently, the outlier (such as the lowest/highest coverage of one TD ranking) does not impact the interpretation of all generated TD rankings.

Table 4.5: Characteristics of Extremistan Data Observed from [112] with Examples from Our Case Studies in Selected TDM Activities

Characteristics	Description	Example in general (i.e. the distribution of population's net worth)	Example in TD identification (i.e. the frequency distribution of TD per component)	Example in TD measurement (i.e. The frequency distribution of TD efforts)	Examples in TD prioritisation (i.e. The frequency distribution of cost in generated TD rankings)
Heavy-tailed distribution	The data is shaped in the heavy-tailed distribution (either fat-tail or long-tail distribution - see Table 4.3).	The distribution of the population's net worth is heavy-tailed as outliers (i.e. those with the highest net worth) are rare and far apart from the average.	Leptokurtic in the frequency distribution of TD per component in Accumulo (74.72583) and Cocoon (10.84639), both are heavy-tailed, indicating that the frequency distribution of TD per component in both two projects has wider and flatter shapes with fatter tails than the normal distribution. The highest outlier (e.g. the highest number of TD in a component) for Accumulo and Cocoon are 80 and 15. They are roughly 23 times and 6 higher than the mean in respective projects (i.e. 3 for Accumulo, 2 for Cocoon). This phenomenon suggests that the range of outcomes from these heavy-tailed distributions is more diverse when compared to the range of outcomes from normal distributions. Decision-makers should consider this diversity in identifying issues, as it may require different expertise or measurement techniques.	Leptokurtic in distribution of TD efforts for Accumulo is 28.90611, while that for Cocoon is 3.81942. The highest outliers for Accumulo and Cocoon are 740 and 190, respectively, which are approximately 25 and 7 times higher than their respective averages (30 for Accumulo and 27 for Cocoon). This phenomenon can prompt decision-makers to consider diverse TD measurement techniques.	The frequency distribution of cost in generated TD rankings for Accumulo is Leptokurtic (943.455). The highest outlier (e.g. the most frequent cost value) is 44, which is 44 times higher than the mean (1) of the distribution. This phenomenon suggests difficulties in ranking diverse issues (e.g., coding, maintainability, performance, security and/or reliability related among the many others), making prioritisation not straightforward in the absence of additional information.
Hard to predict	The outliers are hard to predict as they do not restrict to the past data.	The past data alone is insufficient to predict the future outliers of the population's net worth.	The occurrence of TD in a specific component cannot be accurately predicted based solely on the frequency of TD in previous commits of Accumulo and Cocoon. The density of TD in a new component may depend on factors such as its size, complexity, responsible developers' experience, evolving identification criteria, or other relevant variables that may differ from those of earlier commits. This can be particularly true if the system undergoes significant changes that make certain data obsolete, irrelevant, or inconclusive, requiring additional information and data debugging.	The effort required to address TD in a specific component may be challenging to forecast based solely on the effort spent addressing earlier commits of Accumulo and Cocoon. Newer versions could involve different types of TD, varying complexity or size, and potentially necessitate modifications in TD measurement profiles/configuration, skills, and competencies that were not required for previous commits.	Data from generated TD rankings alone cannot accurately forecast the cost associated with addressing them. The cost depends on several variables such as the number of TD in the backlog, repayment strategies employed, utilised prioritisation approaches, or other relevant factors that could change in the future. As a result, relying solely on past data to forecast the cost of addressing new TDs may not be accurate.

Table 4.5 Characteristics of Extremistan Data Observed from [112] with Examples from Our Case Studies in Selected TDM Activities (Continued)

Characteristics	Description	Example in general (i.e. the distribution of population's net worth)	Example in TD identification (i.e. the frequency distribution of TD per component)	Example in TD measurement (i.e. The frequency distribution of TD efforts)	Examples in TD prioritisation (i.e. The frequency distribution of cost in generated TD rankings)
Winner takes all	The outliers can shape the outlook of the population.	Outliers can indicate inequality of net worth in the population.	Outliers (i.e. components with the highest number of TD) can be interpreted as components/areas in the project which required the most attention.	Outliers (i.e. A set of TD with the highest TD refactoring effort) can indicate areas in the backlog that require the most mitigation efforts.	Outliers (i.e. TD rankings with the lowest cost) can indicate the ideal performance, which can be utilised to benchmark the cost for the rest of the distribution.
Severe consequences	The outliers carry severe/large consequences.	Outliers may contribute to the net worth of the population more than other members.	50% of TD in the backlog is resided in 19% of the top performers in Accumulo, whereas 50% of TD in the backlog in Cocoon is resided in 12% of the top performers. If TD in these top components is ill-identified, it can lead to a significant number of hidden TD, supporting the late detection in a set of severe TD.	For Accumulo, 80% of the total TD efforts in the backlog come from 6% of the top performers. For Cocoon, 80% of the total TD efforts in backlog is accounted for by 15% of the top performers. If TD efforts in these TD are underestimated, it can lead to a TDM crisis, where unexpected additional resources are required to mitigate the backlog. If this issue is left ignored/unnoticed, it can support the growth of severe TD.	The top 5% of all high-quality generated TD rankings come from the top 0.06% of TD rankings with the lowest cost. If TD rankings come from these top performers are improperly utilised/misidentified, ineffective TD rankings may be utilised, where TD rankings use the higher cost or/and severe TD is improperly treated.

Sensitive/Fragile strategies

Unpredictability is one of the natural challenges in decision-making. Unpredictability can be common in parameters and inputs informing TD calculation and decisions informed by these calculations [81, 4, 97, 89], the growth of TD interest [40, 41, 43], and its likely impact on the systems [19] and return of TD repayment, if any [65], taking into consideration various contexts (e.g., cloud-based services vs conventional system [6]), and requirements [1].

Decision-makers alleviate this challenge by producing TD decisions restricted to specific assumptions and/or certain conditions. Such restrictions, however, can render TD decisions sensitive/fragile to changes in the problem context or the system conditions. A TD repayment strategy and its action plans, for example, can only be applicable for a specific project, within the current sprint, or when considering TD security risks as the top priority.

Changes in the problem context or the system conditions can render commonly used TD techniques to become fragile for the new contexts (i.e., TD techniques can become ineffective, inadequate or inappropriate). Although these issues can be mitigated with a re-assessment of current decisions and periodic decision-making, they incur project resources. This shortcoming can motivate decision-makers to take decision-making shortcuts, potentially leading to Black Swan Blindness.

Black Swan Blindness

As in Taleb's argument about the presence of Black Swan Blindness and its symptoms in decision-making under Deep Uncertainty, we see that TDM decision-making is not different. Black Swan symptoms in TDM can be caused by decision-making

shortcuts that were exercised by engineers and developers to the TDM decision-making domain, as in taking a shortcut decision in choosing a TDM process that is not quite adequate or suitable for the context under question [79]. Table 4.6 illustrates decision-making shortcuts in TD identification, TD measurement and TD prioritisation, which can result in Black Swan Blindness.

The following is the list of Black Swan Blindness, their unintentional causes from community smells and their possible Process Debt. We also elaborate on how the limitation of conflict resolution shortcuts supports Black Swan Blindness admission.

- **Confirmation Bias** can exist when factions in the organisation employ an ineffective communication channel. An ineffective communication channel can condition social interactions and limit its effectiveness(e.g. Black Cloud [90, 118]), or when taking an inappropriate communication method (e.g. Informality excess [116]), the unwillingness of team members to adapt (e.g. Unlearning) or communicate (e.g. Radio silence or bottleneck [118, 117]).

  When different factions inadequately communicate, each faction can anchor towards evidence that supports their opinions.

  For TDM decision-making, Confirmation Bias can take the form of the rules of thumb, "good" practices or preference or bias to a certain TDM decision. Practitioners may unknowingly promote Confirmation Bias in TDM decision-making when they substitute formal documentation, communication or meeting plan for less resource-intensive means of communication (e.g. online/social platform, informal and small meetings, self-organisation of sub-teams. etc.)

  Since the objective of conflict resolution shortcuts is to find a consensus based on the vote results, practitioners' preference, and experts' competency, the resolution shortcuts lack an adequate assessment that can lead to Confirmation Bias in the final decision. When left unrecognised, Confirmation Bias can lead

to a myopic, inadequate or inappropriate outcome.

- **Silent Evidence** can occur when one makes a decision under inappropriate information management (e.g. Architecture by osmosis [115]), communication gaps (e.g. Bottleneck, Informality excess, Invisible architecting [115], Leftover techie [118], Radio silence), task coordination issues (e.g. Organisational silo), negative work environments (e.g. Lone Wolf, Lonesome architecting [115], Priggish members[116], Power distance, Prima donnas, Sharing villainy [118]), or when one make a decision which anchoring to past experience (e.g. Institutional isomorphism [116]).

  For TDM decision-making, examples of Silent Evidence include hidden TD, TD measurement inaccuracies or an unacknowledged dispute on a TDM decision. Process Debt from Silent Evidence can occur in TDM decision-making when practitioners do not probe for evidence relevant to decisions (e.g. due to a lack of project resources, deadline, etc.). As conflict resolution shortcuts do not provide a guideline on evidence examinations, evidence related to decisions or problem context can be inadequately investigated, probed or reviewed, possibly rendering suboptimal decisions to be exercised.

- **Narrative Fallacy** may be exercised in the developing environment with inappropriate information management (e.g. Informality excess, Invisible architecting), with the lack of conditions for effective communication or information sharing (e.g. Black Cloud, Sharing villainy), or in a development task with a limited number of developers (e.g. Code red).

  In TDM decision-making, Process Debt from Narrative Fallacy can exist when practitioners consider narrations as equally important as concrete evidence. This can happen when practitioners do not invest in or intentionally skip evidence examination (e.g. due to the confidence in imposing decisions, the unavailability of related project resources, etc.). Notable examples of a de-

cision with Narrative Fallacy include a TD decision based on experts' stories/opinions, academic literature or a decision based on the popularity or the reputation of a solution.

Conflict resolution shortcuts do not have a restriction on a method to help in seeking a consensus. Hence, practitioners can be vulnerable to Narrative Fallacy when supporting narrations of decisions are presented. A decision with Narrative Fallacy can be unverifiable as it may not be rationally selected. This issue can enable an ill-suited or insufficient decision to be finalised.

- **Ludic Fallacy** may be utilised in the organisation with sub-optimal collaboration in development tasks, either among team members (Lone wolf, Cognitive distance, Disengagement [116], Priggish members, Prima Donnas), task coordination issues (Code red, Leftover techie, Newbie free-riding [116], Organisational Silo, Organisational Skirmish), a previous conflict (e.g. Dispersion), or when team members adhere to the previous experience (e.g. Cookbook development [116], Institutional isomorphism)

  For TDM decision-making, Ludic Fallacy can be considered as Process Debt when practitioners see that more complex or realistic decisions are infeasible to utilise (e.g. due to lack of knowledge, inexperience, limited project resources). Notable examples include TD decisions which support the use of prototype tools or solutions or TD decisions that support the use of tools or solutions under simplistic assumptions.

  As conflict resolution shortcuts do not concern the quality of imposing solutions, decisions with overly-simplistic assumptions can be selected, and their limitations can be undocumented. These issues increase the risk of a final decision being misunderstood, over-exaggerated or utilised out of context.

- **Tunnelling Bias** in development environments, as an example, can occur when practitioners operate in an isolated manner and without an effective commu-

nication channel. This can be rooted in cooperation issues (e.g. Architecture Hood [118], Cognitive distance, Disengagement, Leftover techie, Organisational silo, Radio silence, Bottleneck), or sub-optimal information/knowledge management (e.g. Black cloud, Informality excess, Sharing villainy).

TD practitioners can intentionally admit Process Debt in a decision with Tunnelling Bias when they are unable to find other TD decisions which are applicable to their problem context. An example of a decision with Tunneling Bias can be a TD approach which applies in a specific context or is utilised under restrictive assumptions.

Decisions with Tunnelling Bias can be included and selected via conflict resolution shortcuts because those shortcuts do have prerequisites on the quality of imposing decisions. A decision with Tunnelling Bias can be resource-intensive to utilise as it is restricted to a specific assumption or condition. Additionally, a decision with Tunnel Bias can be difficult to use as it might require practitioners to have relevant skills or experience. When a decision with Tunnel Bias is misused, it can be incomplete, inapplicable or insufficient to mitigate the TD problem.

- **Epistemic Arrogance** can exist when practitioners anchor to their previous experience (e.g. Cookbook development, Institutional isomorphism), unwilling to collaborate due to organisational issues (e.g. Black Cloud, Architecture hood, Code red, Cognitive distance, Leftover techie, Obfuscated architecting, Sharing villainy), their personality (e.g. Disengagement, Lone wolf, Lonesome architecting, Priggish members, Prima donnas, Unlearning [116]), or a conflict from the past decision (e.g. Dispersion).

  For TDM decision-making, Epistemic Arrogance can be intentionally admitted as Process Debt when practitioners do not explore the other TD perspectives (e.g. due to inadequate knowledge, inexperience, unavailability of experts, the

deadline, etc.). Examples of a decision with Epistemic Arrogance include a TD decision based on a generalisation of findings in TD literature, a TD decision based on an extrapolation of the evaluation result, or an interpolation of the experience report.

Due to the lack of quality assessment in conflict resolution shortcuts, decisions with Epistemic Arrogance can be inadequately reviewed and selected. A final decision with Epistemic Arrogance may be an ineffective or insufficient decision as it is partially relevant or irrelevant to the problem context.

Table 4.6: Black Swan Blindness Observed from [114] with Examples in Selected TD Activities based on Our Case Studies

Black Swan Blindness	Description	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Confirmation Bias	The situation where a decision-maker utilises (biased) evidence conforming to her/his beliefs or claims in support of her/his decision.	Bias in the choice of metric, as an example. Consider where analysts use Cyclomatic Complexity as the sole metric and bias to the belief that it is the ideal code quality metric to inform the TD identification threshold.	Downplaying the actual TD effort, where a TD analyst/a tool reports TD measurement lower than the actual measurement to conform biased to her/his own judgement, experience, choice of metric, etc.	Bias in the choice of TD prioritisation can be based on analysts' experience, common practice and/or popularity. As an example, popularity can influence analysts to the choice of code-centric TD prioritisation with a focus on one TD type (i.e. code-level).
Silent Evidence	The situation where evidence related to a problem is left unconsidered during decision-making.	TD from the developer's turnovers may not be considered during TD identification and can be ignored. This may be due to the inexperience of TD analysts, lack of comprehension and/or visibility of the root causes of the issues, and likely implications of the issues, especially if the TD was admitted by predecessors.	The effort estimation of code-related TD does not include additional testing efforts of refactored code. This issue can be due to difficulties in estimation (e.g. hard-to-estimate testing efforts), the lack of a suitable approach (e.g. the lack of measurement that estimates testing efforts) and/or limited TDM constraints (e.g. manually estimating testing efforts can be cost-intensive).	Evidenced TD of other aspects (e.g. performance, reusability, etc.) and its impact are not considered when ranking TD due to the limited number of supporting approaches (e.g. there is no TD prioritisation approach which explicitly considers performance or reusability of the system), uncommon knowledge (e.g. the impact of TD in performance or reusability is understudied).
Narrative Fallacy	The situation where a biased narrative or a story is used as evidence/justification to inform decision-making.	An analyst focuses on identifying TD in the most crucial components based on developers' stories or fallacies. Developers' stories may be utilised as evidence due to the lack of experience, knowledge, resources, etc, to elicit the stories as software documentation. As an example, fallacy on performance strict requirements for a given system(e.g. 5 sec per operation) that are not grounded on empirical evidence.	Misconception or narrative fallacy of the TD effort required can be used to measure an issue. As an example, COTS TD may cross several issues and may require diverse measurement approaches. The absence of standardised and industrially accepted TD measurement for COTS may give narrative fallacy (e.g., focusing on interface mismatch as a popular TD issue and using misconfiguration/compatibility as measurement while ignoring other important dimensions - e.g., security, performance, etc.)	Focusing the prioritisation on a narrative fallacy. Clients' non-technical narration (e.g. profitability, criticality of the manifested components, etc.) are considered when ranking TD. The reliance on non-technical narration can be due to the lack of formal documentation, changes/deviations from existing concrete evidence, the lack of resources in eliciting the narration, etc.

Table 4.6: Black Swan Blindness Observed from [114] with Examples in Selected TD Activities based on Our Case Studies (Continued)

Black Swan Blindness	Description	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Ludic Fallacy	The situation where an unrealistic or mismatched assumption is utilised in making a TD decision.	TD identification approach is utilised out of its application context (e.g. Using developers' comments to test-related TD). This issue can be caused by the lack of an identification approach specific to the application context and/or the limited project resources to fine-tune the existing approach.	A COTS static code analysis tool estimates the significance of several TD issues (e.g. a long method, a complex class) equally regardless of their manifestation contexts (e.g. complexity of a manifested function/component, possible ripple effects, etc.). This issue can be due to simplified assumptions (e.g. each line of code can be roughly converted to an equal amount of refactoring efforts) and/or a simplified measurement method (e.g. anchoring to one software metric to estimate TD refactoring efforts).	A TD prioritisation approach is utilised outside of its intended scope (e.g. using a code-level software metrics-based prioritisation approach to prioritise test-related TD). This issue can be due to the absence of an approach that fits for the problem context (e.g. the lack of an approach specifically for test-related TD) and/or the lack of resources to individually utilise TD approaches that match their type or context.
Tunnelling Bias	The situation where an assumption, perspective or interpretation is narrowingly utilised to produce a decision.	Anchoring on using only one approach to identify TD. This can be due to common practice in the TD community, past experience and/or the lack of knowledge of using other means.	Estimates TD effort with a certain scenario in mind. This can be due to the lack of diverse experience (e.g. to make other trajectories), the lack of acknowledgement of other situations and/or the lack of project constraints to probe for different scenarios.	Prioritising TD under a restrictive assumption (e.g. TD parameters are precisely measured). This is due to the absence of an approach in which different assumption can apply (e.g. An approach which tolerates TD parametric distortion).
Epistemic Arrogance	The situation where a decision-maker overestimates one's own judgement when making a decision.	Intentionally utilising one TD identification approach to cut corners (e.g. meeting the project deadline, budget, etc.). This decision can be exercised when a decision-maker is confident that the identification approach is suitable to the project and/or any issues from the decision can be amended in the future, or can be negligible.	The effort of all TD from a hard-coded address is estimated/profiled equally by an expert regardless of manifested components. This decision can be realised if the expert believes that TD effort estimates will not vastly differ from their past encounter.	A TD ranking is utilised without consideration of hidden TD in the ranking. This decision can be used when decision-makers believe that TD is accurately identified/measured and/or negative consequences from the decision are negligible.

Collectively, we view TDM decision-making as a process that is complicated by outliers that can be present in TD data of the subject environment and the aggregated population (i.e. Extremistan Data). These factors render TD decisions to be naturally sensitive/fragile to a change in the problem context or the system conditions. The impact of these shortcomings can be intensified when decision-makers utilise conflict resolution shortcuts to resolve a TDM decision conflict (i.e. Deep Uncertainty) while minimising the use of project constraints. Collectively, these issues result in TDM decisions that are not only context-sensitive but also Black Swan-induced.

One of the fatal Black Swan events in TDM is an eruption of severe TD. As this Black Swan event requires a considerable amount of project resources to mitigate, it supports a development crisis. Worse, this Black Swan event may lead to early obsolescence of the manifested system, especially when the mitigation cost from this issue is higher than implementing a new system.

This Black Swan event is caused by a failure to recognise the growth or the manifestation of TD. Thus, it can apply to any system environment regardless of TD type. To illustrate, we use possible outcomes from conflict resolution shortcuts when they are utilised to resolve a potential TDM decision-making disagreement in Table 4.2. For TD identification, conflict resolution shortcuts can contribute to an ineffective TD identification approach, which can systematically misidentify or hide TD in the system. Conflict resolution shortcuts on TD measurement decisions can yield an inadequate TD measurement method, which can systematically undermine the impact of TD in the backlog. For TD prioritisation, conflict resolution shortcuts can leave imposing TD rankings insufficiently verified, contributing to ill-suited TD repayment strategies. This can render significant TD to be improperly mitigated.

As decision-makers may not acknowledge the drawbacks of conflict resolution

shortcuts, any of the above outcomes can be left undiscovered as long as corresponding TDM decisions are in effect, and decision-makers may start to re-assess the effectiveness of the decisions only when it is necessary to do so (e.g. due to stakeholders suspicious, customer complaints, decision-makers or project turnover, etc.). Hence, any of the above outcomes are aggravated before their discovery, leading to TD manifestation and accumulation.

In Section 4.5, we investigate a practical TDM Deep Uncertainty problem that is rooted in a TD decisions conflict and relies on Extremistan data. This problem is vulnerable to conflict resolution shortcuts, which can result in an eruption of severe TD. The aim of this case is to concretely illustrate that Deep Uncertainty can reside in the TDM of systems and TD types under consideration.

4.5 Deep Uncertainty in TD Rankings

In this section, we explore the problem of Deep Uncertainty affecting TD ranking and prioritisation. We use this problem to illustrate a practical situation, where TD decision-makers converge into discrepancy or disagreement in TD rankings.

To address this problem, we present a MORDM-based framework to alleviate Black Swan Blindness and Deep Uncertainty in TD ranking. The framework has the following four iterative steps:

1) Scenario Elicitation: Practitioners utilise the Formal Approach for Scenario Analysis to elicit possible TD ranking methods that are suited for the given problem, each supported by scenario(s) with the utility scores. The aim of this step is to capture possible TD ranking approaches, minimising Tunnelling Bias and Silent Evidence. Elicited scenarios and their utility scores feed as input into MOEAs

objective functions. The original MORDM follows a bottom-up approach to discover the system scenarios and uses statistical data analysis; instead, we leverage Formal Approach for Scenario Analysis, which is specialised for the software systems, to discover the system scenarios in a top-down manner.

2) Black Swan Blindness Mitigation: TD practitioners can then consult Black Swan Blindness mitigation guideline (Table 4.7) to mitigate Black Swan Blindness in captured TD rankings. In case Black Swan Blindness is admitted as Process Debt, it should be noted/documented for future mitigation. The aim of this step is to assess, sanitise and document any evidence, narration, or knowledge that is attached to the rankings.

3) Solution Generation: To promote the robustness of TD decisions, practitioners select solutions which perform well across different scenarios into a set of robust solutions. This step follows the original MORDM framework, where MOEAs with defined objective functions are utilised to generate feasible solutions. In terms of objective functions, we aim to generate TD rankings that maximise the sum of each scenario-specific utility score and minimise the sum of global TD cost of TD rankings.

4) Vulnerabilities Discovery: Practitioners utilise pay-off matrices of objective values to determine the strengths and weaknesses of solutions in the set. This can be done when two players are a pair of different objective functions. By leveraging the concept of Nash equilibrium, practitioners assess the strengths and weaknesses of each solution by looking at Nash equilibrium/equilibria respective pair-wise decisions. The aim of using Nash equilibrium is to have a non-biased, traceable, objective value-based visualisation for practitioners. To leverage the concept of optionality, solutions with diverse strengths and weaknesses are kept for exercising by practitioners when there are changes, which are documented and elicited as scenarios, or

undocumented, in the system.

Like the original framework, we promote the use of interactive visualisation, where practitioners consider appropriate. This includes the ones that are not utilised in each step and controlled experiments below.

4.5.1 Definition

During TD prioritisation decision-making, a decision-maker can impose TD ranking of TD items in the backlog. This TD ranking is one among all TD rankings in the solution space U. A TD ranking imposed by Decision-maker X is denoted by:

$$D_X \in U \tag{4.1}$$

X TD ranking includes n decisions, a non-negative number of decisions corresponding to the number of TD in the backlog.

$$D_X = \{d_1, d_2, \cdots, d_n\} \tag{4.2}$$

Due to the unpredictability in the TD problem landscape (illustrated in Section 4.4.2), each TD ranking is restricted to utilised assumptions or conditions, which are utilised in every decision in the ranking. For example, a COTS-based TD prioritisation tool [13] may primarily focus on TD in coding, which emphasises a specific set of TD quality attributes (e.g. maintainability, evolvability, security, etc.).

$$R \Rightarrow d_1, d_2, \cdots, d_n \tag{4.3}$$

Each decision comes with probability, given the level of confidence, should this decision be exercised. The probability of each decision is dictated by its achievement of performance indicators. TD ranking is influenced by Extremistan data (illustrated in Section 4.4.2), where decision-makers often utilise outliers to indicate the probability of them being mitigated.

$$d_i = \{P_i, Cn_i\} \tag{4.4}$$

Where P_i indicate the probability of the decision will be exercised within the given context (i.e. Cn_i). For example, in a COTS-based TD prioritisation approach, the probability for TD to be mitigated can be implied from its TD index and its principal, where the probability for TD with a high TD index and low principal to be mitigated is higher than TD with a high TD index and high maintenance effort and so on. TD index and TD principal are calculated differently in various COTS tools, where some of these focus on the combination of several software metrics, the number of violated functions, the number of issues, time to fix, etc. Thus, data of TD index and TD principal can be subject to different TDM environments. The maximum value, the minimum value of the TD index and the TD principial of one TD backlog can only be known when one firstly initialises a corresponding COTS TD analysis.

In case decision-makers collectively accept an imposing TD ranking, we consider as they impose the same TD ranking. For a team with m decision-makers, this can be denoted as:

$$D_1, D_2, D_3, \cdots, D_m = D_\alpha \tag{4.5}$$

and

$$D_\alpha = \{d_\beta\} \tag{4.6}$$

Where $|D_\alpha| = 1$, D_α and d_β are an ideal decision set and an ideal decision respectively.

When a conflict on a suitable TD ranking exists, it means there is no common TD ranking imposed by decision-makers:

$$D_1, D_2, D_3, \cdots, D_m = \varnothing \tag{4.7}$$

Deep Uncertainty in TD ranking can occur when a decision-maker utilises different assumptions/conditions, resulting in a different TD ranking being imposed. For example, Decision-maker X imposes the use of a COTS code-based TD prioritisation approach, which is different from Decision-maker Y, a security expert, who oversees the security impact of TD outside the scope of the above TD prioritisation approach (e.g. vulnerabilities in software architectures, database, etc.), and imposes the use of risk-based TD prioritisation [97].

Since the causes of Deep Uncertainty in TD ranking stem from different assumptions/conditions on TD rankings, it can involve more complicated cases such as a conflict from the different combinations of multiple TD prioritisation approaches. These cases can happen in practice as one may utilise a similar approach with a different emphasis on TD components or try to compromise/complement TD prioritisation in different approaches. For example, one may try to utilise multiple COTS TD prioritisation tools with emphasis on different TD types. Nevertheless, each combination of TD rankings ultimately yields different decision sets and/or TD probability to be mitigated.

Formally, Decision-maker Y imposes another TD ranking included in the solution space U:

$$D_Y \in U \tag{4.8}$$

Via a pair-wise comparison, we denote Deep Uncertainty in TD ranking between

TD rankings imposed by Decision-maker X and Decision-maker Y as:

$$D_X \neq D_Y \tag{4.9}$$

Where the degree of similarity between TD rankings imposed by Decision-maker X and Decision-maker Y can be measured with the Jaccard similarity index as:

$$J(D_X, D_Y) < 1 \tag{4.10}$$

$J(X, Y)$ is Jaccard similarity of X and Y and valued from 0 (i.e. no similarity) to 1 (i.e. total similarity). To mitigate a conflict in TD rankings, one must aim to increase the Jaccard similarity index between TD rankings imposed by different decision-makers:

$$\max_{i \in [m]} J(D_i, D_{i+1}) \tag{4.11}$$

Whereas $[m] = \{1, \cdots, m\}$ is a number of decision-maker(s) until m. Conflict resolution shortcuts (e.g. voting, experts' opinions, suppressing differences) can help decision-makers in collectively concluding one TD ranking (Equation 4.6). Due to the limitations of these shortcuts, however, they can be attributed to Black Swan blindness (Section 4.4.2).

4.5.2 Black Swan Blindness Mitigation

Table 4.7: Possible Black Swan Blindness Mitigation Observed from [112] with Examples in Selected TD Activities based on Our Case Studies

Black Swan Blindness	Possible Mitigation	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Confirmation Bias	Provide negative evidence for falsifying the claim.	To counter the reliance on a single TD identification method, one may point out the presence of TD identified by different identification methods. Different TD Identification methods can be observed from a third-party TD dataset, a backlog from other projects, an empirical study, etc.	In case one suspects that the TD parameter is over-/underestimated, one may request/investigate for its rationale or evidence of how it is estimated. As an example, one may investigate measurement profiles or consistency in TD measurement across different components.	To neutralise bias to a specific TD type during prioritisation, one may provide references that point out the potential impact of different TD types. For example, one may present a study/an industrial case where TD from an architectural level can lead to premature obsolescence of the manifested component.
Silent Evidence	Request and verify evidence that supports the claim.	Considering the diverse development aspects which can result in TD during the identification. For instance, one may investigate possible causes of TD in different TD types (e.g. code smells, admitted, data consistencies, etc.) before choosing the TD identification approaches.	Include and verify different refactoring scenarios and their impact when manually estimating TD effort. For example, one may conduct a scenario analysis to elicit a possible range of TD parameters, its consequences, and their additional efforts.	Investigate possible TD impact in several quality attribute aspects when prioritising them. As an example, one may consider the TD impact from security, profitability, and reusability. Then, weigh them conforming to the focus of the system when ranking TD.
Narrative Fallacy	Prefer experimentation or evidence over narration.	Prefer conducting TD identification based on concrete evidence/documentation over the narration. For instance, one may transform and document developers' narration into additional requirements and/or consider using identification criteria/approaches which are aligned with the said requirements.	Verify the TD measurement method, its configuration, and TD parameters with concrete evidence/reference. As an example, one may verify whether the existing TD measurement method can sufficiently represent the significance and/or severity of TD by observing its effectiveness in other projects, case studies, other measurement methods, etc.	Documenting customers' non-technical aspects into concrete evidence and cooperating them with suitable prioritisation approaches when ranking TD. For example, one may document clients' non-technical narration into additional requirements in software specification and/or introduce TD approaches in which they consider these requirements.

Table 4.7: Possible Black Swan Blindness Mitigation Observed from [112] with Examples in Selected TD Activities based on Our Case Studies (Continued)

Black Swan Blindness	Possible Mitigation	Examples in TD identification	Examples in TD measurement	Examples in TD prioritisation
Ludic Fallacy	Utilise a more appropriate approach(es) for modelling real-life situations.	Ensure that the underlying assumptions/application contexts of the approach are suitable to the environment. For instance, one may verify the limitations/assumptions of utilised identification approaches are not in contrast with the environment. In case there is a lack of an approach that is perfectly suitable to the environment, it must be noted to raise awareness for other decision-makers.	Verify the COTS measurement profile, and if it is necessary, re-configure the profile, ensuring that it is capable of differentiating TD from common issues. As an example, one may introduce additional software metrics (e.g. cohesion, coupling, etc.) in measuring TD from a long method and a complex class.	Explore the scope/limitation of the prioritisation approach and its impact on the project. If it is infeasible to amend, note this issue during decision-making. For example, one may explore using the combination of COTS, which emphasises different development levels to prioritise code-level and test-related TD. In case such a combination is infeasible to exercise, one may note the limitation for future amendments.
Tunnelling Bias	Employ an evidence-based approach.	Considering the use of diverse identification criteria for identifying TD. For instance, one may use a combination of different TD identification approaches when identifying TD (e.g. using a code-level, an architectural-level and requirement-based identification approaches to identify TD).	To alleviate overly restrictive TD parameters, promote the use of a formal method of scenario analysis to probe for scenarios and their range of TD parameters. As an example, one may utilise scenario analysis and/or planning poker to complement the existing TD measurement method.	Verify that the underlying assumptions of the selected prioritisation approach are applicable before use. For example, decision-makers may verify the measurement of TD in the backlog to ensure that they conform to the assumption that they are precisely measured.
Epistemic Arrogance	Probe for the limitation of the source, the evidence, and/or the claim.	Document the use of a decision-making shortcut for future remediation. For example, one may document the use of a single TD identification approach, possible causes and their remediations when it is applicable.	Aim to individually assess the impact of TD from a hard-coded address. Note the issue for decision-making and future amendments if it is currently infeasible. As an example, one may complement different TD measurement methods (e.g. software metrics, architectural-based, wide-band Delphi, etc.) to differentiate TD from hard-coded addresses in different components. If this is impossible, this should be noted for future alleviation.	Elicit potential risks, their mitigation, and/or alternative routes of the TD ranking before its utilisation. To illustrate, one may assess and document potential risks from selected TD ranking (e.g. hidden TD, inaccuracies TD measurements, etc.), their possibilities (best-worst cases), and their mitigations before using the ranking. Additionally, the documented risk log should be periodically monitored and updated to conform with the current situation.

Black Swan Blindness can be unavoidable when one takes a decision-making shortcut, either intentionally or unintentionally (as described in Section 4.4.2). To mitigate this, we illustrate the non-exhaustive list of Black Swan Blindness in selected TD activities, including TD identification, measurement and prioritisation (Table 4.6) and their potential mitigation (Table 4.7). By doing so, practitioners can note potential Black Swan Blindness and plan for its mitigation. Black Swan Blindness can be managed akin to risk management since the impact of each blindness can be dynamic, and its mitigation can require different degrees of project resources. Depending on the location of Black Swan Blindness, practitioners can re-initiate this framework or continue the next step of this framework once Black Swan Blindness is noted and/or mitigated.

4.5.3 Scenario Elicitation

Assumptions and conditions utilised, when a decision-maker imposes a TD ranking can be inadequately understood or unknown to others. This is because different decision-makers can have different levels of competency and different interpretations of the problem. Without any mitigation to this issue, the imposed TD ranking may suffer Black Swan Blindness as the decision is inadequately assessed. As an attempt to alleviate the presence of this phenomenon, we propose the use of scenarios to cluster a set of assumptions and conditions. By using scenarios, a decision-maker can make example(s) from the said situation, informed by their assumptions/conditions, and communicate the example(s) to other decision-makers in a more concrete manner. On the other hand, other decision-makers can verify whether the given example is feasible and whether grounded assumptions/conditions are justified or suitable for rendering a related TD ranking to be a candidate solution. For example, Decision-maker Y may communicate his opinions through a set of system interactions and

vulnerable attack vectors, conveying to other decision-makers the necessity to consider a security risk-oriented TD prioritisation. Formally, we substitute a set of the unbound set of assumptions/conditions R (Equation 4.3) with a scenario set related to a given context S:

$$S \Rightarrow d_1, d_2, \cdots, d_n \tag{4.12}$$

And

$$S = \{s_1, s_2, \cdots, s_o\} \tag{4.13}$$

Where o is a positive integer. Each scenario represents a situation/environment the system will encounter, encapsulating the group of related assumptions/conditions, which is used consistently throughout the TD backlog:

$$s_i = \{TD_1, TD_2, \cdots, TD_p\} \tag{4.14}$$

Where S_i is ith Scenario and TD_p is $p$the Technical Debt. Decision-makers can verify scenarios and exclude underlying assumptions/conditions in a bundle or exclude an imposing TD ranking if the assumptions seem to be infeasible. From the above example, other decision-makers may falsify a TD ranking imposed by Decision-maker Y if they find that the given scenario is unrealistic. As a scenario is a product of assumptions/conditions, the number of members in S can be less than R. TD rankings, derived from a scenario set with a smaller number of members, can contribute to a more similar foundation, yielding more similar TD rankings sets between different decision-makers.

We leverage the Formal Scenario Analysis technique [46] to elicit run-time scenarios. This top-down approach is intuitive, human-centric, "standalone", and it can still be applied in the absence of historical data when analysing scenarios of software systems. More importantly, by leveraging a top-down approach, system scenarios can be established at the early phase of RDM, as well as it is more efficient than

using statistical data analysis from the bottom-up. Thus, it is possible to apply scenario-specific objective functions to search for robust decisions via MOEAs in the next step.

The objective of scenarios is to help decision-makers in assessing the quality of imposing TD rankings. Scenarios may not resolve a conflict in TD ranking, as decision-makers might employ scenarios which are feasible to the system. From the above example, a TD ranking by Decision-makers Y can be included in the list of candidates, if a given scenario is considered feasible and an imposing TD ranking is considered justified by other decision-makers. However, elicited scenarios can be used as a mechanism for resolving such conflicts in the next step/iteration.

4.5.4 Solutions Generation

In this step, we aim to maximise the robustness and the similarity of TD rankings by selecting TD rankings generated with MOEAs. By doing so, a set of selected TD rankings are said to be resistant to changes and have a similar level of quality and cost-effectiveness. As these sets of TD rankings cover different assumptions/conditions, they can satisfy different decision-makers, potentially alleviating conflicts in TD rankings between them.

The robustness can be observed from the utility score for different scenarios. A robust TD ranking refers to a TD ranking which performs well across different scenarios. For example, a TD ranking, which is the combination of different TD prioritisation approaches, can be considered a robust TD ranking as it considers different TD impacts, types of TD, different TD metrics, etc. Thus, it is more resistant to changes in assumptions/conditions. The scenario-specific utility score

can be calculated by:

$$U(s_i) = \sum_{TD_j \in s_i} U(TD_j) \tag{4.15}$$

Whereas $U(TD_j)$ is the utility score of jth Technical Debt. Based on the concept of Antifragility[113], practitioners can utilise the sum of utility score in each scenario as "stressors" to compare the performance of each TD ranking before making a decision. This applies until there is a change in the set of scenarios and/or the utility score calculation. If such change exists, the new set of scenarios and/or the utility score calculations are then acting as new stressors for the next iteration. The utility score of TD is subject to TD prioritisation approach(es) selected and/or transformed by decision-makers:

$$U(TD_j) = f(TD_j) \tag{4.16}$$

Where TD_j is jth Technical Debt and $f(TD_j)$ is a utility function of ith Technical Debt. Depending on the selection of TD prioritisation approach(es), it can be directly translated (e.g. cost-benefit-based) or transformable (e.g. SQALE-rating-based) to its utility score.

The objective function of scenario-specific utility scores, for ith scenario, it can be denoted by:

$$\max U(d, s_i) = \sum_{TD'_j \in s_i} U(TD'_j) \tag{4.17}$$

Until oth scenario

$$\max U(d, s_o) = \sum_{TD'_j \in s_o}^{k} U(TD'_j) \tag{4.18}$$

Where $U(TD'_j)$ is the utility score of jth TD in each decision set until k. TD rankings that are similar on the sets of scenarios and the level of budget. For example, TD rankings based on COTS tools can be similar to each other as they cover the same set of assumptions/conditions (e.g. code-based TD, maintainability of TD, code-based TD principal, etc.). Thus, they may have a similar assessment of TD issues

with a general agreement on the level of investment to resolve these issues. The budget objective function is denoted as:

$$\max B(d) = B - \sum_{i=1}^{k} C(TD'_i) \tag{4.19}$$

Where B is a project budget for TD prioritisation, TD' is ith TD included in each decision set until k. The cost for ith Technical Debt is denoted by $C(TD_i)$.

Though any MOEAs are applicable in this step, it is recommended to utilise MOEAs that aim to generate diverse decisions since they can be more distinguishable by decision-makers, who must individually acknowledge their strengths and weaknesses in the next step. We recommend building a set of TD rankings which are robust and similar to each other. That is, TD rankings which perform well across different scenarios and use the same level of budget. This conforms to a concept of optionality in Antifragility[113]. For the optionality, decisions are included in a decision set rather than a single decision. All decisions are archived in our framework, and decision-makers can exercise selected decisions subject to the system's conditions/environmental changes. When changes in documented conditionals/environmental (i.e. scenarios) occur, decision-makers may alter the current decision set to the one which yields the most utility score. If undocumented conditionals/environmental changes rise, decision-makers can consult archived decisions in the set by comparing undocumented changes with the closest existing scenarios. Then, decision-makers may choose whether to exercise the most suitable decision set for undocumented changes or re-initiate the first step of the framework.

In the next step, we present a visualisation technique which helps decision-makers in acknowledging strengths and weaknesses in TD rankings.

4.5.5 Vulnerabilities Discovery

This step aims to guide the decision-makers, by evaluating the strengths and weaknesses of generated Pareto-optimal decision sets to be chosen. We utilise pay-off matrices to illustrate strengths and weaknesses, driven by Nash Equilibria. Nash Equilibria is chosen due to the neutrality towards all decision-makers as well as the guaranteed existence of Nash Equilibria in every finite game. Thus, the strengths and weaknesses of different TD rankings are always presented when visualising pay-off matrices.

Practically, decision-makers plot pair-wise pay-off matrices where different objective functions act as players and each solution is treated as an option. Utility scores of different objective functions and different decision sets then can be input accordingly. Then, decision-makers observe utility scores of different pairs in order to find strengths (i.e. their Nash Equilibrium/Equilibria) and weaknesses (i.e. pairs with lowest utility scores) towards different scenarios.

More precisely, we consider a selection of TD rankings as a cooperative game. Formally, let D_l^{rank} be the set of TD rankings based on lth objective function, and $l = 1, \cdots, o+1$ (i.e. the total number of objective functions). For a ranking decision d^{rank}, $d\prime^{rank} = (d\prime_l^{rank}, d\prime_{-l}^{rank})$ is a strategy profile, where $d\prime_l^{rank}$ is a ranking decision based on an objective function l and $d\prime_{-l}^{rank}$ are ranking decisions based on all objective function except l. If $u_l(d_l^{rank}, d\prime_{-l}^{rank})$ is a payoff as a utility function from objective function l, $d\prime_l^{rank}$ is a (weak) Nash Equilibrium if:

$$u_l(d\prime_l^{rank}, d\prime_{-l}^{rank}) \geq u_l(d_l^{rank}, d\prime_{-l}^{rank}) \tag{4.20}$$

For all $d_l^{rank} \in D_l^{rank}$. Since there can be more than one weak Nash Equilibrium in

one game, those Equilibria are included and increases the probability of corresponding decision to be utilised (i.e. $P_i^{rank} \in d_i^{rank}$). We relax the restriction of Equation 4.6 from $|D_\alpha|= 1$ to be $|D_\alpha^{rank}|\geq 1$ whereas $D_\alpha^{rank} = \{d_1^{rank}, d_2^{rank}, \cdots, d_q^{rank}\}$. q is the total number of TD rankings in the set selected by practitioners.

4.6 Case Study A: Apache Accumulo

We present two case studies to explore the effectiveness of our approach. These two case studies differ in the number of identified TD and the system size. Those studies support answering research questions on the capability of our approach to different degrees of diversity in identified TD. Supporting scripts and mining queries for those studies can be found at (https://github.com/DUTDPr/DUTDPr-repo). We utilise scenarios elicited by Formal Scenario Analysis [46] to illustrate how to manage possible conflicts in TD rankings in those studies. We take a control experiment approach in assuming conflict presences, as these studies do not come with a TD ranking list. Henceforth, one should not imply the lack of evidence of a conflict in TD rankings to no conflict. A conflict in TD ranking may be informally acknowledged, resolved or ignored by decision-makers, rendering them to be silent evidence.

The first case study is Apache Accumulo, where the identified TD is obtained from The Technical Debt Dataset, Version 1.01 by Lenarduzzi et al. [68]. Since the relevant dataset version considers all developmental branches for this project, it might not be regarded as a faithful representation of TD in the real-world project. However, the premise is that by looking at different branches, each with varying development responsibilities and resource constraints, the likelihood of identifying TD can be increased. Hence, the diversity of identified TD of all developmental branches can be higher than in a project with only one branch inviting the need for analysis

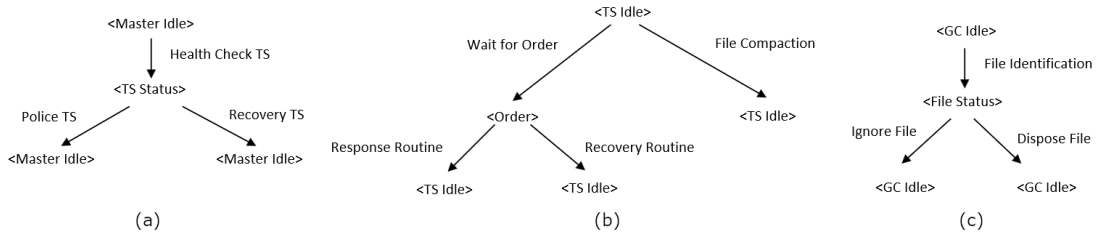


Figure 4.1: Accumulo: Scenario trees of the system components: a) Master b) TabletServer c) Garbage Collector

and prioritisation to agree on the genuine and most relevant TD for the project. This dataset, with its diverse branch setup and TD, provides a neat environment for exploring the applicability of our approaches in hypothetical projects of non-trivial scale with a higher degree of TD diversity. Henceforth, the case can provide us with an opportunity to test the scalability and applicability of our techniques in complex development setups.

4.6.1 Scenario Elicitation

Practitioners start our scenario elicitation step by identifying user or system components and their interactions before constructing scenario trees. Based on the manual of Accumulo version 1.3 (i.e. the closest version to the commit creation date), there are two parties: clients and the server, where clients can only send data manipulation requests while the server receives, executes requests and maintains the health of TabletServers. Scenario trees of the server components, which are Master, TabletServers and Garbage Collector, are illustrated in Figure 4.1. From interactions of different components and their responsibilities, three scenarios can be established: S1) Server-Client Response: this scenario relates to the Master server, which receives data manipulation requests from clients. S2) TabletServer recovery: this scenario occurs when one TabletServer is failed and is detected by the Master server. S3) File Maintenance: this scenario mostly relates to file operations performed periodically

by the Garbage Collector. The responsibilities of each component in three scenarios are shown in Table 4.8. Once scenarios are defined, TD in each file is mapped according to corresponding components, whereas common libraries are included in all scenarios. The mapped TD backlog is provided in the above repository. For an example of a conflict in TD rankings based on these scenarios, one may present S1 to impose a TD ranking based on the use of COTS tools [13] to broadly prioritise TD in the backlog, as this operating scenario is related to all components. Another decision-maker who uses S2 for an illustration may impose a TD ranking from a risk-oriented TD prioritisation approach [97] to minimise attack vectors, which may be visible during TabletServer recovery. Another one may impose a TD ranking with the use of a cost-benefit TD prioritisation approach [103] to prioritise TD to mitigate as one believes the system is mature enough and should only be maintained during its low-workload scenario (e.g. Garbage collector) and any TD repayment should be based on its profitability of components.

Table 4.8: Defined Scenarios based on Components Responsibilities

S1: Srv & Cli Response	S1.1 Master	Health Check TS, Police TS
	S1.2 Tablet Server (TS)	Wait for Order, Response Routine
	S1.3 Garbage Collector (GC)	File Identification, Ignore File
S2: TabletServer Recovery	S2.1 Master	Health Check TS, Recovery TS
	S2.2 Tablet Server	Wait for Order, Recovery Routine
	S2.3 Garbage Collector	-
S3: Files Maintenance	S3.1 Master	-
	S3.2 Tablet Server	File Compaction
	S3.3 Garbage Collector	File Identification, Dispose File

4.6.2 Solutions Generation

Once all TD are mapped in scenarios, practitioners input parameters for a project constraint and an MOEA. Table 4.9 indicate defined project constraints and MOEA parameters in this case study. The commit with the creation date of ”2011-10-04T00:46:07Z” is selected. After the exclusion of irrelevant packages to the core project (i.e. ”example” and ”samples” packages), the TD backlog yields a total of 1912 TD. The project budget is 20% of 41941, which is the total effort required

Table 4.9: Project constraint and MOEA parameters

Description	Values
No. of variables (n_{var}) - Accumulo	1912
No. of variables (n_{var}) - Cocoon	869
Budget - Accumulo	8389
Budget - Cocoon	4327
Evaluation	500000
Population Size	100
Crossover Probability (SPX)	0.9
Mutation (BitFlip)	$1/n_{var}$

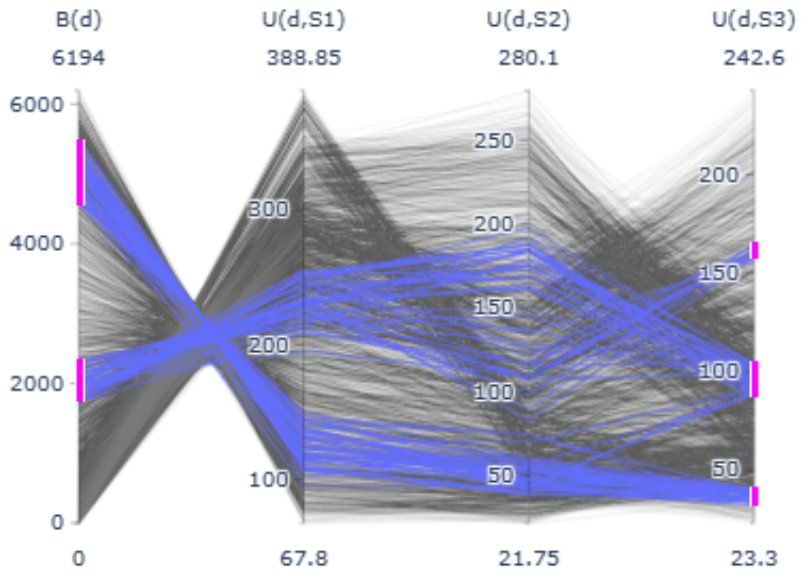


Figure 4.2: Example of an interactive parallel coordinates plot of generated solution. Blue lines indicate different solutions and their objective values, and vertical axes represents different objective functions.

to mitigate all 1912 TD. 2492 unique solutions are generated from 25 runs of the NSGAII algorithm via JMetalPy version 1.5.5 (8 duplicated solutions are omitted). Practitioners can visualise those solutions in an interactive parallel coordinates plot to investigate the characteristics of generated solutions. Figure 4.2 is an example of an interactive parallel coordinate plot of generated solutions via plotly-express version 0.4.1. Fitness values of generated solutions, generated solutions, and script for an interactive parallel coordinates plot are included in the repository. Further statistical analysis of generated solutions is shown in the evaluation section.

After the characteristics of generated solutions are acknowledged, practitioners

Table 4.10: Accumulo: Fitness values of selected solutions

d	$B(d)$	$U(d, S1)$	$U(d, S2)$	$U(d, S3)$
305	0	388.85	86.45	46.1
1199	6	342.25	280.1	110.6
899	6	301.45	167.4	242.6

Table 4.11: Accumulo: Pay-off matrix of selected solutions in fitness function $U(d, S1)$ against $U(d, S2)$ (Nash equilibria in bold)

		$U(d, S2)$		
		#305	#1199	#899
$U(d, S1)$	#305	388.85,86.45	**388.85,280.1**	388.85,167.4
	#1199	342.25,86.45	342.25,280.1	342.25,167.4
	#899	301.45,86.45	301.45,280.1	301.45,167.4

can select a set of solutions. This can be established via an off-the-shelf software or spreadsheet. As an example, practitioners select solutions that yield the most fitness function in each scenario-specific objective, which are solution numbers 305, 1199 and 899 for fitness values of scenario S1, S2 and S3, respectively. The fitness values of selected solutions are shown in Table 4.10. After that, practitioners can then utilise pay-off matrices to determine the suitability of solution sets in different aspects. This can be manually plotted, plotted via off-the-shelf software, or an algorithm for the computation of equilibria (e.g. NashPy in Python). Table 4.11 illustrates a pay-off table of selected solutions with fitness function $U(d, S1)$ against $U(d, S2)$. Table 4.12 summarise the total Nash equilibria of selected solutions in different fitness function competitions. All competitions in Table 4.12 are done via off-the-shelf software and later verified by NashPy version 0.0.28. From Table 4.12, it can be concluded that solution #1199 and solution #899 are capable of handling all defined scenarios with the least budget used among all three solutions.

Table 4.12: Accumulo: Total Nash equilibria of selected solutions in different fitness function competitions

d	$U(d, S1)$ vs $U(d, S2)$	$U(d, S2)$ vs $U(d, S3)$	$U(d, S1)$ vs $U(d, S3)$	$B(d)$ vs $U(d, B1)*$	$B(d)$ vs $U(d, S2)$	$B(d)$ vs $U(d, S3)$	Total
#305	1	0	1	1	0	0	3
#1199	1	1	0	1	1	1	**5**
#899	0	1	1	1	1	1	**5**

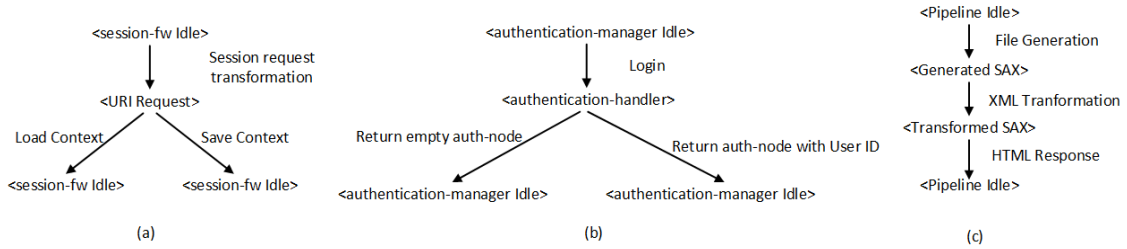


Figure 4.3: Cocoon: Scenario trees of processes: a) URI Authentication b) Read/Write session context c) Pipeline Processing

4.7 Case Study B: Apache Cocoon

The second project aims to explore the potential of our approach in a real industrial system. To achieve this, we utilise the TD backlog of Apache Cocoon, which can be found in The Technical Debt Dataset version 2.0 [68]. This version of the dataset only considers the main developmental branch of the project, and thus, it can illustrate a more faithful presentation of identified TD of the system.

4.7.1 Scenario Elicitation

According to Apache Cocoon manual [7], Cocoon can be customised and flexible to different levels of clients (e.g. XML-focused users, offline and online developers). Therefore, there are a significant number of use cases for Cocoon. Here, we choose three scenarios which apply to the majority of clients. Those scenarios are URI authentication, read/write session context and pipeline processing. In general, URI authentication is a process which authenticates clients via a defined Java class. Read/write session context relates to data manipulation by the clients after logging in. Pipeline processing is related to creating XML before responding to clients via HTML. The code-level interactions and/or sequences for each scenario in detail can be observed in [7]. Figure 4.3 illustrates trees of the aforementioned scenarios. Conflicts in TD rankings can stem from assumptions/conditions utilised in these three

scenarios. A decision-maker with a background in software architecture may point out a potential architectural degradation during URI authentication and impose a TD ranking focusing on architectural TD [132]. In contrast, one who focuses on code-based TD may believe that TD can be significantly propagated in the most frequently utilised components (e.g. read and write context) and impose a TD ranking based on an approach that focuses on the severity of code-based TD [3]. On the other hand, another decision-maker may illustrate the necessity of pipeline processing for the software life span, point out possible TD impact of pipeline processing to customers (e.g. internal errors in XML files), and imposes a TD ranking with the use of a customer-oriented multiple decision strategy criteria TD prioritisation approach [96].

4.7.2 Scenario Generation

After scenarios are defined, it is possible to input project constraints and commerce for solution generation via MOEA. Table 4.9 indicate defined project constraints and MOEA parameters used in this case study. Among 26322 commits, we chose the commit "2003-03-09T00:11:46Z" as it is the first identified commit. In this commit, a total of 11451 TD is identified. However, there are packages that are irrelevant to the system (e.g. "sample", "test", and "source" packages) and issues with 20 TD rendering their effort to be empty. After those irrelevant packages and issues are excluded, 869 TD is selected. For this case study, 20% of the total effort required to mitigate all 869 TD (i.e. 4327) is input as the goal budget. After 25 runs of the NSGAII algorithm via JMetalPy version 1.5.5, 2488 unique solutions are generated. Practitioners can utilise an interactive parallel coordinates plot to acknowledge an overview of generated solutions and select solutions to be included in a set of solutions similar to Figure 4.2. Table 4.13 shows an example of select

Table 4.13: Cocoon: Fitness values of selected solutions

d	$B(d)$	$U(d, S1)$	$U(d, S2)$	$U(d, S3)$
208	0.4	0.4	15.6	1573.4
1797	3.4	927.6	1587.8	101.6
2380	78.4	1317.6	1386.6	142.2

solutions in case practitioners select generated solutions based on the fitness values of scenarios S1, S2 and S3, respectively.

4.8 Evaluation

Evaluation of a decision-making framework can be difficult. This is because some of the inputs and steps of the framework may require human involvement and experts' judgement. Additionally, the inputs may be situation-, application- or domain-dependent, and it may require retrospective reflection on the quality of the decision after a lapse of time. However, we follow desired criteria for decision-making in software engineering [23] and a software engineering evaluation guideline to assess randomised algorithms [9]. We specifically use the below-mentioned criteria:

Neutrality: The subjectivity of practitioners shall be controlled and standardised to produce a bias-free evaluation. Our framework requires some inputs from practitioners, including TD, their estimates and selected TD prioritisation approach(es). We use a case study to evaluate our approach. We select the TD backlog and a TD prioritisation approach and use them in all the cases throughout the evaluation. Additionally, we omitted the part where practitioners select generated solutions. This is because this step can heavily rely on practitioners' inputs and competency. The evaluation used statistical tests to evaluate all generated solutions against the baseline approach adopted in this chapter.

Informative: The application of an approach must be informative to be applied

in different settings. In the case of controlled experiments, the results should show potential, and the information should assist in replication into the real-world environment. Nevertheless, we use real-world examples. In particular, the TD backlog in our evaluation is a real-world TD backlog, and much of the information is derived from actual documentation linked to the TD. Although the TD prioritisation ranking approach is hypothetical, TD calculation is based on widely used TD metrics, including expert-defined severity score and SQALE (maintainability) ratings, henceforth providing the necessary information for replication and wider application beyond our context.

Replicable: Effective evaluation must be reproducible yet verifiable and falsifiable. The evaluation shall provide evidence to ensure that similar results can be obtained and reproduced, following similar the said procedure and inputs. For our evaluation, mining queries, scenario-mapped TD backlog, MOEA running scripts, generated solutions and evaluation scripts are provided. Additionally, the areas that require randomness are explicitly seeded, where possible.

To demonstrate the potential of our approach, we perform statistical tests of all outcomes from our framework and compare them against the baseline approach (explained in detail in the next sub-section). For reproducibility, we provide all scripts in every step to ensure that outcomes from our framework can be traceable and available for future verification or falsification.

We evaluate our framework by looking at its effectiveness in producing high-quality solutions (i.e. solutions with high fitness value), its ability to alleviate Deep Uncertainty and its ability to improve existing approaches. As practitioners in the real-world environment can produce different decisions on the same TD backlog (due to rationales and/or assumptions, etc.), we aim to gather different TD prioritisation decisions as much as possible as the baseline. For both projects, we use a constrained

random generation approach as our baseline (its rationale is detailed below). For each solution, we randomly pick the number of TD to be selected in the decision and the number of TD to be prioritised for TD mitigation. Then, we add that solution as a baseline sample only if the cost for that decision does not exceed the defined budget. The seeded random generation script is developed in Python 3.9 and provided in our repository.

Deep Uncertainty in TD rankings is caused by a conflict in TD rankings. Thus, any TD ranking, as long as it is in the feasible solution space (e.g. within a project budget), is applicable to cause a conflict. In practice, decision-makers utilise external evidence (e.g. information on the problem context, the system environment, their competency, and their judgement) to filter only feasible TD rankings to impose. Thus, we believe a constrained random generation is the best representation for illustrating the search space without reliance on the problem context, which can be subject to perceivers, and software artefacts, which, if there are any, can be incomplete, inadequate or biased as a conflict in TD rankings may be informally acknowledged, ignored or treated as a trivial matter.

We specifically aim to answer the following research questions:

RQ1. Does our framework produce high-quality solutions better than the baseline approach? The objective of this research question is to investigate the ability of our framework to capture severe TD. Answers to this question can enable us to indicate the strengths and weaknesses of our framework when compared to the baseline case.

RQ2. Does our framework have better potential in alleviating Deep Uncertainty when compared to the baseline approach? This research question aims to explore the ability of our framework to produce solutions with a similar level of

quality. As this question is directly related to the main problem of our interest, answers to this question can indicate the usefulness of our approach to producing similar solutions, which then alleviate Deep Uncertainty in TD prioritisation.

RQ3. Does our framework improve the quality of solutions? This question is based on our conjecture that our framework can be utilised to improve existing TD prioritisation approaches without their replacement. This research question looks into the improvability of our framework compared to the baseline, in capturing severe TD and in producing solutions similar in quality. The answer to this question can point out the potential of our framework in being incorporated with existing TD prioritisation approaches to produce high-quality solutions and alleviate Deep Uncertainty in the respective activity.

RQ4. How does our framework perform under different numbers of identified TD? The objective of this research question is to investigate the scalability of our framework. Answers to this question can indicate the potential and limitation of our framework when utilised in systems with identified TD in different scales, enabling practitioners' awareness when utilising the presented framework.

4.8.1 Evaluation Setup

Our evaluation is based on two projects. Those projects are conducted with the same goal budget (i.e. 20 per cent of the total effort required to mitigate all identified TD) under the default "Sonar-Way" profile, and each project is individually compared with the baseline. For the first project, Apache Accumulo on a commit with the creation date of "2011-10-04T00:46:07Z" is selected. This project can be accessed in The Technical Debt Dataset, Version 1.01 [68]. After the exclusion of irrelevant packages to the core project (i.e. "example" and "samples" packages), the

TD backlog yields a total of 1912 TD. We define the goal budget for our evaluation to be (i.e. 20 per cent of the total effort required to mitigate all 1912 TD). We use Scenario Elicitation and Solutions Generation steps described in sub-section 4.6.1 and 4.6.2, respectively, to map all TD to scenarios and generate solutions to be evaluated. After 25 runs of an MOEA, the total number of 2492 unique solutions are sampled, and 8 redundant solutions are discarded. The second project is Apache Cocoon. This project can be accessed in The Technical Debt Dataset Version 2.00 [68]. Among 26322 commits, we chose the commit "2003-03-09T00:11:46Z" with the creation analysis key "AV4laW8sG0XxMwG_V0EC" as it is the first identified commit. In this commit, a total of 11451 TD is identified. However, there are packages that are irrelevant to the system (e.g. "sample", "test", and "source" packages) and issues with 20 TD rendering their effort to be empty. After those irrelevant packages and issues are excluded, 869 TD is selected. Selected TD are relevant to three scenarios of interest. These scenarios are illustrated in 4.7.1 and 4.7.2. After 25 runs of an MOEA, a total number of 2488 unique solutions are generated.

Since decisions from practitioners in the real-world environment may or may not utilise scenario-specific assumptions/conditions, the global utility score is instead used for evaluating these two samples. The global utility score is based on a hypothetical calculation of the TD score. For TD_j, it can be mathematically expressed as:

$$U(TD_j) = Main(TD_j) * Svrty(TD_j) \tag{4.21}$$

Where $Main(TD_j)$ and $Svrty(TD_j)$ are the inverted maintainability rating and the severity of TD_j, respectively. Those parameters are defined by developers and/or SonarQube and are available in the data set. The total TD score of decision d_i can

be calculated by:

$$U(d_i) = \sum_{TD_j \in d_i} U(TD_j) \tag{4.22}$$

This equation is utilised to calculate samples from our framework and the baseline approach to incorporate statistical tests for answering RQ1. For RQ2, as our aim to alleviate Deep Uncertainty is to maximise the Jaccard similarity index of all decision-makers (Eq. 4.11), we calculate the sample-wide average Jaccard similarity index of decision d_i as:

$$J(d_i) = \frac{\sum_{d_j \in D_{samp}} J(d_i, d_j)}{n_{samp}} \tag{4.23}$$

Whereas D_{samp} is a set of decisions in a sample, n_{samp} is the number of samples (which is 2492 and 2488 in our evaluation). $J(d_i, d_j)$ is the Jaccard similarity index of selected TD positions between d_i and d_j. This equation is utilised to calculate the average sample-wide Jaccard similarity index of all solutions in both two samples. For answering RQ3, the improvability index of a value A over a value B is defined as a following:

$$I(A, B) = \frac{A}{B} \tag{4.24}$$

Thus, the improvability index total score of decision d_i over decision d_j is calculated as:

Table 4.14: Accumulo: Statistical Properties of All Samples

Metrics	Sample source	Mean	ST.Dev	Min	Max
$U(d_i)$	(O)ur Approach	304.30556	93.16472	109.75	419.45
	(B)aseline	216.67701	123.43575	0.25	419.4
$U(j_i)$	Our Approach	0.1867	0.01302	0.14950	0.22230
	Baseline	0.05713	0.02339	0.00083	0.09631
$I(U(d_i))$	O ->B	0.80939	0.33457	0.51657	1.97428
	B ->O	7.30442	51.1803	0.72557	1217.222
$I(J(d_i))$	O ->B	0.30752	0.02191	0.257	0.38214
	B ->O	5.83299	13.11227	1.93862	225.5499

Table 4.15: Cocoon: Statistical Properties of All Samples

Metrics	Sample source	Mean	ST.Dev	Min	Max
$U(d_i)$	(O)ur Approach	3086.838	1093.748	314	4327
	(B)aseline	2346.327	1242.31	1	4327
$U(j_i)$	Our Approach	0.16245	0.02585	0.09393	0.2245
	Baseline	0.05024	0.01882	0.00133	0.07768
$I(U(d_i))$	O ->B	1.0098	0.88558	0.54225	7.47238
	B ->O	6.41536	73.25143	0.71339	3086.838
$I(J(d_i))$	O ->B	0.31731	0.05164	0.22377	0.53481
	B ->O	5.15059	9.1216	2.09118	121.8514

$$I(U(d_i), U(d_j)) = \frac{U(d_i)}{U(d_j)} \tag{4.25}$$

And the sample-wide average improvability index of decision d_i defined as:

$$I(J(d_i), J(d_j)) = \frac{J(d_i)}{J(d_j)} \tag{4.26}$$

The improvability index is less than 1 when there is a degradation from the first value to the second value. It is equal to 1 when there is no improvement or degradation, and it is greater than 1 when there is an improvement from the first value to the second value. This equation is utilised to calculate the improvability of the global utility in both samples.

Table 4.14 and 4.15 show statistical properties of all samples produced by Equation 4.22 to 4.26. Once calculated samples are produced, all samples are statistically tested for their normality via the Shapiro-Wilk tests. Samples from our approach are tested against samples from the baseline in respective metrics for their monotonic

Table 4.16: Accumulo: Result from Shapiro-Wilk tests of all samples

Metrics	Sample source	test statistic	p-value
$U(d_i)$	Baseline	0.94937	0
	Our Approach	0.91137	0
$J(d_i)$	Baseline	0.95595	0
	Our Approach	0.99683	0
$I(U(d_i))$	B -> O	0.0859	0
	O -> B	0.80971	0
$I(J(d_i))$	B -> O	0.24095	0
	O -> B	0.98332	0

Table 4.17: Cocoon: Result from Shapiro-Wilk tests of all samples

Metrics	Sample source	test statistic	p-value
$U(d_i)$	Baseline	0.95177	0
	Our Approach	0.89814	0
$J(d_i)$	Baseline	0.91967	0
	Our Approach	0.98218	0
$I(U(d_i))$	B -> O	0.03873	0
	O -> B	0.53784	0
$I(J(d_i))$	B -> O	0.28718	0
	O -> B	0.97361	0

relationship via Spearman's Rank Correlation. All statistical tests in this paper are calculated with a 95 per cent confidence interval using Python version 3.9.6 and SciPy version 1.7.3. Table 4.16 and 4.17 shows that the p-value of the Shapiro-Wilk test in all samples is less than 0.05, and therefore, there is enough evidence to reject a null hypothesis that each sample is normally distributed. Table 4.19 and 4.17 indicate that p-values from all samples in respective metrics are more than 0.05. Therefore, it failed to reject a null hypothesis that there is no statistically significant correlation between each pair of samples. Since the normality of all samples cannot be assumed, and there is no monotonic relationship with each pair of samples in respective metrics, one-tailed Mann-Whitney U tests are chosen to answer our research questions.

The null hypothesis of our one-tailed Mann-Whitney U test is the probability of

Table 4.18: Accumulo: Result from Spearman's Rank Correlation tests of all samples and metrics

Metrics	correlation	p-value
$U(d_i)$	-0.00366	0.85491
$J(d_i)$	0.00335	0.86736
$I(U(d_i))$	-0.00365	0.85565
$I(J(d_i))$	0.00335	0.86736

Table 4.19: Cocoon: Result from Spearman's Rank Correlation tests of all samples and metrics

Metrics	correlation	p-value
$U(d_i)$	0.02014	0.31536
$J(d_i)$	0.0056	0.78026
$I(U(d_i))$	0.02014	0.31536
$I(J(d_i))$	0.0056	0.78026

Table 4.20: Accumulo: Result from Mann-Whitney U tests of all samples

Metrics	MWU statistic	asymptotic(p-value)
$U(d_i)$	4394007	0
$U(j_i)$	6210064	0
$I(U(d_i))$	853338	1
$J(U(d_i))$	0	1

randomly selected values from a sample of our approach to be greater than randomly selected values from a sample of the baseline is **not greater** than the other way around (i.e. the probability of randomly selected values from a sample of the baseline to be greater than randomly selected values from a sample of our approach).

One-tailed tests are selected as all research questions are interested in improvements of one sample over the other.

4.9 Result and Discussions

4.9.1 Research Question 1

For Accumulo, Table 4.14 and 4.20 respectively show statistical properties (the first row and the second row) and the results from the Mann-Whitney U test of corresponding samples (the first row). According to Table 4.14, our approach reports 16 per cent higher mean than that of the baseline for the considered different samples. Our approach has also reported 13 per cent smaller standard deviation and the highest minimum of total TD scores on the used samples. Mann-Whitney U test of samples in the total TD score of our approach and the baseline (Table 4.20)

Table 4.21: Cocoon: Result from Mann-Whitney U tests of all samples

Metrics	MWU statistic	asymptotic(p-value)
$U(d_i)$	4190724	0
$U(j_i)$	6190144	0
$I(U(d_i))$	1144032	1
$J(U(d_i))$	0	1

indicate that there is enough evidence for rejecting the null hypothesis (p-value less than 0.05). However, it is important to note that both approaches are only 0.05 different in producing the maximum TD scores. This is because we do not include the total TD score as one of the objective functions.

For Cocoon, the first and second rows of Table 4.15 show statistical properties of $U(d_i)$ from our approach and the baseline. It reports that our approach achieves 13 per cent higher mean than that of the baseline. Additionally, our approach has reached the highest minimum of total TD scores and has achieved 6 per cent smaller standard deviation than those samples. Table 4.21 show the Mann-Whitney U test of samples in the total TD score of our approach and the baseline that there is enough evidence for rejecting the null hypothesis (p-value less than 0.05).

To answer RQ1: We conclude that our approach statistically produces higher quality solutions than the baseline, where the outcomes from our approach can achieve 13 to 16 per cent higher than the total TD score of baseline on average.

4.9.2 Research Question 2

The results incorporating RQ2 are located on the third and the fourth row of Table 4.14 and the second row of Table 4.20. According to Table 4.14, our approach exhibits more than 50 per cent higher mean than the baseline approach when considering the chosen samples. Additionally, our approach also reports higher minimum and higher maximum Jaccard similarity index (50-100 per cent higher), as when

compared to the baseline on the used samples. Table 4.20 indicates that there is enough statistical evidence to reject the null hypothesis (p-value less than 0.05).

The result is in a similar fashion for Cocoon. The third and fourth rows of Table 4.15 indicate that our approach has achieved a 52 per cent higher mean than samples from the baseline. Additionally, our approach produces samples with significantly higher minimum and maximum Jaccard similarity index, at 48 and 97 per cent, respectively. Mann-Whitney U test of samples in Jaccard similarity index of our approach and the baseline (Table 4.21, second row) indicate that there is enough statistical evidence to reject the null hypothesis (p-value less than 0.05).

To answer RQ2: We conclude that our approach produces more similar solutions; henceforth, it has better potential in alleviating Deep Uncertainty. Compared to the baseline, our approach yields the sample-wide Jaccard similarity index at least 52 per cent higher on average. Additionally, our approach shows a higher minimum-maximum sample-wide Jaccard similarity index when compared to the baseline one.

4.9.3 Research Question 3

For Accumulo, Results in the fifth row of Table 4.14 indicates statistical performance in the improvability index of the total TD score of samples when moving from our approach to the baseline, whereas the sixth row of the same table shows statistical performance in the improvability index of the total TD score of samples when moving from the baseline to our approach. The mean of the total TD score of samples when moving from our approach to the baseline (0.80939) is significantly lower than the mean of the total TD score in samples when changing from the baseline to our approach (7.30442). This indicates that changing from the baseline to our approach generally improves the total TD score (i.e. the improvability index on average is

higher than 1).

The seventh row in Table 4.14 indicates statistical performance in the improvability index of the sample-wide Jaccard index of samples when changing from our approach to the baseline, and the eighth row in Table 4.14 shows statistical performance in samples of the improvability index in the sample-wide Jaccard index of samples when changing from the baseline to our approach. The result shows that the mean of the sample-wide Jaccard index in chosen samples when changing from the baseline to our approach is significantly higher than the mean of the sample-wide Jaccard index in chosen samples when changing from our approach to the baseline.

For the Cocoon case, the fifth row of Table 4.15 shows statistical performance in the improbability index of the total TD score of samples when moving from our approach to the baseline, whereas the sixth row of Table 4.15 shows the other way around (i.e. statistical performance in the improbability index of the total TD score of samples when moving from the baseline to our approach). It can be observed that the mean of the total TD score from samples when moving from the baseline to our approach is approximately 70 per cent higher than the mean of the total TD score from samples when moving from our approach to the baseline.

For samples of the improvability index of the sample-wide Jaccard index, the seventh row of Table 4.15 indicates statistical performance in the improvability index of the sample-wide Jaccard index of samples when changing from our approach to the baseline, and the eighth row of the same table shows statistical performance in the improvability index of the sample-wide Jaccard index of samples when changing from the baseline to our approach. Similar to Accumulo, the result indicates that the mean of the sample-wide Jaccard index in chosen samples when changing from the baseline to our approach is significantly (80 per cent) higher than the mean of the sample-wide Jaccard index in chosen samples when changing from our approach

to the baseline.

For both cases, it is worth noting that the standard deviation of samples in the total TD score and the sample-wide Jaccard index when changing from the baseline to our approach is significantly higher than samples in the same metrics when changing from our approach to the baseline. This shows that improvement when changing from the baseline approach to our approach can vary. This claim is supported by the result indicated in Table 4.20 and 4.21 where we failed to reject the null hypothesis on all tests related to the improvability index. Thus, we encourage practitioners not to blindly accept all generated solutions and select only generated solutions that yield better fitness values than the existing solutions. This can be mitigated by pay-off matrices illustrated in the previous section.

To answer RQ3: Although there is evidence that changing from the baseline to our approach can generally improve the outcomes, expert judgement can be necessary to confirm the findings. Visualisation techniques can also help in visualising the solution landscape to inform decision-making.

4.9.4 Research Question 4

Table 4.22: Result from Cross-Project Mann-Whitney U tests of all samples

Metrics	MWU statistic	p (two-tailed)	p (one-tailed)
$\hat{U}(d_i)$	3118296	0.719	0.36
$U(j_i)$	4837125	0	0
$I(U(d_i))$	2711415	0	1
$J(U(d_i))$	2800264	0	1

To evaluate Research Question 4, we conduct cross-project statistical hypothesis tests. The objective of those statistical tests is to validate whether the quality of samples from a project with a larger TD number is worse than a project with a smaller TD number. We achieve this by conducting two-tailed and one-tailed Mann-Whitney U tests, whereas samples from Apache Accumulo represent samples

from a project with a larger TD number and samples from Apache Cocoon represent samples from a project with a smaller TD number. The null hypothesis of our two-tailed Mann-Whitney U test is that the probability of a randomly selected value from Accumulo is **not different** than the probability of a randomly selected value from Cocoon.

In case the two-tailed Mann-Whitney U test indicates a statistical difference between randomly selected values from Accumulo and Cocoon, one-tailed Mann-Whitney U Tests are used to investigate those samples further. For one-tailed Mann-Whitney U Tests, the null hypothesis is that the probability of a random sample from Accumulo to be greater than a random sample from Cocoon is **not greater** than the other way around (i.e. the probability of a random sample from Cocoon to be greater than a random sample from Accumulo).

Since the total TD score illustrated in Equation 4.22 can be subject to the number of TD and available budget for each project, it is difficult to conduct cross-project statistical tests without modification of this metric. Hence, we tweaked Equation 4.22 to be relative to the maximum total TD score of the sample. Formally, the proportioned TD score of decision d_i can be expressed as:

$$\hat{U}(d_i) = \frac{U(d_i)}{U(d_{max})} \tag{4.27}$$

Whereas $U(d_{max})$ represents the maximum total TD score of the decision sample. A result of a two-tailed Mann-Whitney U test (Table 4.22, first row) indicates that there is no significant statistical difference in proportioned TD score between decision samples from Accumulo and decision samples from Cocoon. For the sample-wide Jaccard index, the one-tailed Mann-Whitney U test (Table 4.22, second row)indicates that it is insufficient evidence to reject the null hypothesis for

the sample-wide Jaccard index that the probability of a decision samples from Accumulo is greater than decision samples from Cocoon is **not greater** than the other way around. We postulate that because the higher number of TD can lead to more diverse solutions being generated, thus the sample-wide Jaccard index of samples from Accumulo is generally lower than the sample-wide Jaccard index of samples from Cocoon. However, we conjecture that diverse solutions can contribute to a higher chance of improving the existing solution. In other words, a project with a higher TD number might yield higher improvability in proportioned TD score and the sample-wide Jaccard index. This supports by the one-tailed Mann-Whitney U test, which indicates that random samples of the improvability of proportioned TD score from Accumulo are greater than random samples of the improvability of proportioned TD score from Cocoon (Table 4.22, third row). Similarly, the one-tailed Mann-Whitney U test indicates that random samples of the improvability of the sample-wide Jaccard index from Accumulo are greater than random samples of the sample-wide Jaccard of proportioned TD score from Cocoon (Table 4.22, fourth row).

To answer RQ4: We conclude our approach produces solutions for TD mitigation without significant statistical differences related to the number of identified TD. We also found that our approach has more potential to improve the existing solution with projects with a higher identified TD number. However, it should be noted that our approach might produce more similar solutions in projects with a smaller identified TD number.

4.10 Threat to Validity

Threats to internal validity concern the mining queries and the data set itself. As we mined our data set from the repository, we cannot guarantee the correctness of the TD manifestation and admission, as the data set carries little semantic information that relates to the source of the admitted debts, confidence, and measurement.

Scenario mapping is based on the functionalities of each component, file naming convention and the project structure. Though this might not be an optimal representation of the real project, it is one of the most effective ways to rationalise the responsibilities of each component without introducing additional business rules and documentation. The improvability index in our evaluation is utilised under the assumption that related parameters are different from zero. This assumption is reasonable as those descriptive parameters start from one (i.e. "E" in the inverted maintainability rating and "INFO" in the severity rating). We utilise MOEA runners from JMetal 5.11; however, we cannot guarantee the exact result after 25 runs as JMetal is randomly seeded by default.

It is important to note that a TD prioritisation approach that we have utilised in our evaluation is a hypothetical one, where its utility score is calculated using commonly used software metrics by the TD community. This decision is primarily to preserve the reproducibility of our evaluation by not introducing additional software documentation or business rules apart from the real-world software reference of the project. Although the result might differ when a formulation of the TD utility score is changed, the overall result can be similar since our framework is the extension of the existing TD prioritisation approach.

As a threat to conclusion validity, it is important to note that we used an asymptotic (approximated) p-value during the Mann-Whitney U tests that are utilised for

our conclusions instead of the exact (look-up table) approach. This is simply because the look-up table approach is limited to the number of samples less than 30. We acknowledge the limitation of the Jaccard similarity index on the inability to differentiate binary sets (e.g. the Jaccard similar index of [0,1,1] and [1,1,0] is 1 (i.e. the total similarity)) and mitigate this by calculating the Jaccard similar index of selected positions in each decision instead (e.g. [2,3] and [1,2] are represented for the two sets above). Hence, different calculations (or the lack thereof) can result in totally different samples and conclusions.

Threats to external validity are related to the generalisation of our test results; it should be noted that our framework's overhead cost can heavily rely on the TD identification, measurement, and prioritisation approach(es) selected by practitioners. Our evaluation does not include the evaluation of pay-off matrices as the existence of a Nash Equilibrium of a finite game can be mathematically proven [86]. Since there always exists a (weak) Nash Equilibrium in a finite game, pay-off matrices are always compatible to use as the visualisation of generated solutions.

As our framework utilises several techniques in resolving Deep Uncertainty in TD rankings (e.g. MOEAs, Nash Equilibrium), there can be a learning curve and cost associated with applying these techniques (e.g. fine-tuning or running MOEAs, establishing objective functions by their own, understanding Nash Equilibrium). This issue can accumulate the cost of learning and/or understanding utilised techniques and associated TD prioritisation approaches into the cost of using the framework. However, we believe those costs outweigh the possible loss from a Black Swan event from Deep Uncertainty in TD rankings (e.g. an eruption of severe TD) which can contribute to a development crisis or early software obsolescence.

4.11 Conclusion

The contribution is a novel exploration of the presence of Deep Uncertainty and Black Swan during TD decision-making. We have explored the manifestation of Black Swan in TD decision-making, covering identification, measurement and prioritisation, and have provided mitigation guidelines to practitioners for acknowledging and mitigating potential Black Swans. We have formulated a systematic framework to alleviate Deep Uncertainty in TD prioritisation and rankings. The framework used formal Scenario Analysis and Many-Objective Optimisation.

Formal Scenario Analysis is used to elicit TD scenarios that relate to their manifestation and cover assumptions, rationales, severity, progression etc. Many-Objective Optimisation is utilised to generate solutions based on these scenarios-based fitness functions. Optionality and Nash Equilibrium are used to select a set of generated solutions that can better cope with uncertainty in a changing environment.

We use statistical tests, including Mann-Whitney U, to evaluate the applicability of our framework. The results of our evaluation indicate that our framework is capable of improving the quality of TD rankings, alleviating disagreement on TD ranking and Deep Uncertainty in TD prioritisation.

Overall, we explore Deep Uncertainty in TDM decision-making through a systematic mapping of theoretical aspects of Deep Uncertainty to TD counterparts. Further studies and research are needed to explore Deep Uncertainty in TDM decision-making in various industrial domains and workplace contexts. This can, for example, include conducting multiple interviews, experience reports and/or analysing feedback from TDM partitioners, decision-makers, and software developers. The result can provide a better understanding of how practitioners can better tailor or leverage our techniques for more robust and transparent TDM decision-making un-

der Deep Uncertainty and how our techniques can make the ill consequences of deep uncertainties visible to practitioners and be part of their decision-making.

Chapter 5

Skin in the Game-driven Automated Method to Alleviate Erratic Uncertainty

5.1 Introduction

Producing well-informed decisions in Technical Debt Management (TDM) is key to effective and efficient TDM. TDM decision-making can be challenging as it can suffer from ad-hoc decision management, bias, communication gaps and the inaccessibility of relevant information and evidence that can support the decision [58].

We refer to effective TDM responsibility sharing as a situation when individual TDM decision-makers share the responsibility for the negative or positive outcomes of the decision and collectively agree on the likely risks and payoffs of the decisions. Without effective responsibility sharing, decision-makers can be divided and

coupled based on their trajectory loss or benefit of their decisions. The absence of mechanisms to encourage collective and individual responsibility sharing can lead to distrust, bias, irresponsible attitude and tribal memory among decision-makers when managing TD, and can hamper the ability to produce TDM decisions which are accepted by all decision-makers. The absence of effective responsibility-sharing mechanisms during TDM decision-making can contribute to social and process debts, rendering Technical Debt (TD) decisions imbalanced, unverifiable, untraceable and sub-optimal decisions leading to:

- Inability of developers and analysts to converge on an agreement related to the root causes of the debt, its manifestation, importance, severity and/or repayment.
- Decisions that can suffer from bias in the analysis, where analysts may focus on concerns that are relevant to their expertise while ignoring others.
- The exclusive access of ”hands-on” information related to past decisions enjoyed by certain team members with tribal memory[58]) can contribute to untraceable and unreliable TD decisions as the utilised information is vulnerable to attrition or forgetting.

Additionally, the need for responsibility-sharing mechanisms is important in complex organisational structures, highly decoupled sub-teams, and where competition between factions in the team can be expected. The absence of such mechanisms can lead to communication gaps among different stakeholders and, consequently, ineffective TD decision-making. The absence of effective responsibility-sharing mechanisms can influence the quality of various TDM activities and their decision-making. Consider, for example, TD monitoring decision-making: various stakeholders may need to take individual and shared responsibility that relates to ”what” and ”how” to

monitor TD, and they should be able to understand the measurable risks of their decisions. These decisions can be important as they can impact TD identification, cost, and consequent management of these debts. Particularly, the absence of TDM responsibility sharing can lead to sub-optimal TD monitoring decisions being exercised. In TD monitoring decision-making, these sub-optimal decisions include ill-suited monitoring methods and monitoring criteria. These sub-optimal decisions can lead TD to be ill-timely detected and improperly mitigated. Although consequences from the absence of responsibility sharing can be fatal, research on TDM responsibility sharing is unfounded.

To alleviate this gap, we argue that decision-makers and decision-making during TD monitoring need to incorporate Skin in the Game thinking in the decision-making. Skin in the Game refers to the phase in which decision-makers take *measurable risks* when making a decision [114]). Skin in the Game thinking enforces decision-makers to promote *transparency* and a close understanding of the underlying risks revolving around the decision. This thinking can yield more *fair* and efficient decisions [114]. Other phases of TD management can also benefit from Skin in the Game thinking in their decision-making.

Our work is the first to discuss "Skin in the Game" thinking in TD management. We map the characteristics of TDM decision-making with that of Skin in the game thinking, including TD identification, TD measurement, TD prioritisation and TD monitoring decisions.

TDM decision-making can suffer from asymmetric information (i.e. the situation where one has more information than the others), asymmetric payoffs (i.e. the situation where one benefits from a decision more than the others), intolerant minority (i.e. the situation where the minority imposing and insisting on a decision for the whole population) and tolerant majority (i.e. the situation where the ma-

jority intentionally withstand or being neutral to an imposing decision) - detailed in section 2. The presence of these issues requires Skin in the Game thinking. These issues can lead to ineffective responsibility-sharing and hence sub-optimal decisions in TDM that are infeasible to trace and verify. To mitigate this issue, we provide systematic guidelines which stakeholders can utilise to prevent their fruition. These guidelines are inspired by the Skin in the Game thinking in decision-making. Since TD monitoring is a critical phase in TDM decision-making, we research how TDM monitoring can benefit from the guide to promote effective responsibility sharing. We propose a novel TD monitoring model which can mitigate asymmetric information, considering the perspectives of various stakeholders involved in the monitoring. The model prioritises which TD instance to monitor by weighting and monitoring the percentage of change of the weight over time. The weight of each TD instance is established after discussions between different decision-makers. Stakeholders involved in the TD management exercise can use weighting to cast their assessment in a more objective and transparent way regarding the importance of a TD instance from their own perspective. Dimensions of TD can be assigned a weight to reflect the importance of TD from given perspective(s) that are part of the analysis (e.g. business cruciality, etc.). Decision-makers can use different techniques to observe the TD. We use the percentage change to observe changes in the monitored TD(s) over time. The percentage of change can serve as an effective means to scope the discussions on TD cases that require analysts' attention. We propose a novel automated TD monitoring technique that combines Behavioural Cloning [94], and Replicator Dynamics [102]. Behavioural Cloning minimises human intervention in establishing monitoring conditions as it can learn from samples produced by practitioners. Replicator Dynamics is used to automatically select the optimal monitoring strategy from the archive. The technique can concretise situations where manual intervention by developers and maintainers is necessary, alleviating analysis pressure.

To demonstrate the ease of establishing monitoring conditions, we utilise Behavioural Cloning to mimic practitioners' behaviour. With this technique, TD monitoring conditions are centralised, and they can be automatically tracked and mitigated. These advantages can also make decision management for TD monitoring more systematic. Collectively, our contributions are the following:

- We present guidelines based on Skin in the Game [114] to ensure that decisions during TD monitoring decision-making and the potential risks are transparent, justified, responsibility shared and acknowledged by different stakeholders involved in the decision-making.
- We supplement existing TD monitoring approaches by a Skin in the Game-based guideline to mitigate asymmetric information in decision-making involving various stakeholders and decision-makers, and to better reconcile perspectives on the importance or severity of the monitored TD.
- We introduce a novel approach that combines Behavioural Cloning [94] and Replicator Dynamics, as described above. The combination can reduce unnecessary maintenance or development workload by using automatic monitoring strategies or prompting for manual intervention when necessary without compromising the effectiveness of the existing TD monitoring.

We use two industrial-scale projects and controlled experiments to demonstrate and evaluate our work. The projects are Apache Accumulo and Apache Cocoon. More than 2000 commit timestamps and more than 10000 unique TD from those two projects are sampled. We compare the results produced by our method to that of two-conditions TD monitoring methods: do-nothing and human intervention.

Our evaluation targets the ability to capture important TD and the ability to represent the significance of TD. Mann-Whitney U tests are conducted for our eval-

uation. The evaluation results indicate significant statistical differences in cost-effectiveness and the ability to represent TD significance, where our method outperforms two-conditioned TD monitoring methods.

5.2 Background: Skin in the Game

Skin in the Game is an aphorism which indicates that a decision-maker has incurred risks to an imposing decision or goal. The incurred risk can take shape in many forms, including monetary, shareholder, etc. "Skin" in this aphorism is a measurable risk or an investment, whereas "game" refers to the field under discussion. This aphorism is commonly used in the field of finance and business. This aphorism is discussed in detail by Taleb, where the author advocates that Skin in the Game is essential for fair and efficient risk management [114]. The author also points out that decision-making, in general, is prone to several symptoms which require Skin in the Game. Notable symptoms are [114]:

- **Asymmetric Information:** The situation where one does not have information as equal to the other. This situation can lead to sub-optimal decisions made by uninformed decision-makers.
- **Asymmetric Payoff:** The situation where one does not have the balance of positive and negative consequences of decisions as to the other. The unfair consequences or the lack of penalties can encourage one not to carefully make well-thought decisions.
- **Indifference/Tolerant Majority:** The situation where the majority of the population can endure and/or is neutral towards imposing changes. This incident encourages changes in decisions that do not benefit or represent the majority.

- **Stubborn/Intolerant Minority:** The situation where the minority of the population imposes and insists on changes that affect the whole population. This situation encourages changes that benefit and mirror only the minority.

These symptoms can depend on organisational structure; they can be infeasible to eliminate. In a hierarchical organisation, for example, it might be infeasible to eliminate the asymmetric payoff between the pinnacle and the rest of the organisation. The author proposes decentralisation in an organisation as a promising practice to alleviate these symptoms. More precisely, decentralisation can contribute to:

- **Informed population:** By decentralising the populations, the individuals can have a similar environment and/or receive similar responsibilities. This instance can reduce the existence of uninformed individuals as they can observe or inform each other of their similarities. Additionally, similar environments and/or responsibilities can influence individuals to share information, which reduces the situation where an individual has more information than others.
- **Non-immune decision-makers:** As there is a smaller number of members in decentralised factions, it can be easier for each member to trace and track individuals who are responsible for imposing decisions via collective memory. This advantage can increase the presence of non-immune decision-makers, which minimises the presence of asymmetric payoff in the organisation.
- **Participation of the population:** Since members in the decentralised environment have more direct consequences from decisions, they are more inclined to participate in decision-making. This instance decreases the presence of decisions made only by the stubborn/intolerant minority. Direct consequences from decisions can also encourage the population to police or review imposing decisions. This instance alleviates the presence of the indifference/tolerance majority during decision-making.

It is crucial to differentiate between two types of decentralisation: one related to the concept of Skin in the Game thinking (mentioned above) and another associated with contemporary software engineering practices such as microservices and DevOps. This study concentrates on applying the three benefits that arise from Skin in the Game thinking to managing and mitigating uncertainty in TDM decision-making without drawing any parallels or comparisons between the two types of decentralisation. The relationship and/or similarity between decentralisation in those concepts is not an objective of the thesis and requires separate research endeavours.

5.3 Related Work

There are several studies which explore the possibility of categorising TD monitoring results for interpretation. A study by Brown et al. envisions a monitoring approach in the software-critical systems, which warns and determines the level of TD over time via several decision processes [22]. A study by Marinescu leverages a concept of design flaws to measure TD in the systems, which enables the ability to continuously monitor TD [78].

The weighting of the monitoring TD set is partially motivated by [63], which indicates that valuable business and software development metrics were omitted during the decision process. Additionally, Bellomo et al. illustrate that there is a gap in effort visibility between the technical and the business stakeholders, which leads to a call for a TD monitoring approach with a focus on the system quality attribute [17].

The presented method from this chapter aims to be an extension of the existing TD monitoring approaches and software metrics. Several compatible studies include

a threshold-driven approach presented by Eisenberg [36], in which the study employs several code-related metrics to monitor and prioritise TD and to determine when the mitigation of TD is imperative. The SQALE method is one of the promising routes. This method is presented by [72] and [71]. These studies illustrate the potential of utilising the SQALE method to capture, measure and monitor code-based TD. Software metrics to measure code quality and its evolution presented by [20] and [106] can also be one potential for the application of our method.

Although this is the first study which applies Behavioural Cloning for TDM, several TDM studies utilised supervised learning, which is the underlying learning style of Behavioural Cloning. Notably, Tsustzura et al. investigate how machine learning can be applied in TDM. The paper reports that supervised learning is the dominant learning style [121]. Perez et al. present a semi-automated model-driven approach to review architectural TD and its impact. This study leverages the use of supervised learning (Naïve Bayes) to identify the advantages and disadvantages of architectural decisions located in project artefacts [92]. Most recently, Tu and Menzies utilise semi-supervised learning to minimise the labelling cost in self-admitted TD during TD identification [122].

While there are no existing TDM studies which leverage the use of Replicator Dynamics, Nash equilibrium and game theory are commonly discussed. In TDM, [77] leverages the use of epsilon Nash-equilibrium in the search for an optimal strategy for alleviating the refactoring problem. For Software Engineering, [47] apply the theory of static games with complete information for producing a 2-player software crowdsourcing model. [137] proposes an optimal software release policy from the Nash Equilibrium of mixed strategies. A study by Tong et al. presents a Nash equilibrium-driven approach for producing effective strategies in software requirement change management [120].

5.4 Approach to Skin in the Game in Technical Debt Management Decision Making

5.4.1 Asymmetries in TDM Decision-Making

We observe that the above asymmetries can be manifested in TDM when decision-makers attempt to alleviate the TD unpredictability problem. Examples of a TD unpredictability problem include the growth of TD interest [40, 41, 43], and its likely impact on the systems [19]. These TD unpredictability problems can relate to decision-making in several TDM activities. For example, the cost-benefit TD approach [103] alleviates the unpredictability of TD repayment by defining the urgency to repay TD as benefits and TD principal as cost, and TD should be repaid when its cost outweighs its benefit. Thus, this approach relates to TD identification, TD measurement, TD prioritisation and TD monitoring.

Specifically, a decision-maker may utilise evidence, assumptions, judgement, and interpretation in the search for a solution to a TD unpredictability problem. These factors can be prerequisites for other decision-makers to understand, verify or falsify an imposing solution. The imposing solution to the TD unpredictability problem can be a TD approach, framework or method which a decision-maker considers suitable for alleviating the problem based on the above supporting factors. For instance, due to the unpredictability of TD impact, a decision-maker may impose the use of an SQALE-based approach to identify, measure, and monitor the impact of TD in the backlog [71]. This approach requires knowledge of SQALE, an assumption that SQALE quality attributes are correlated to TD impact (e.g. TD with B or better in SQALE rating is less likely to cause a severe impact), acknowledgement of their limitation (e.g. constrained to code-based TD). Without understanding/acknowledging

these, other decision-makers might be unable to inspect the imposing approach for their suitability of the problem.

The prerequisites of an imposing solution can divide the audience into factions, corresponding to their capability in verifying/falsifying the imposing solution: Capable minority and Incapable majority.

- **Capable minority:** This group refers to decision-makers capable of verifying or falsifying an imposing solution. Decision-makers in this group meet the prerequisites of the imposing solution. Thus, they can improve and revise the imposing solution without relying on those outside the group. Since prerequisites of the imposing solution can involve the combination of the right level in corresponding knowledge, skills, experience, and having similar judgement and interpretation, the number of those who are capable of verifying/falsifying the imposing TD approach can be less than those who are unable to do so, rendering those who are capable as the minority of stakeholders. From the above example, there can be fewer numbers of those who have experience with SQALE or understand the SQALE-based approach than those who do not know SQALE or acknowledge this approach prior.

- **Incapable majority:** This group refers to decision-makers/stakeholders who are unable to assess, verify or falsify an imposing solution. Their incapability can be due to failure to meet the prerequisites of the imposing solution or due to unavailability during TDM decision-making. Members in this group are convertible to Capable minority when they meet the prerequisites of the imposing solution.

Factions based on the level of competency is one of community smells defined as Cognitive distance [116]. This division can result in ineffective responsibility shar-

ing, where the Capable minority find a solution for the Incapable majority. As the Incapable majority may be unable to refute the imposing solution, the imposing solution may be blindingly accepted and utilised by them, leading to unfair consequences. Specifically, factions based on the level of competency can contribute to the following TDM asymmetries:

- **Exclusive Information:** Capable minority may be unwilling to communicate to Incapability minority as frequently as communication within the group. Unwillingness to communicate can be a result of Radio silence or bottleneck [118, 117]. Unwillingness to communicate renders the information flow of the imposing solution to be exclusive to the Capable minority, resulting in asymmetric information in different groups.

  For example, the Capable majority may believe they might have to suffer from a preparation cost in communicating with the Incapable majority (e.g. for making a presentation, creating a report, visualisation, etc.). Thus, they may decide to only inform those in the Incapable majority only when there is a major change in the imposing solution (e.g. the assessment is finished, the imposing solution is finalised, or is rejected.).

- **Unfair Consequences:** Since the Capable minority are responsible for assessing, verifying or falsifying the imposing solution without reliance on the Incapable majority, they may initiate these activities under their collective perception of expected gain from the imposing solution. Due to ineffective communication/knowledge management, the Incapable majority might not notice this before the imposing solution is finalised, resulting in the solution that benefits one at the loss of the others.

  For example, the Capable minority may finalise the imposing solution by using an SQALE-based monitoring approach when they consider the approach

suitable for the problem domain. Without adequate consideration of potential overhead, however, other stakeholders (e.g. maintainers, developers) may suffer from configuration difficulties, learning curve, etc., which they may have to mitigate these issues on their own (e.g. working overtime, paying for training courses, etc.).

Ineffective communication/knowledge management can be a sign of community smells from Black Cloud [90, 118].

Unfair consequences can result in an ineffective TD decision. When this is unrecognised, it can deteriorate the quality of TDM. For instance, without consideration of its feasibility, a selected TD monitoring approach can lead to an imbalanced workload between developers/maintainers and TD analysts, where developers/maintainers are burdened by the number of TD to be monitored. An overwhelming workload of developers/maintainers can lead to late recognition of TD changes. If this issue is unnoticed, it can support the manifestation and growth of TD.

Additionally, we observe three limitations in the current TD literature. These limitations can influence the existence of the above asymmetries as follows:

- **High entry requirements in TD solutions:** Most TD approaches, frameworks, or methods require practitioners with specific levels of knowledge, experience, specific assumptions and/or interpretations.

  For example, to utilise a portfolio theory-based TD prioritisation approach [40], one must have a basic financial knowledge of Portfolio Theory and assume that the TD repayment return of the portfolio conforms to a normal distribution. To utilise a monitoring approach proposed by Seaman et al. [103]., one must have knowledge of TD from code smells and be capable of estimating TD efforts based on historical data, Likert scales or experts' opinions.

High entry requirements in a TD solution can divide stakeholders based on their capability, forming into Capable minority and Incapable majority, as explained above.

- **The lack of explicit guidelines on representing opinions:** Several TD approaches, frameworks or methods lack a guideline for eliciting opinions and representing them. This limitation can demotivate the Capable minority to elicit stakeholders' opinions and prefer other methods. When stakeholders' opinions are related to those in the Incapable majority, this issue can deter possible information/knowledge sharing between different groups, collectively worsening Asymmetric Information in them.

  For instance, a monitoring approach by Seaman et al. [103] utilises TD parameters, which can be measured by the Likert scale, historical data or experts' opinions. However, they do not provide specific guidelines or requirements for eliciting such parameters from experts. Another example is an SQALE monitoring approach [71], which indicates that the remediation cost of TD should be calibrated by the project or the organisation. However, they do not provide a method for one to follow. Limitations from these studies may motivate practitioners to use more accessible means (e.g. Likert scale, historical data, default values) than eliciting from stakeholders' opinions, deterring opportunities for sharing knowledge/information between different groups.

- **Inability for the majority to verify the imposing solution:** Most TD approaches, frameworks, or methods do not provide a route to verify their possible outcomes without having relevant knowledge, skill, experience, judgement or interpretation. This limitation can render the Capable minority to verify their imposing solution, unknowingly contributing to the solution with unfair payoffs.

  For instance, maintainers in the Incapable majority, who do not have knowl-

edge of metrics in a specific COTS TD tool [13], are unable to justify a produced TD monitoring decision. Due to this issue, they can suffer false positives TD due to an improper COTS TD profile.

The Capable minority with Exclusive Information may treat prerequisites as the norm, finding them unnecessarily to be formally documented and managed. This incident can cause ad-hoc decision management, where the detail of TDM decisions can reside in "tribal memory", and there is a lack of formal decision tracking and management.

Ineffective communication/knowledge management between different groups can complicate group members in future communication, potentially widening communication gaps between them. Communication gaps and ad-hoc decision management are identified as significant TDM issues in IBM industrial experience report [58].

5.4.2 Guideline for managing TD decisions with Skin in the Game

To mitigate these asymmetries in technical debt decisions, we propose a guideline which brings the advantage of decentralisation proposed by Taleb to mitigate asymmetric information and asymmetric payoff, indifference/tolerant majority and stubborn/intolerant minority during TDM decision-making. We illustrate the guideline by providing examples used in TD identification, TD measurement, TD prioritisation and TD monitoring. The guideline and examples in four selected TDM activities are shown in Table 5.1. Overall, a well-informed population can alleviate the presence of Exclusive Information. Non-immune decision-makers can reduce the presence of Unfair Consequences from imposing decisions, and the separation of Capable Minority and Incapable Majority can be eased with the participation of the population.

Table 5.1: Suggested guidelines for alleviating Exclusive Information, Unfair Consequences, Capable minority and Incapable majority with examples in TD identification, measurement, prioritisation and monitoring.

Symptoms which require Skin in the Game	Suggested guidelines for TDM	Example in TD identification	Example in TD measurement	Example in TD prioritisation	Example in TD monitoring
Exclusive Information	A pool of information related to each TD activity to all stakeholders related to decision-making. Evidence/information utilised during decision-making must be verified that it is bias-neutralised, traceable and falsifiable.	TD identification reports from a cost-of-the-shelf tool are presented to all stakeholders related to decision-making. Report of TD identification from a cost-of-the-shelf tool includes the name and version of the tools, utilised profile and parameters, which can be replicated when necessary.	Utilised software metrics and their calculations are included in a pool of information for TD measurement decision-making. Sources of utilised software metrics are included in the pool of information during decision-making.	Before its approval, TD ranking and its calculation are presented for further review by stakeholders during decision-making. Rationale behind TD ranking and the source for its calculation is provided to all stakeholders related to decision-making.	References for candidate TD monitoring approaches are present in the search for suitable TD monitoring approaches. Method and proposed parameters are included in all candidate TD monitoring approaches.
Unfair Consequences	All stakeholders involved in decision-making acknowledge and accept their benefit or loss from imposing decisions Approved decisions by stakeholders must be traceable to the proposed stakeholder(s).	After imposing a decision, TD experts forewarn that they might not suffer from unidentified TD as much as the other developers. The proposers of approved TD identification approach selection are documented for further reference.	Developers who are responsible for TD mitigation point out and accept that they suffer from mis-estimated TD due to sub-optimal decisions more than decision-makers. Stakeholders responsible for an approved TD measurement profile in COTS are included in the profile description.	A part of the stakeholders indicates their potential loss due to infeasible TD ranking to the rest of the stakeholders participating in decision-making, which is accepted by the rest of the stakeholders. Names of stakeholders who proposed an approved TD prioritisation re-positioning are documented in the meeting note.	TD analysts warn that maintainers can suffer the most from the presence of monitored TD based on their proposed TD monitoring approaches selection. Names of stakeholders who propose approved TD monitoring conditions are included in the meeting note.
Capable minority and Incapable majority	Stakeholders must be informed regarding the potential consequences of decisions during decision-making. Each stakeholder must indicate the rationale behind the approval/disapproval of each decision	Stakeholders acknowledge that they can collectively suffer from project penalties due to unidentified TD. A developer indicates the reason for his/her approval on a selection TD identification approach is his/her agreement with the proposers on possible types of TD in the system.	Before decision-making, participating stakeholders are informed of the possibility of mis-estimated TD and its consequences to customers. Developer indicates that his/her reason for agreeing with an estimate of one TD is because he/she cannot refute the proposer's information.	Stakeholders acknowledge that improper TD rankings based on their decisions can contribute to potential development crises. TD analyst agrees with the proposed TD ranking because the approach behind the ranking is widely used in other software projects.	Stakeholders are warned that unsuitable TD monitoring approaches based on their decisions might lead to the presence of severe TD, which can be labour-intensive to mitigate. A TD expert approves the proposer's monitoring conditions as he/she believes that those conditions are justified.

5.5 Applying and Formalising Skin in the Game Thinking in Technical Debt Monitoring

We contribute to a method and guidelines that use Skin in the Game thinking for TD monitoring and decision-making leveraging the mapping described in Section 5.4. The method consists of three steps.

5.5.1 Three Steps Method for Applying Skin In the Game Thinking to TD Monitoring

Ineffective responsibility sharing can manifest in any TDM activities. As described in Section 5.4.1, the mapping to Skin in the Game can benefit any of these activities. For this contribution, we specifically focus on TDM monitoring, and we describe a three steps method that uses this mapping. The method can help in mitigating ineffective responsibility sharing in TD monitoring. Nevertheless, prescribed methods are suited for other TDM activities as they can follow similar reasoning below.

There are three reasons that we specifically investigate monitoring as a TDM activity using Skin in the Game thinking: Firstly, since TD monitoring decisions are related to ”what” and ”how” to monitor TD, TD monitoring decisions can be related to the TD monitoring approach, TD monitoring criteria and its mitigation. Those subjects can affect TD analysts, developers and maintainers. As three different roles of TD stakeholders are involved, it can be more challenging for them to establish shared responsibility. Secondly, decisions related to TD monitoring can directly impact possible TD manifestation or TD growth in the future. This factor can be postulated differently depending on the perspectives of stakeholders on TD monitoring decisions. Thus, it can be difficult for them to agree on possible consequences

and related TD monitoring decisions collectively. Thirdly, current state-of-the-art TD monitoring approaches and commercial off-the-shelf tools are mainly based on a specific TD technical, financial or economic perspective. Stakeholders to TD monitoring decisions can have different opinions on a suitable TD monitoring approach depending on their speciality, their experience, their preference, etc. This characteristic hampers the ability of stakeholders to TD monitoring decisions to converge on possible consequences of the decisions and responsibility of TD monitoring decisions.

We present a novel TD monitoring method that leverages the mapping and its guidelines described in Section 5.4, and we describe a novel formulation based on Skin in the Game. This method addresses the TD monitoring problems formally defined in Section 5.5.2, and consists of the following steps:

1. Approaches Selection: Practitioners select a suitable TD monitoring approach based on our guideline presented in Section 5.4.2. The objective of this step is to balance potential Exclusive Information imposed during decision-making by presenting relevant information or evidence and documenting related rationale in a systematic manner. During this step, practitioners discuss and negotiate possible consequences from decisions to different classes of decision-makers. As information or evidence from different decision-makers is put together, Exclusive Information can be mitigated during discussions.

2. Weighting of monitored TD: This step aims to balance Exclusive Information in different classes of decision-makers (Formally defined in Section 5.5.2). During this step, practitioners discuss with different classes of stakeholders to establish the weight of monitored TD. Monitoring weight can be estimated via various estimation methods (e.g. planning poker, Wideband Delphi, etc.). The weight of each TD can encapsulate its sensitivity to monitoring and/or importance to the system, the organisation, etc., which can be shared with

different classes of stakeholders. This advantage can alleviate the existing communication gaps. Furthermore, explicit monitored TD weighting can contribute to traceable evidence of TD monitoring configuration, which mitigates the presence of ad-hoc decision management and Exclusive Information from certain decision-makers.

3. Establishing automated monitoring conditions: The objective of this step is to mitigate unfair payoff of workload between TD analysts, maintainers or developers (Formally defined in Section 5.5.2). To achieve this, we present a novel approach which automatically selects corrective actions based on TD monitoring conditions via Behavioural Cloning. In addition, we use an automated technique to select a stable monitoring strategy with Replicator Dynamics. With the combination of these two, it is possible to have an option which is more effective than do-nothing and costs less than manual human intervention. First, practitioners archived alternative TD monitoring strategies in case the system requires human intervention. During this step, practitioners share and discuss monitoring conditions and alternate TD strategies with participating stakeholders. Knowledge and information sharing during this step can narrow communication gaps between the different classes of stakeholders. Next, practitioners map monitoring conditions and preferred actions via demonstrations. With the application of Behavioural Cloning, automated actions can then be conditioned and monitored to conform to practitioners' behaviour.

As these three steps involve several TD monitoring decisions, it is intended to be utilised with the presented guideline based on Skin in the Game in Section 5.4.2.

5.5.2 Asymmetries in TDM monitoring decision-making

In this section, we describe limitations in current TD monitoring approaches in the literature, which can support the result of ineffective responsibility sharing to be utilised. Specifically, we observe that the current TD monitoring approaches lack consideration in representing opinions due to their lack of weighting set (described in Section 5.5.2) and the inability of different stakeholders to reject the imposing solution, which can be observed in the limited corrective actions from TD monitoring results, rendering unfair workload between developers/maintainers and TD analysis inevitable (described in Section 5.5.2).

To alleviate the first limitation, we present a TD monitoring model. This model consists of a TD weighting set, combined with the percentage change, to represent the magnitude of change between different points in time (described in Section 5.5.3). The weighting set can be used to elicit the importance of TD from stakeholders, embedding opinions from different perspectives in the model.

Based on our model, we propose an automated TD monitoring method. This method utilises automated techniques to alleviate the second limitation. Specifically, automated techniques are presented to alleviate the burden of developers/maintainers, filtering only when their intervention is necessary. These automated techniques are Replicator Dynamics and Behavioural Cloning (Section 5.5.4).

Weighting in TD monitoring

Let TD be a set/backlog of TD consisting of many td. For TD_α, it can be expressed as:

$$TD_{\alpha} = \{td_1, td_2, \cdots, td_p\} \tag{5.1}$$

During TD identification and measurement, the weight of each TD can be defined and represented by its context, severity, urgency, relevant goals, related business rules, etc. The weight of each TD can be represented as TD parameters, such as TD interest and/or TD severity, depending on selected TD measurement approaches. For the weighting set of TD_{α} from TD measurement, it can be denoted as:

$$W_{mea}(TD_{\alpha}) = \{w_1, w_2, \cdots, w_p\} \tag{5.2}$$

For $\sum W = 1$. Since several TD monitoring approaches do not emphasise the individuality of each TD in the backlog, the weighting set of TD monitoring is unfounded by default. Without any explicit representation for the TD monitoring weighting set, maintainers might be required to treat all monitored TD equally:

$$W_{mon}(TD_{\alpha}) = \{\frac{1}{p}, \frac{1}{p}, \cdots, \frac{1}{p}\} \tag{5.3}$$

As all monitored TD is treated as equal, it might be necessary for maintainers or developers to intervene in case there is any change in monitored TD. This can cause unfair payoffs between TD analysts and maintainers or developers, as TD analysts might not receive fair consequences (i.e. the workload) related to the outcomes of TD monitoring decisions. To mitigate this, decision-makers need to specify the importance of monitored TD. However, as decision-makers have different specialities and backgrounds, they can have different trajectories in monitored TD. There can be conflict between different decision-makers as they have different perspectives on the

weighting set. Without a reconciliation approach, an improper weighting set might be utilised. By doing so, Exclusive Information can be manifested as information only from certain decision-makers is utilised in the weighting set selection.

Limited Actions from TD Monitoring Result

For most available TD monitoring approaches, there are two decisions that take different costs. First, it is suggested to initiate human intervention once monitoring conditions are met. Human intervention can be related to re-prioritisation, investigation of monitored TD, etc. This decision results in the cost of human intervention. For TD monitoring approach x (i.e. TDM_x), TD_α and d_{hu} as human intervention, it can be denoted as:

$$C(TDM_x, TD_\alpha, d_{hu}) = C_{x,\alpha,hu} \tag{5.4}$$

Note that d_{hu} is a decision in a member of the decision set $D = \{d_{hu}, d_{not}\}$. $C_{x,\alpha,rep}$ is the cost of human intervention for TD monitoring approach x with the TD backlog α. In case the monitoring conditions are not met, practitioners do not need to do anything, which yields no additional cost:

$$C(TDM_x, TD_\alpha, d_{not}) = 0 \tag{5.5}$$

Whereas d_{not} is a decision of doing nothing and a member of a decision set D. Since there are only two decisions available (i.e. $|D|= 2$), maintainers or developers might be left to non-systematically solve the issue when the human intervention cost is higher than their budget. This circumstance is possible since the human

intervention cost is related to the number of monitored TD that met the monitoring conditions. This instance can cause unfair workloads between different classes of stakeholders, as TD analysts might not have to compensate for this issue.

5.5.3 Weighting of monitored TD

For a set of monitored TD, each monitored TD has the monitoring weight, which is represented in a set of monitoring weights. For TD backlog α defined in 5.1, a set of its monitoring weight can be:

$$W_{mon}(TD_{\alpha}) = \{wm_1, wm_2, wm_3, \cdots, wm_p\}. \tag{5.6}$$

The monitoring weight of each TD can represent volatility, severity, sensitivity or importance of that TD depending on the focus of practitioners. Each monitoring weight can be estimated by stakeholders, commercial-of-the-shelf software or combinations of these sources. For stakeholder-oriented estimation, it can be conducted with mainstream estimation methods. By doing so, a wider audience of decision-makers can participate and inform of the importance of each monitored TD from the weighting set. When more decision-makers are informed of the significance of monitored TD, Exclusive Information, where specific information can be accessible for certain factions of decision-makers, can be mitigated.

Based on the monitoring weight of TD, it is possible to represent the contextual-based utility score for all monitored TD into a single value. In this paper, we propose the use of percentage change. Percentage change is widely used in finance to represent the magnitude of changes between two different time points. Since TD monitoring is related to changes in values at different points in time, percentage

change can be a compatible representation to illustrate the growth or the decay of each TD before and after receiving TD monitoring results.

For percentage change of td_1 between t_1 and t_2 (i.e. $PC(td_1, t_1, t_2)$), it can be calculated by:

$$PC(td_1, t_1, t_2) = |\frac{V(td_1, t_2) - V(td_1, t_1)}{V(td_1, t_1)}| \times 100 \tag{5.7}$$

Whereas $V(td_1, t_1)$ and $V(td_1, t_2)$ are the value of td_1 measured at t_1 and t_2 respectively. It is important to note that we take the absolute value from the percentage change since the abnormality of monitored TD between different times can be both positive and negative in practice. For example, changes in TD misestimation can contribute to different values than the previous timestamp, which can be higher (i.e. positive percentage change) or lower (i.e. negative percentage change). By utilising the monitoring weight of TD, the backlog-wide percentage change can be established. The backlog-wide percentage change of $TD\alpha$, between t_1 and t_2 can be denoted as:

$$PC(TD_\alpha, t_1, t_2) = \sum_{i=1}^{p} PC(td_i, t_1, t_2) \times wm_i \tag{5.8}$$

Since the percentage change represents the magnitude of changes in monitored TD between different points in time, stakeholders can understand the abnormality in monitored TD by comparing the average percentage change of the backlog with the current percentage value of the backlog. Hence, the percentage change can minimise the necessity to understand the knowledge behind utilised TD monitoring metrics, and thus it is possible to use the percentage change to communicate with

a wider range of stakeholders where there is an absence of knowledge in utilised TD monitoring metrics. This advantage can facilitate bridging the communication gaps between stakeholders with different knowledge of TD.

Not only this equation can facilitate practitioners during TD monitoring by encapsulating changes in all monitored TD between different points in time. It can enable the use of an automated option to mitigate TD after receiving TD monitoring results. The detail of this is illustrated in the next subsection.

5.5.4 Automated Monitoring Corrective Actions

We introduce an additional corrective action as an automated alternative. The objective of this automated option is to reduce the practitioner's mitigation cost to less than a manual human intervention, whereas its outcomes can be more effective than do-nothing. To achieve this, Behaviour Cloning is employed to propose appropriate corrective actions according to monitoring conditions. Additionally, Replicator Dynamics is utilised to automatically propose an alternative TD monitoring strategy when human intervention is required.

Behavioural Cloning is chosen based on the characteristics of the problem landscape. Monitoring conditions and their mitigation routes are highly subjective. These factors can be subject to the size of the system, stakeholder preferences, the cruciality of the system, etc. Hence, the concrete application of monitoring conditions of a specific system at a given time can be impractical for other systems and their future uses. Additionally, the subjectivity of monitoring conditions can be impractical to elicit. More precisely, monitoring conditions and their mitigation routes can be established based on past projects, trial-and-error, lessons learned, knowledge of stakeholders, preferences, etc. These sources of decisions can be difficult to trace

and explain in a precise manner. Thus, it is more feasible to establish monitoring conditions from the surface than to model them. Based on these observations, the imitation of stakeholders' behaviour in the mapping monitoring conditions and their mitigation routes is a promising route to establishing an automated alternative.

Formally, we define that there can be three members in the set of corrective actions:

$$D = \{d_{not}, d_{hu}, d_{auto}\} \tag{5.9}$$

Where d_{auto} is an automated selection of archived TD monitoring strategies. Each mitigation is triggered based on the lower and upper bound of a monitoring value:

$$f(d) := \begin{cases} d = d_{not} & L_{not} \leq d \leq U_{not} \\ d = d_{auto} & L_{auto} \leq d \leq U_{auto} \\ d = d_{hu} & L_{hu} \leq d \leq U_{hu} \end{cases} \tag{5.10}$$

Where $f(d)$ is a decision function depending on a monitoring value, L_{not}, L_{auto} and L_{hu} are a lower bound value of a decision of do nothing, automated monitoring strategy selection, and human intervention, respectively. U_{not}, U_{auto} and U_{hu} represent an upper bound value for a decision to do nothing, automated monitoring strategy selection, and human intervention. These lower- and upper bounds can be gathered from stakeholders. There are several methods compatible with obtaining these ranges, including direct demonstration, questionnaire, TD backlog analysis, conducting an estimation method, combinations of those approaches, etc.

These ranges concretely define when human intervention is required, enabling negotiation between decision-makers from different factions before their realisation. Consequently, unfair payoffs between different factions of decision-makers can be eased with this step. Once these ranges are obtained, they can be input as training data for Behavioural Cloning.

We leverage the use of Replicator Dynamics to propose an optimal TD monitoring strategy. We utilise Replicator Dynamics as it can produce non-mutated solutions when comparing quasi-species equation-based approaches or genetic algorithms. This step starts when practitioners archive alternative TD monitoring strategies (e.g. based on potential cost-saving, cost-benefit of manifested components, architectural significance of manifested components, etc.). Each archived strategy is then allocated with initial equal populations. For archived strategy s, the number of its population at time t_0 is:

$$Ss(t_0) = \frac{1}{N} \tag{5.11}$$

Where N is the total number of archived strategies. Archived strategies are completed against each other, and the competition result of each strategy is represented as its fitness. Fitness for each archived strategy is the sum of the competition outcomes multiplied by the number of the competition's population. For archived strategy s at time t, it can be calculated as:

$$f_s(t) = \sum_{r=1}^{N} Sr(t) \times T(s, r) \tag{5.12}$$

Where $T(s, r)$ is the competition outcome from archived strategy s and archived

strategy r. The competition outcome can be calculated based on the utility score, which practitioners previously defined. To determine whether one performs better than other strategies, the average fitness of the population can be used for comparison. The average fitness of the population at time t is defined as:

$$\overline{f(t)} = \sum_{r=1}^{N} Sr(t) \times f_r(t) \tag{5.13}$$

Where f_r is the fitness of archived strategy r. Based on the two equations above, the replicator equation for each archived strategy can be found. The replicator equation of a strategy is the number of the population. This is multiplied by the difference between the fitness of that strategy and the average fitness of the population. For archived strategy s at time t, it can be expressed as:

$$\frac{dSs(t)}{dt} = Ss(t-1) \times (f_s(t-1) - \overline{f(t)}) \tag{5.14}$$

The replicator result is then added to the population for the next epoch. For archived strategy s at time t_1, the number of population can be calculated by:

$$Ss(t_1) = Ss(t_0) + (dSs(t_0) \times (f_s(t_0) - \overline{f(t_0)} \times dt) \tag{5.15}$$

These steps can be done iteratively until the ending condition is met. Once the ending is reached, an archived strategy with the highest average number population is selected as a proposed TD strategy for practitioners. In case there is more than one strategy with the highest average population, strategies with the highest average population are reported to practitioners instead. Although, in the worst case, the

number of outcomes might require practitioners to select an optimal strategy on their own, the cost to select an optimal strategy is less than the cost of manual human intervention for re-prioritisation, investigation, etc. Thus, the objective of an automated alternative can be met regardless of the number of its outcomes.

5.6 Case Studies

We use two industrial-scale projects. In this section, we discuss how our method and its rationale can be applied to the chosen projects. In Section 5.7, we report on the evaluation. These are Apache Accumulo [8] and Apache Cocoon [7]. These two projects are chosen based on differences in their backlog. Those projects are large industrial projects with more than 2000 manifested components and a number of commits of more than 1000. However, Apache Accumulo has a smaller number of manifested components than Apache Cocoon, but it has a wider range of effort required to mitigate TD in each component. Additionally, Apache Cocoon has a significantly smaller number of collaborators per commit than Apache Accumulo. This is because the data set of Apache Accumulo has considered all developmental branches, whereas the data set of Apache Cocoon has focused on only the main developmental branch.

Since Apache Accumulo considered all developmental branches, it might not be a faithful representation of the development environment. However, there exist situations where multiple collaborators are responsible for the same component at the time. As a result, the number of identified heterogenous TD and the number of collaborators in each component can be more diverse than a project with only one development branch. This setting can lead to a TD monitoring challenge in resource allocation as monitoring the higher number of identified TD can be labour-intensive.

Therefore, we select this project as it offers an ample opportunity to demonstrate our method for its cost-effectiveness and scalability within a complex TD monitoring environment.

We look at TD based on their manifested file/component. This decision aims to reduce the subjectivity in the importance of TD. The importance of TD can be subject to types of TD, manifested functions, and stakeholders' specialities, where the importance of those is not explicitly addressed in their software documentation. We consolidate TD into their component via mining queries, where the effort and severity of a component are derived from its manifested TD. The effort of each component is the sum of values in the "efforts" column. Values in the "efforts" column are defined by SonarQube with the default profile. The severity of each component is the mean of values in the "severity" column. Values in this column are inputted by developers and provided in the data set.

In both of our controlled cases, we assume that changes in TD monitoring results can be distinguished by the mean, minimum and maximum value of the population. This assumption aims to mimic practitioners without their reliance during evaluation. Practitioners may observe changes in TD monitoring results and subconsciously categorise the degree of change as usual (e.g. close to the mean), trivial (e.g. close to minimum or minimum) or significant (e.g. close to maximum or maximum) before deciding on a suitable action. Based on this assumption, we utilise quartiles to categorise different degrees of changes in TD monitoring results, where 1 (i.e. one minute) is used to distinguish between lower-bound and upper-bound in different quartiles. Though these are hypothetical estimations, they can be a starting point in eliciting more precise estimates from stakeholders.

For reproducibility, three weighting sets utilised in both cases are based on their respective software documentation and their TD backlog. These three weighting

sets illustrate the situation when TD analysts impose TD monitoring solutions from a different perspective. Architectural Weighting set is based on the number of components during its operation, described in the "Accumulo Design" section of the software manual for Accumulo [8]. For Cocoon, the Architectural weighting set is based on diagrams and descriptions provided in the "Concept" section of its documentation [7]. The architectural weighting set represents an imposing TD monitoring solution that focuses on technical aspects of components (e.g. SQALE-based [71] or COTS code-based TD monitoring approach [13]). Functional weighting set is derived from the description of different components provided by authors in the "Accumulo Design" section for Accumulo and in the "Overview" section for Cocoon. This weighting set represents an imposing TD monitoring solution that looks at business aspects of components (e.g. cost-benefit-based TD model [103]). Architectural and Functional Weighting sets are converted from the Likert scale, ranging from 1 (lowly important) to 5 (highly important). Severity-based weighting set is based on the "Severity" column of the table of aggregated components (created by the mining queries described above). This weighting set represents an imposing TD monitoring solution which focuses on system security (e.g. security/risk-based TD model [97]). The severity weighting set ranges from 0 (i.e. unimportant) to 1 (highly important).

TD backlog of Apache Accumulo and Apache Cocoon are obtained from The Technical Debt Data Set version 1.01 and version 2.0, respectively [68]. For both cases, we chose the earliest commit creation date, which has more than one component changed. These are "10/14/2011 12:31:52 PM" for Apache Accumulo and "3/10/2003 4:35:45 PM" for Apache Cocoon. Data sets are fetched, and changes in commits for each component are analysed by mining queries, which we presented in our repository [11]. Statistical Properties of those two cases are shown in Table 5.2. Those cases follow the step provided in Section 5.5.1 and are analysed individually as their results can differ.

Table 5.2: Statistical Properties of Apache Accumulo and Apache Cocoon

Description	Accumulo	Cocoon
No. Of Manifested Component	2542	8701
No. Of Changes in Commit	1025	1535
Average Components per Commit ($\overline{CppCm}$)	8.23024	1.76091
Average No. Of Author per Commit	10.28	1.76091
Average No. Of Author per Component ($\overline{ApC}$)	1.24905	1
Average Effort per Component ($\overline{EpCm}$)	834.6283	26.56974
Effort Percentage Change per Commit with Architectural Weighting (Ef-PCCmA) - 1st Quartile	1258.97436	284.9642
Ef-PCCmA - 2nd Quartile	7279.55217	614.87261
Ef-PCCmA - 3rd Quartile	36860.15667	1203.92854
Ef-PCCmA - 4th Quartile	9124310.504	65121.67534
Minimum Effort per Component	1	1
Maximum Effort per Component	210560	990
Average TDS per Component	309.875	9.96374
TDS Percentage Change per Component (TDS-PCCp) - 1st Quartile	70	67.2186
TDS-PCCp - 2nd Quartile	96.66667	90.0885
TDS-PCCp - 3rd Quartile	342.0202	99.67915
TDS-PCCp - Standard Deviation	38840.56669	1740.169
Human Intervention Cost	1176.92193	1.37524

5.6.1 Case A: Apache Accumulo

Table 5.3: Effort Percentage Change TD Backlog of Apache Accumulo

td_i	Component	Effort	Change Effort	$PC(td_i, t_1, t_2)$	Arch	Func	Sev
1	client/···/ThriftTableOperationException.java	8960	1942->8960	361.38	4/5	3/5	0.4/1
2	data/···/InitialMultiScan.java	10880	99->10880	10889.9	2	2	0.4
3	data/···/InitialScan.java	10880	99->10880	10889.9	2	2	0.4
···	···	···	···	···	···	···	···
39	tabletserver/···/TooManyFilesException.java	16000	97->16000	16394.85	4	4	0.4

Table 5.3 shows the effort percentage change of Apache Accumulo. There are 38 components changed in this commit with the creation date of "10/14/2011 12:31:52 PM". Change Effort indicates the change in effort from the previous commit creation date, which is later utilised to calculate $PC(td_i, t_1, t_2)$ in the next column (Equation 5.7). For this project, it is possible to establish three weighting sets based on different rationales. Architectural Weighting set is based on the number of operating components, and Functional Weight is based on the responsibilities of each component. Those two weighting sets are based on Apache Accumulo documentation [8]. The severity weighting set is based on values defined by the authors, which included the data set. Based on these weighting sets, it is possible to calculate for $PC(TD_\alpha, t_1, t_2)$ as Equation 5.8. $PC(TD_\alpha, t_1, t_2)$ based on Architectural weighting set is 4361.86328. $PC(TD_\alpha, t_1, t_2)$ with Functional weighting set is 4436.77960, and

$PC(TD_\alpha, t_1, t_2)$ based from the severity-based weighting set is 5215.87497. Assuming that $PC(TD_\alpha, t_1, t_2)$ based on the Architectural weighting set is currently in use, it is possible to define lower bounds and upper bounds of monitoring value. In this case, we define U_{not}, U_{auto} and L_{hu} as the 1st, 2nd and 3rd Quartile for Ef-PCCmA of the project, and L_{not}, L_{auto}, U_{hu} as the minimum effort percentage change possible (i.e. 0), $U_{not} + 1$, the 4th Quartile of Ef-PCCmA of the project and respectively. We used Behaviour Cloning architecture and the baseline scripts from [91] with these lower- and upper-bound for monitoring values. To accompany this script, we utilise gym version 0.18.3, Tensorflow version 2.6.0 and NumPy version 1.19.2. A revised version of the "ready-to-run" Behaviour Cloning script, the TD Backlog in detail and its analysis are included in our repository.

Since the total percentage change in the effort of this commit falls between L_{auto} and U_{auto}, automated monitoring strategy selection is initiated, searching for a stable strategy with Replicator Dynamics. We set the payoff for each strategy based on $PC(TD_\alpha, t_1, t_2)$ of each weighting set as strategies in the competition. This fitness function based on each weighting set represents the ability of that weighting set to capture most of the percentage change in the effort of the commit. We set the termination criteria of Replicator Dynamics to 10^5 iterations. Figure 5.1 shows Replicator Dynamics from strategy derived from these three weighting sets. It is evident that Strategy C (i.e. a strategy derived from the severity weighting set) is the most preferable. This can be verified by Table 5.4, which illustrates a payoff table of different weighting sets when complete with each other. Based on the table, it is evident that a weighting strategy based on severity yields the best possible outcomes against other strategies. Thus, it is the most stable strategy to be chosen by an automated monitoring strategy selection.

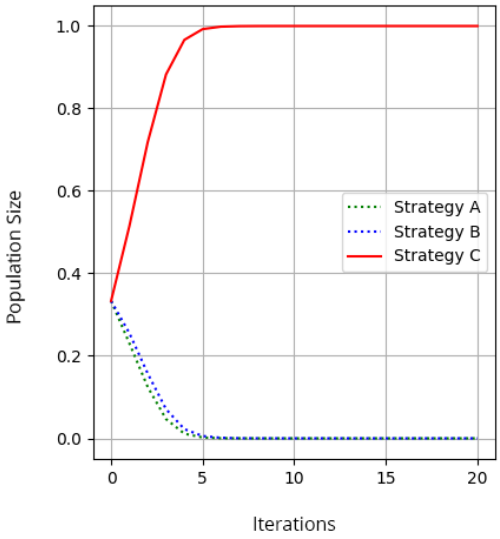


Figure 5.1: Apache Accumulo: Replicator Dynamics of Three Strategies (Strategy A - Architectural Weighting Set, B - Functional Weighting Set, C - Severity Weighting Set). Population size plateau afterwards.

Table 5.4: Apache Accumulo: Payoff Table for Three Different Weighting Sets. Each value represents the payoff of strategies in the row completed against strategies in the column.

Strategies	Arch	Func	Sev
(Arch)ictectural-based weighting	0	-74.9163	-854.012
(Func)tional-based weighting	74.91633	0	-779.095
(Sev)erity-based weighting	854.0117	779.0954	0

5.6.2 Case B: Apache Cocoon

Table 5.5: Effort Percentage Change TD Backlog of Apache Cocoon

td_i	Component	Effort	Change Effort	$PC(td_i, t_1, t_2)$	Arch	Func	Sev
1	mail/···/MessagingException.java	10	90->10	88.88889	1	1	0.2
2	mail/···/Part.java	2	10->2	80	1	1	0.2
3	mail/···/Session.java	2	64->2	96.875	1	1	0.2
4	mail/···/MimeMessage.java	2	122->2	98.36066	1	1	0.2

There are four effort percentage changes of Apache Cocoon for the commit "3/10/2003 4:35:45 PM", which is shown in Table 5.5. $PC(td_i, t_1, t_2)$ in the table is calculated from the previous values and changed values of each component, which are shown in the fourth column. The Architectural weighting set, Functional weighting set are based on Apache Cocoon user manual [7] whereas the author-defined severity weighting set is included in the data set. Since the manifested components are related to the mailing component, which is not the core contribution of Apache Cocoon, those components are lowly weighted in all three weighting sets. It is possible to establish lower and upper-bound for automated monitoring values. Similar to the previous case, it is possible to define L_{not} and U_{not} from the minimum effort percentage change possible (i.e. 0), the 1st Quartile of Ef-PCCmA, L_{auto} and U_{auto} as $U_{not} + 1$, and the 2nd Quartile of Ef-PCCmA of the project, and L_{hu} and U_{hu} as the 3rd and 4th Quartile for Ef-PCCmA of the project respectively. The same Behaviour Cloning scripts from the previous case can also be applied here. By assuming that the architectural weighting set is previously used, it is possible to determine that the total effort percentage changes of the commit is less than U_{not}. Thus, those changes are ignored.

5.7 Evaluation

Evaluation of any computation method can be challenging when it involves human input. Our method depends on the weighting set and the lower and upper-bound set,

which are defined by practitioners. Furthermore, the effective choice of lower- and upper bounds has a direct influence on the effectiveness of our automated monitoring strategy selection. This can be observed in Section 5.6 as the stated inputs are application- and context-dependent. Based on desired characteristics of decision-making in software engineering [23] and software engineering evaluation statistical guidelines to assess randomised algorithms [9], we evaluate our method with the following criteria:

Neutrality: We control and standardise the subjectivity of practitioners. The aim is to produce a bias-free evaluation. To achieve this, we divide our contribution and substitute human input with case studies and related evidence throughout the evaluation. Some part of the evaluation and the baseline approach assumes the range of possible scenarios from human input. Statistical hypothesis tests are utilised to evaluate all samples from our method against the ones from the baseline.

Reproducible: To ensure that our evaluation is replicable and falsifiable, we provide our repository, which can be publicly accessible on [11]. Mining queries, TD backlog for all cases, analyses, and evaluation scripts are included in our repository. Where it is possible, we also provide analyses used in this paper in a spreadsheet format.

Informative: As one of our objectives is to make sure that the evaluation result can be widely applied in different settings, our controlled experiments use real-world projects with different software purposes and scales. All rationale utilised in this section and Section 5.6 is derived from actual documentation in responding projects, where they can be applied to other real-world projects.

We aim to evaluate whether our method based on Skin in the Game thinking produces better outcomes than the baseline of traditional two TD monitoring de-

cisions (i.e. do-nothing or human intervention). The baseline is divided into three different scenarios and expressed in detail in the next subsection. Specifically, our evaluation investigates the following characteristics:

Cost-effectiveness: We evaluate whether our method can alleviate Unfair Consequences when maintainers or developers have to compensate when the cost to achieve monitoring conditions, defined by TD analysts, is higher than the budget. We define cost-effectiveness as the balancing ability to produce cost-saving TD monitoring decisions and the effectiveness of TD monitoring decisions to capture severe TD. Since an automated TD monitoring decision presented by our method aims to reduce human intervention to capture severe monitoring TD, we conjecture that our method can produce promising results against the baseline.

Contextual weighting sets: We observe the ability of our method and the baseline to produce contextual weighting sets of monitoring TD. The contextual weighting set of monitoring TD can be business capability, monitoring sensitivity, cruciality of manifested component, etc., depending on practitioners. The ability to produce the contextual weighting sets can indicate the ability to elicit and inform the importance of each monitoring TD to a wider range of stakeholders. This can alleviate Exclusive Information where information on monitoring TD resides in certain factions of decision-makers. Although the effectiveness of the weighting set can be subject to practitioners' usage, we conjecture that, when it is properly utilised, the weighting set can represent the contextual importance of monitoring TD outside existing TD parameters.

Those characteristics can be summarised into the following research questions:

RQ1. Does our method produce cost-saving decisions better than the baseline approach? Answers to this question can shed light on the possibility that our

approach can produce decisions which cost less than the baseline. This answer is related to the ability of our method to facilitate Unfair Consequences between TD analysts and maintainers or developers.

RQ2. Does our method produce monitoring decisions that capture TD better than the baseline approach? Answers to this question contribute to an understanding of the cost-effectiveness of the method. Specifically, answers to this question indicate the strength or weaknesses of our produced decisions in terms of capturing and monitoring severe TD.

RQ3. Is our method capable of representing a contextual TD monitoring weighting set better than the baseline approach? Answers from this can illuminate the usefulness of our method to elicit and centralise relevant information as a weighting set for collaboration with a wider audience. This can mitigate Exclusive Information, where the information is located within only a certain class of decision-makers.

5.7.1 Evaluation Setting

We evaluate our method with two real-world industrial projects. Their statistical properties are shown in Table 5.2. For Apache Accumulo, all 2542 manifested components and 1025 changes in commits are considered. Similarly, all 8701 manifested components and 1535 changes in the commit of Apache Cocoon are included in our evaluation. In both projects, statistical analysis and tests are conducted in samples produced from our method and samples produced from the baseline.

We consider a traditional TD monitoring approach with two decisions (i.e. do-nothing or human intervention) as our baseline. Based on real-world projects, we

further divide the baseline into three scenarios.

S1 - TD Critical: This scenario represents the situation where practitioners prioritised preventing the growth in monitoring TD more than cost-saving. This case can apply to a project with a high number of existing severe TD. In this scenario, U_{not} and L_{hu} are the 1st Quartile of produced samples and $U_{not} + 1$, respectively.

S2 - Neutral: This scenario represents the situation where practitioners have information or knowledge of the current project. However, they are undetermined on the budget to mitigate monitoring TD. In this scenario, U_{not} and L_{hu} are the 2nd Quartile of produced samples and $U_{not} + 1$, respectively.

S3 - Optimistic: This scenario represents the situation where the threshold between those two decisions is optimistically defined. This scenario aims to mimic the greenfield project where practitioners have no prior information or knowledge of the current project. In this scenario, U_{not} and L_{hu} are the 3rd Quartile of produced samples and $U_{not} + 1$, respectively.

As the monitoring decisions are related to human intervention, we standardised the cost of human intervention. In our evaluation, the cost of human intervention can be calculated as follows:

$$C(TDM_x, d_{hu}) = \frac{\overline{CppCm} \times \overline{EpCm} \times \overline{ApC}}{60} \tag{5.16}$$

Where $\overline{CppCm}$ is the average number of manifested components per commit, $\overline{EpCm}$ is the average number of efforts per component, and $\overline{ApC}$ is the average number of authors per components, divided by 60, which is 60 minutes per hour. The unit of human intervention is in an hour. Since the effectiveness of the Replicator

Dynamics and an automated TD monitoring strategy selection are dependent on the practitioners' input and competency, we envision the usage of our method to be three different scenarios ranging on the effectiveness of those contributions. These scenarios are the situation where our automated TD monitoring strategy selection effectively works only 1 per cent, 50 per cent and 90 per cent of the time. To answer RQ1, we measure the total cost for an automated TD monitoring strategy as follows:

$$C(TDM_x, d_{auto}) = (C(TDM_x, d_{hu}) - P(e)) \times C(TDM_x, d_{hu}) \tag{5.17}$$

Whereas $P(e)$ is the probability of our automated TD monitoring strategy selection effectively (as 0.01, 0.5 and 0.9). This equation is utilised to calculate the hypothetical total cost for our automated TD monitoring strategy with different degrees of effectiveness when comparing with the baseline.

RQ2 can be answered when one evaluates the capability of monitoring decisions, in which we measure our methods and the baseline via the global utility scores. To reduce practitioners' assumptions or rationales, we standardise the global utility score to represent the ability to capture severe TD. For td_i, the TDS score can be calculated as:

$$TDS(td_i) = Ef(td_i) \times Sv(td_i) \tag{5.18}$$

Where $Ef(td_i)$ and $Sv(td_i)$ are the total effort to mitigate td_i and severity of td_i respectively. To answer RQ3, we measure the similarity of weighting sets compared to the ground truth. The premise is that a weighting set, which is more similar to the ground truth, can represent the contextual importance of TD better than the

other. We measure the similarity of the weighting set with the Jaccard index. For weighting set W_a and W_b, it can be formally expressed as:

$$J(W_a, W_b) = \frac{W_a \cup W_b}{W_a \cap W_b} \tag{5.19}$$

We set the Functionality weighting as the ground truth for our evaluation. This weighting set represents the contextual importance of components when considering non-technical aspects of components (e.g. customer requirements, business opportunities, etc.), which can be vastly different from considering them from a purely technical perspective. Hence, the Functionality weighting set is most suitable to establish this weighting set as the ground truth for our evaluation. The weighting set that represents our approach is the architectural weighting set, as it can contain information outside of the data set (i.e. software architecture). The architectural weighting set is based on the number of operating components. The weighting set of the baseline is the severity-based weighting set. The severity-based weighting set is derived from the author-defined severity of each component. The first two weighting sets for Apache Accumulo and Apache Cocoon are based on [8] and [7] accordingly.

Equation 5.16 and 5.17 are utilised in the cost analysis of samples from our method and samples from the baseline to answer RQ1. We use Equation 5.18 in the TDS analysis of samples from our method and samples from the baseline and thus answering RQ2. RQ3 is answered by the similarity analysis of the weighting set of samples from our method and samples from the baseline, which is driven by Equation 5.19.

Since the normality of all samples cannot be assumed, we select Mann-Whitney U tests to answer our research questions with a 0.05 level of significance. Specifically, the One-tailed Mann-Whitney U test is utilised as a statistical hypothesis test for

Table 5.6: Mann-Whitney U Hypothesis Test Result of Cost Analysis Samples

Benchmark Scenarios	Description	Effectiveness of Automated Strategy Selection					
		Accumulo			Cocoon		
		0.01	0.5	0.9	0.01	0.5	0.9
S1	MWU stat	28954578	28952485	28952485	2968992	2968992	2968992
	asymptotic(p-value)	0	0	0	0	0	0
S2	MWU stat	39973176.5	39971084	39971084	4109067	4109067	4109067
	asymptotic(p-value)	1	1	1	1	1	1
S3	MWU stat	51228677.5	51228678	51226585	5250831	5250831	5250831
	asymptotic(p-value)	1	1	1	1	1	1

answering RQ1. The null hypothesis of RQ1 one-tailed Mann-Whitney U is the random samples of cost analysis from our method are **not less than** the random samples of cost analysis from the baseline. For RQ2, the null hypothesis of one-tailed Mann-Whitney U is the random samples from TDS analysis of our method are **not greater than** the random samples of TDS analysis from the baseline. Lastly, the null hypothesis of Mann-Whitney U is the random samples of the Jaccard index between the ground truth, and our method are **not greater than** the random samples of the Jaccard index between the ground truth and the baseline.

5.7.2 Research Question 1

Table 5.7 shows the statistical properties of samples from our approach against samples from the baseline (the first three rows). From the table, our approach reports almost 30 per cent less cost than the baseline in the TD critical scenario for all cases. Table 5.6 illustrates the Mann-Whitney U test result of samples from our method against the baseline. The table indicates that there is enough evidence to reject the null hypothesis (p-value less than 0.05).

To answer RQ1: We conclude that our approach statistically produces decisions that cost less than the baseline in TD critical scenarios, which can cost less than down to 30 per cent.

Table 5.7: Statistical Comparison of Our Approach and Traditional Approach

Average Percentage Change	Benchmark Scenarios	Effectiveness of Automated Strategy Selection					
		Accumulo			Cocoon		
		0.01	0.5	0.9	0.01	0.5	0.9
Average Percentage Change - Cost	S1	-0.33013	-16.5063	-29.7114	-0.3330044	-16.6502	-29.9704
	S2	48.78903	24.64092	4.928185	49.4267751	24.96302	4.992604
	S3	199.624	150.9957	111.2992	198.85355	149.926	109.9852
Average Percentage Change - TDS	S1	-0.70626	-0.35669	-0.07134	-6.5553207	-3.31077	-0.66215
	S2	0.007185	0.359258	0.646664	0.07091074	3.545537	6.381967
	S3	1.168644	1.524805	1.815549	9.29115918	13.08593	16.1837

5.7.3 Research Question 2

Table 5.8: Mann-Whitney U Hypothesis Test Result of TDS Score Samples

Benchmark Scenarios	Description	Effectiveness of Automated Strategy Selection					
		Accumulo			Cocoon		
		0.01	0.5	0.9	0.01	0.5	0.9
S1	MWU stat	33392723.5	33392724	34308182	3425292	3425292	3508642
	asymptotic(p-value)	1	1	1	1	1	1
S2	MWU stat	39966836.5	39966837	39966837	4109067	4109067	4109067
	asymptotic(p-value)	0	0	0	0	0	0
S3	MWU stat	48967816.5	48967817	48967817	5022343	5022343	5022343
	asymptotic(p-value)	0	0	0	0	0	0

Answers of RQ2 can be incorporated by Table 5.7 (the last three rows), where our approach reports the ability to capture a TDS score more than the baseline in the neutral and optimistic scenario. For the Apache Cocoon case, the table reports that it can capture TDS score up to 18 per cent more than those two scenarios. The Mann Whitney U test result of samples from our method against the baseline scenarios (Table 5.8) indicates that there is enough statistical evidence to reject the null hypothesis (p-value less than 0.05).

To answer RQ2: We conclude that our approach statistically produces decisions that are more effective than the baseline in TD neutral and optimistic scenarios, which can store more TDS scores up to 18 per cent.

5.7.4 Research Question 3

Table 5.9 indicates statistical performance in samples of the Jaccard index between the ground truth and our method against samples of the Jaccard index between the ground truth and the baseline. According to the table, there is enough statistical significance to reject the null hypothesis.

To answer RQ3: We conclude that our approach is statistically capable of producing contextual weighting sets more similar to the baseline.

Table 5.9: Mann-Whitney U Hypothesis Test Result of Jaccard Index Samples

Benchmark Metrics	Description	$J(GroundTruth, OurApproach)$	
		Accumulo	Cocoon
$J(GroundTruth, Baseline)$	MWU stat	256244.5	126059.5
	asymptotic(p-value)	0	0

5.8 Threat to Validity

Threats to internal validity relate to the data set. Although we explicitly indicate differences and analysis of projects from different versions of the data set, we cannot guarantee their correctness as issues of the data set might be uncovered in the future.

We scope our automated method to the identical number of TD between different time points. The introduction, substitution, refactoring or removal of a component can lead to the admission, manifestation or omission of TD, which can fluctuate the TD number in the backlog. We argue that these incidents require re-instantiation of early TDM activities (e.g. TD identification, measurement, prioritisation, repayment), which inevitably lead to manual intervention by TD analysts and practitioners.

We utilise the corresponding project documentation in the functional and archi-

tectural weighting sets. As those documents can be subject to interpretation and documentation changes in the future, the replicated weighting sets might be different from the ones which we include in our repository. However, it is believed that the project documentation is one of the most reliable sources of the projects, as it is a centralised source of information as well as those project websites can be traced, archived, and/or recalled in the future.

To preserve our evaluation objective on reproducibility, we utilise information and evidence which is directly related to the project or the data set. Due to the limited information provided by the data set, the calculation of the TDS score is based on available parameters, and it is hypothetical. However, TDS calculation is based on common software metrics in the TD community (i.e. TD principal and TD interest). Thus, we believe the overall result in the real world, which extends from our TDS score calculation, can be similar to the result presented in the paper. Identified TD in this paper is categorised by its manifested components. This is due to the absence of unique identification of TD on a finer detail (lines, comments, etc.) in the data set.

It must be noted that due to a large number of samples, we use an asymptotic (approximated) p-value to incorporate the Mann-Whitney U tests instead of the traditional method (i.e. a lookup table). This is because the traditional method is compatible with a number of samples less than 30.

The threat to external validity is related to the interpretation of our evaluation result. It should be noted that our evaluation omits the evaluation of Behavioural Cloning. This is because the effectiveness of Behavioural Cloning is subject to monitoring conditions defined by practitioners, which can vary. Hence, we evaluate our method by assuming different degrees of effectiveness from the outcomes of Behavioural Cloning instead.

5.9 Conclusion

The novel contribution of this paper is the investigation of the characteristics which require Skin in the Game during TDM decision-making. We have explored the presence of asymmetric information, asymmetric payoff, intolerant minority and tolerant majority in TD identification, measurement, prioritisation and monitoring. We present a systematic method to alleviate asymmetric information and asymmetric payoff in TD monitoring. The method utilised the concept of weighting set, percentage change, Replicator Dynamics and Behavioural Cloning.

The weighting set is used to elicit and indicate the significance of each monitoring TD. The percentage change is calculated to represent and determine conditions when monitoring TD can be ignored, when automated TD monitoring decision selection is to be initiated, and when human intervention is required. Replicator Dynamics is applied to select archived strategies when an automated monitoring strategy selection is initiated. Behavioural Cloning is utilised to mimic monitoring conditions from practitioners and to track monitoring conditions.

Statistical analysis and Mann-Whitney U are utilised to evaluate our method. The evaluation result indicates that our method is capable of producing cost-effective contextual TD monitoring decisions, which mitigate asymmetric information and asymmetric payoff between different factions of decision-makers.

There are potential overhead and learning curves in adapting our automated method. Specifically, practitioners may need to understand the techniques we utilised in our method, find appropriate monitoring conditions, archive weighting sets for Replicator Dynamics and train Behaviour Cloning on their own. These are contributing factors to the effectiveness of our method, which we aim to minimise in our future work. One of the future routes includes investigating appropriate moni-

toring conditions and possible weighting sets based on historical data. The end goal of this route is to present a supporting toolset with templates based on historical data. This route can substitute a prerequisite in understanding techniques used in our method to a lower one: understanding the toolset and templates. Nevertheless, we argue that the benefit of our method at this stage outweighs its cost as our method provides a route to verify transparency, fairness and feasibility of the imposing TD monitoring approach, those of which were previously infeasible during decision-making.

Based on our evaluation results, our method generally performs worse in a project with a higher number of TD (i.e. Accumulo case). However, its cause is still inconclusive as our method may suffer from the coarse-grained weighting set, which renders TD less distinguishable from each other, or it is because our weighting set and the ground truth are considered in different perspectives, and these differences are more prominent with a higher number of TD. One of our future works is to closely investigate this phenomenon by observing our method in the industrial setting. We expect to utilise a more fine-grained weighting set elicited from different stakeholders, incorporating our investigation on the usefulness/applicability of our method in the industrial environment.

Chapter 6

Reflection and Appraisal

6.1 Summarised Answers to Research Questions

In this subsection, reflection on how each individual research question was addressed throughout the thesis is provided. Common limitations and future works that apply to several research questions are further explained in separate subsections below.

RQ 1.1: What are the causes of Information Uncertainty in TDM decision-making?

In Chapter 3, we have investigated possible classes of uncertainty in computerised models. The investigation helps us to understand that Information Uncertainty results from information deficiency of a problem of interest, including TDM decision-making.

The occurrence of Information Uncertainty is not exclusively confined to just the decision-making aspects of TDM. This issue can appear in any part of the TDM lifecycle, whether it is related to the input, process, or output phase. Examples of

these are insufficient data, restricted source code access, technical issues related to source code or data (such as readability or encryption), and so on. Hence, Information Uncertainty during the TDM lifecycle is one limited area in which Information Uncertainty may occur within the overall TDM process.

Furthermore, the perception of Information Uncertainty can vary depending on various factors, such as the expertise and confidence level of different stakeholders involved in TDM. This subjectivity adds an extra layer of complexity to the issue. A more objective manner in identifying Information Uncertainty can benefit in representing and communicating TD suffered from Information Uncertainty. However, as this issue is not a primary research question, further research is necessary to address this issue.

RQ 1.2: What are TDM problems rooted in Information Uncertainty? Which TDM activities are affected by it?

We have elaborated that Information Uncertainty roots in various sub-activities in TD identification and measurement. Information Uncertainty can worsen the quality of TD decisions in the above two TDM activities, resulting in misidentification and/or misestimation of affected TD.

Currently, the study focuses on identifying the consequences and underlying factors of TDM in a straightforward manner by examining their direct results. It aims to clarify the relationship between TDM activities and their outcomes explicitly. However, the impact of combining various TDM techniques and other related factors, such as financial restrictions during information acquisition, restricted access to information or excessive confidence among decision-makers, may also influence the presence of Information Uncertainty, potentially leading to unique effects on TDM decision-making. Further investigation is necessary in this area since it remains

unexplored.

RQ 1.3: How to tackle Information Uncertainty in TDM? What are the benefits of the said mitigation?

We have presented an approach to mitigate Information Uncertainty in TD identification and measurement decision-making. The premise of this approach is to manage Information Uncertainty in a systematic manner. Goal-obstacle and scenario analysis is used to elicit information, which in turn helps TD affected by Information Uncertainty become clearer. The value of the information analysis is utilised to prioritise TD with Information Uncertainty for future mitigation. The evaluation shows that TD decision-making decisions produced by our approach are clearer/more predictable and more reliant on perturbation when compared with the traditional approaches.

At present, the contributions are carried out in a generic setting, that is, an online e-microservices commerce system, assuming a centralised TD backlog. This setting may not apply to all environments, such as closed-loop systems or projects with multiple TD backlogs. It is necessary to conduct field studies and research to test the compatibility of the presented contributions with a broader audience.

Moreover, the evaluation process relies on entropy and TD score metrics, which might not directly translate into contributions' applicability. Further evaluations involving stakeholders such as developers, TD analysts, and experts can help solidify the applicability of the findings. This step can also provide a more concrete understanding of the contributions' practicality in different scenarios.

RQ 2.1: What is the origin of Deep Uncertainty in TD decision-making?

In Chapter 4, we have investigated the level of uncertainty and its characteristics

in decision-making literature. The investigation supports us in locating that Deep Uncertainty results from a conflict in decisions, which is applicable to TDM decision-making.

At present, the concept of Deep Uncertainty in the thesis relies on the definition from the broader field of decision-making, which is not necessarily specific to TDM decision-making. It might be beneficial to investigate and define Deep Uncertainty explicitly within the software engineering context. The outcome could provide a better understanding of when and where Deep Uncertainty appears in TDM decisions and help researchers identify Deep Uncertainty's presence, which may or may not be exclusive to TDM decision-making. This finding will enhance the awareness of Deep Uncertainty's nature and facilitate further research into the concept.

RQ 2.2: How does Deep Uncertainty affect TDM decision-making? Which TDM activities can suffer from Deep Uncertainty?

We have elaborated on potential disagreement in TDM decision-making that can stem from TD identification, measurement and prioritisation. When these issues are improperly mitigated, they support the introduction of a Black Swan event in the above TDM activities.

At present, the study identifies the underlying factors contributing to Deep Uncertainty during TDM decision-making through a rational-based analysis of various TDM activities. A scarcity of empirical evidence can limit the applicability of the findings in real-life development environments. Conducting research in this area can help substantiate the results and raise awareness among practitioners regarding Deep Uncertainty's presence during decision-making in TDM.

The study suggests that a black swan event could be one consequence of Deep

Uncertainty. However, there is still a knowledge gap concerning other possible effects of Deep Uncertainty in TDM decision-making. Research on this topic can help fill the void and contribute to creating a more comprehensive guideline for managing Deep Uncertainty during decision-making in TDM. This, in turn, could be a valuable resource for practitioners looking to mitigate Deep Uncertainty's impact on their projects.

RQ 2.3: How to handle Deep Uncertainty in TDM decision-making? What is the difference between the said mitigation and without using one?

We propose a framework to mitigate Deep Uncertainty in the context of TD prioritisation decision-making. The framework aims to mitigate a conflict in TD rankings by helping decision-makers in producing a set of high-quality homogeneous TD rankings. The evaluation result indicates that our framework improves the quality and homogeneity of TD rankings when comparing a case without our framework.

The proposed framework could encounter significant applicability challenges, primarily due to the scalability difficulties associated with Many Objective Optimisation-based techniques and the dependence on decision-makers in establishing objective functions. This issue might limit the practicality of the framework, which could potentially restrict its usefulness in real development environments.

The choice of algorithm and hardware used in Many Objective Optimisation can result in performance issues and trade-offs. These limitations have been left unresolved, as they are not a primary focus of this thesis. Future research and implementations on possible mitigation to these constraints can increase the applicability of the framework.

RQ 3.1: What are the causes of Erratic Uncertainty? How can Erratic Uncertainty

be manifested in TDM decision-making?

In Chapter 5, we have investigated decision-making asymmetries and their counterparts in the field of TD. This investigation led to the identification of a possible cause of Erratic Uncertainty, and its manifestation, in TDM decision-making.

It should be noted that the list of potential implications from Erratic Uncertainty discussed in this thesis is not comprehensive as it draws solely from one source, specifically in addressing decision-making within the context of TDM. Further investigation into the origins of Erratic Uncertainty could contribute to a more extensive catalogue of its repercussions. Such findings may reveal a wide spectrum of impacts that Erratic Uncertainty can have in TDM, thereby increasing awareness among practitioners and advocating for the implementation of presented Skin in the Game-inspired contributions throughout the project's lifecycle.

RQ 3.2: Which TDM activities are vulnerable to Erratic Uncertainty? What are their implications?

Based on the above investigation, we have explored the possible asymmetries in the domain of TDM decision-making. The exploration supports us in identifying vulnerable TDM activities and possible implications of Erratic Uncertainty in several TDM activities, including identification, measurement, prioritisation and monitoring.

While this thesis identifies four major symptoms that require the application of "Skin in the Game" thinking due to Erratic Uncertainty, it should be noted that these symptoms can be interpreted differently by stakeholders based on factors such as their level of bias, competencies, roles, and access rights. This could potentially hinder the timely identification of the symptoms, highlighting a need for a more ob-

jective and systematic approach to identifying asymmetric information, asymmetric payoff, capable minority, and incapable majority. As this challenge is not directly related to the primary research questions of the thesis, further investigation into this issue is still necessary.

RQ 3.3: How to mitigate implications from Erratic Uncertainty in TDM decision-making? What are the advantages or disadvantages of the mitigation, comparing an approach without one?

We have presented a Skin in the Game-inspired method to alleviate implications from Erratic Uncertainty. This method aims to mitigate asymmetric information and asymmetric payoff under the context of TD monitoring decision-making. It promotes fairness to improve asymmetric information and asymmetric payoff. The evaluation result shows that our method is capable of producing cost-effective and contextual TD monitoring decisions better than an approach without our method.

As the evaluation is based on statistical tests, it should be noted that these results may not directly translate to the effectiveness of the framework in handling Erratic Uncertainty in real-life situations. This is because the effectiveness of contributions related to Erratic Uncertainty can be affected by various human-related factors outside of a controlled research environment, such as the competencies of practitioners and the selection of alternative monitoring plans.

While the contributions are evaluated collectively, there is a noticeable absence of an individual evaluation for the presented guideline. This lack of assessment can create uncertainty regarding the applicability of this concept. Conducting industrial-based evaluations is a promising approach which may better demonstrate the effectiveness of these contributions in real-life situations.

The choice of libraries and components used for Behavioural Cloning and Replicator Dynamics can have a direct impact on the quality of the method's outcomes, but this aspect has not been addressed in the thesis. Conducting comparative studies or benchmarks for different library choices would facilitate practitioners to use these contributions. This research can move us closer to implementing the above-mentioned contributions as a tool.

6.2 Dealing with Information Uncertainty

TD practitioners can suffer from information deficiency throughout TDM. Common practices (e.g. asking an expert, brainstorming or various estimation methods) utilised to remediate information deficiency are subject to one's satisfaction, rendering the need for more information questionable to others with different opinions. Without the acknowledgement of the presence of Information Uncertainty in TDM decision-making, different views on the need for information can be mishandled, and a TDM decision with Information Uncertainty is blindly utilised, contributing to a myopic, biased, anchored TDM decision, which deteriorates the quality of TDM.

This thesis illustrates the presence of Information Uncertainty in TDM, its consequences in selected TDM activities (TD identification and TD measurement), and its mitigation, enabling one to handle Information Uncertainty in a systematic, rational, and falsifiable manner. Fundamentally, our presented approach in mitigation Information Uncertainty consists following tactics, depending on the accessibility of information:

- When information relevant to Information Uncertainty is readily available, utilise available information to reduce Information Uncertainty. Available in-

formation includes accessible software artefacts or evidence which can be acquired without further approval.

- When implicit information relevant to Information Uncertainty exists, elicit the information and utilise them to reduce Information Uncertainty. Implicit information includes acquirable information by project resources, further approval, actions, and/or additional analysis.

- When relevant information to Information Uncertainty is inaccessible or non-acquirable, valuate and prioritise instances that suffered from Information Deficiency. This ranking is regularly updated and utilised in information acquisition when it is possible. The valuation method must be objective and falsifiable to decision-makers.

6.3 Dealing with Deep Uncertainty

Disagreement during TDM decision-making, when it is irresolvable, can result in Deep Uncertainty in TDM decision-making. This can be prevalent with the diversity of decision-makers, where a proposed decision cannot be substituted with the others. We investigate causes, behaviours, and the manifestation of Deep Uncertainty in decision-making on TD identification, TD measurement and TD prioritisation. We have provided a decision-making guideline under the presence of Deep Uncertainty in the above TDM activities. Fundamentally, the guideline emphasises the following decision-making aspects:

- Evidence-based: We encourage the use of evidence or experimentation over narration or claims. Evidence or experimentation minimises tunnelling bias, cherry-picking or confirmation bias in decision-making. Evidence-based decision-

making shifts the focus of decision-making to evidence instead of to decision-makers competency, facilitating an understanding of opposing sides of a disagreement and finding a compromising decision.

- Verifiable: We encourage decision-makers to verify presented evidence during decision-making. Evidence verification can help in probing for silent evidence and confirmation bias. Verifiable evidence can facilitate decision-makers to falsify or improve proposing decisions, potentially leading to a common ground in disagreement.

- Comprehensive: In case narration or claim is unavoidable, it should be based on concrete TDM situations. Limitations of evidence, narration or claim must be collectively acknowledged by decision-makers. The premise of this aspect is to eliminate proposing decisions and evidence that are overly simplistic or unrealistic. In addition, when the limitation of evidence, narration or claim is explicitly stated, it is easier for decision-makers to distinguish between information and interpretation from evidence, narration or claim, which can help in establishing a common ground during decision-making.

Based on these aspects, we present a framework which helps in resolving disagreement in TD prioritisation. Essentially, the presented framework emphasises the following aspects:

- Objectification: The presented framework encourages decision-makers to formulate performance indicators of suitable TD prioritisation approaches as MOEA objectives, rendering the decision verifiable.

- Common ground-driven: The presented framework encourages decision-makers to select generated TD rankings which perform well in different objectives. This selection strategy aims to promote comprehensive TD rankings, where

TD rankings which exhibit good performance on different objectives indicate its robustness on different TD prioritisation approaches.

- Optionality: The presented framework encourages decision-makers to select a set of TD rankings and exercise them as options when there is a change in assumption or environment. Optionality in the presented framework promotes evidence-based decisions by letting decision-makers utilise TD rankings when it is evidently suitable to do so rather than selecting one of them based on predetermined assumptions.

6.4 Dealing with Erratic Uncertainty

Erratic Uncertainty can relate to uncertainty rooted in the intractable unpredictability of agents, systems, their interaction or their combinations. Several factors in TDM, including the nature/behaviour of systems, the interaction between different components, TD properties such as the impact and the growth of TD, or the combination of the above, can be unpredictable, rendering Erratic Uncertainty to be manifested in TDM. In TDM decision-making, a decision with Erratic Uncertainty (e.g. a decision related to unpredictable nature) can contribute to the lack of effective responsibility sharing.

This thesis investigates the nature of Erratic Uncertainty in TDM decision-making, their causes and supporting factors that contribute to a decision with the lack of effective responsibility sharing in TD identification, TD measurement, TD prioritisation and TD monitoring. Overall, we found that a decision with a lack of effective responsibility sharing can exist when a decision is imposed by the minority, having more information and gaining more benefit from the decision than the rest, and when the decision is approved by the majority. We provide a guideline to notice

and mitigate these supporting factors during decision-making. Fundamentally, we emphasise the following aspects:

- Transparency: We promote transparent decision-making. Particularly, we encourage decision-makers to explicitly inform those affected by imposing decisions on their benefits/losses from the decision, preventing uninformed asymmetric payoffs during decision-making.
- Traceable: We recommend that approved decisions should be traceable to the one who imposed them. The premise of this recommendation is to put the responsibility on the imposing parties, preventing asymmetric payoffs, even when they have declared otherwise.
- Participation: We encourage the majority of decision-makers to police an imposing decision even when the decision does not heavily affect them. The premise of this is to prevent decision-makers from imposing a decision that benefits only a specific individual or a class of stakeholders.

These fundamentals are included in our method, which we presented to mitigate a TD monitoring problem, rooted in the lack of effective responsibility sharing. This method is essentially based on the following aspects:

- Clarity: Our method is built on the model where parameters (i.e. monitoring criteria, the weight of TD instance) can be clearly observed and discussed. This aspect facilitates transparent decision-making, where decision-makers and other classes of stakeholders (e.g. developer, maintainer) can participate in the model input.
- Equality: The premise of the model mentioned above is to balance payoffs in different classes of stakeholders during TDM (i.e. workloads between TD

analysts and developers or maintainers), filtering when their involvement is necessary.

- Automation: In case inequality in payoffs of a specific class of stakeholders is unavoidable, we utilise decision-making automation (i.e. Replicator Dynamics and Behavioural Cloning) to reduce the need for that class, avoiding unnecessary payoffs from a decision.

6.5 Applicability of Contributions

Applicability for contributions can relate to the ability of presented contributions to be applied in different TDM activities. For instance, the applicability of a systematic approach to mitigate Information Uncertainty in TD prioritisation, the applicability of a MORDM-based framework to mitigate Deep Uncertainty in TD identification, etc. More specifically, contributions in Chapter 3 are limited to TD identification and measurement, whereas the presented model is limited to goals-obstacle analysis. A framework presented in Chapter 4 is limited to Deep Uncertainty in TD prioritisation, although the decision-making guideline can apply to TD identification, TD measurement and TD prioritisation. Lastly, the guideline in Chapter 5 can apply to TD identification, TD measurement, TD prioritisation and TD monitoring, whereas the presented automated method is specific to a problem from Erratic Uncertainty in TD monitoring.

As there is no computerised model that can be useful without considering a problem context, we notice that some modifications are required when positioning the existing contributions in this thesis to different TDM activities. However, we argue that the foundations of these contributions can withstand different TDM activities. More precisely, we observed that:

- Chapter 3: Mitigation tactics mentioned in 6.2 can apply to other TDM activities without any modification as these three types of information accessibility exist regardless of TDM activities. The expected value of information analysis can value Information Uncertainty in TD regardless of TDM activities as long as practitioners utilise consistent rationales in estimating the potential benefits of TD instances. The presented probabilistic model can be effectively used to indicate the importance of TD in the backlog throughout the TDM lifecycle if practitioners frequently update TD parameters and the obstruction to a system goal.

- Chapter 4: Three objectives of the framework mentioned in 6.3 can apply to other TDM activities without any modification. This is because those objectives are based on characteristics of Deep Uncertainty. The existing framework can fundamentally help in probing Deep Uncertainty in TD identification and TD measurement. For instance, practitioners may suspect a possible Deep Uncertainty in a TD identification model/a TD measurement method when a generated TD ranking and its objective values do not coincide with what they believe to be severe TD in the backlog.

- Chapter 5: Objectives of a presented method raised in 6.4 are applicable to other TDM activities to mitigate the lack of effective responsibility sharing. Although the model is specifically implemented for TD monitoring, the concept of weighting set and percentage change are applicable to the TD model in other TDM activities. A selection of automated techniques utilised in the method (i.e. Behavioural Cloning and Replicator Dynamics) is applicable to problems of the lack of effective responsibility sharing different TDM activities.

It is worth noting that contributions in the thesis are evaluated on the limits of open-source online systems, using the TD backlogs provided by these projects. Fur-

ther field studies and evaluation are required to evaluate their applicability in more specific software development environments such as MLOps, DevOps, and Global Software Engineering team setups, considering the constraints they can pause (e.g., cultural, distribution, management styles, size, etc.) and catalysts they can specifically occur during TD decision making. Potential domains that could benefit from our contributions involve those where uncertainty in TDM activity and decision-making are often dominant, including E-type and open systems, where the magnitude of uncertainty often varies with the diversity of perspectives and skills. Notable examples include software engineering of intelligent systems, cyber-physical systems, and human-centric autonomous and adaptive systems with human-in-/on-the-loop. Our contributions can benefit closed-loop systems (e.g. software engineering for embedded systems, etc.), where uncertainty in TDM decision-making and management can be experienced due to the novelty of the issue, deployment domain, criticality of the issue and its likely impact if the debt is not structured and managed.

Contributions in this thesis are under the assumption of one central TD backlog. Further research and evaluation may be necessary to test the applicability of our contributions in highly decentralised environments, where projects might have multiple TD backlogs, each possibly managed by separate teams (e.g. large-scale microservices architectures and cloud-based software). At the current stage, our contributions can be applied to selective complex scenarios (e.g., interaction scenarios and security, safety-critical scenarios) under investigation, where the consequences of uncertainty in TDM decision-making could be severe.

6.6 Practicality of Contributions

The practicality of contributions can relate to the ability of practitioners to utilise our contributions to solve their real-world problems. Since the goal of this thesis is to investigate uncertainty in TDM decision-making, most of our contributions are in the primitive phase, where we prioritise the customisability of our contribution over practicality.

Most equations of our models are traceable and can be modified to emphasise specific aspects, without affecting the rest of the contributions. For instance, it is possible to introduce an additional parameter in our probabilistic model in Chapter 3 without affecting the scenario and the expected value of information analysis. Many parameters can be modified, depending on the granularity level that practitioners require. The weight of TD in Chapter 5, for example, can be assigned both in a fine-grained (e.g. different TD weight per TD instance) and a coarse-grained fashion (e.g. different TD weight per system components). However, customisability for our contributions can result in the complexity of practitioners' usage, discouraging practitioners from utilising our contributions at this phase. Practitioners might find our model overly complicated and on a steep learning curve. For example, it can be difficult for practitioners to formulate objective functions in Chapter 4 on their own, which discourages them from using the framework. We believe this shortcoming can be remediated when we develop a tool specifically for each contribution. Although this is a part of our future work, we envision the following techniques to overcome the complexity of our contributions:

- Default value: Many parameters in our contributions can be left as default values unless practitioners intend to customise. For example, it is possible to utilise the weight for each quality attribute as stated in Chapter 3 until

practitioners understand the tools and intend to modify them.

- Preset: Many areas in our contributions can be pre-configured, and automatically assigned after receiving feedback/answers from practitioners. In Chapter 4, for example, it is possible to provide a choice of TD prioritisation approaches with templates of objective functions, reducing the need for practitioners to formulate them manually.

The thesis's contributions are conducted on the open-source online environments, where TDM can be managed by stakeholders (e.g. TD analysts, developers, experts, decision-makers, etc.) throughout development and after deployment. Further research and implementation are required to enhance the practicality of our contributions outside of these conditions, where TDM activities can be compromised due to the nature of the development and/or deployment process. Some environments include closed-loop systems, such as software engineering for embedded systems and the Internet of Things, where developers can have challenges in mitigating TD after their deployment. At the current stage, our contributions can be useful to manage crucial TD at the design time of closed-loop systems, where mitigation of TD can be achieved through updating existing components (e.g. firmware, system, programmable procedures, etc.) through a new iteration of the system and/or when the mass recall of systems in order to mitigate severe and crucial TD are justified.

Other domains that require additional research to the practicality of our contributions are outsourced projects or projects with a high stakeholder turnover rate, where maintenance and development of the system after the initial deployment can be conducted by different sets of stakeholders. TDM activities and decision-making in this domain can be inconsistent, and the total cost of adapting our contributions can be high since they require setting up our contributions when there is a turnover of significant/crucial stakeholders. Our current contributions are relevant to projects

with high turnover rates, particularly in safety-first, mission-critical settings, as the total cost of our contributions may outweigh the severe consequences of suboptimal TD monitoring decisions, which resulted from uncertainty in TDM decision-making, contributing to high refactoring, re-writing costs, and/or financial losses.

It is worth noting that evaluations of our contributions are assessed via control experiments. More in-depth studies and applications in real-life scenarios are necessary to confirm the practicality in actual situations/scenarios. The thesis's contributions are foundational works, which are genetic, flexible, and adaptable, facilitating one in implementing them tailored to practitioners' requirements. These include their level of granularity, precision, TD types, organisational and developmental cultures and/or paradigms. The evaluation of the thesis intends to demonstrate the potential usefulness and adaptability of the proposed contributions in a specific practical context. While the experiments provide insights into their behaviour and performance, they do not necessarily reflect other practical concerns, such as cost-effectiveness, compatibility with existing systems, and variations in inputs provided by stakeholders. These essential aspects cannot always be constrained within research environments yet require considerations of these aspects, and their trade-offs are crucial when implementing the solutions in practice.

Chapter 7

Future Directions

7.1 Exploration of Our Contributions in Specific Problem Domain

As the primary aim of this thesis is to bridge the knowledge gap that relates to uncertainty in TDM decision-making, we have built foundational contributions that do not tie to specific TDM problem domains. However, further studies are needed to explore their relevance to various problem domains and how our proposed approaches can be tailored and applied to the said domains. Below, we highlight how our contribution can be leveraged in different areas of TDM decision-making.

7.1.1 Consideration on Specific TD type

Our contributions have considered commonly acknowledged and discussed components of TD (e.g. TD principal, interest amount, etc.) and are not specific to TD

types. This may require customising part of the techniques to address uncertainty in certain types of TD and stages the TD will manifest. For example, one might find it challenging to apply our probabilistic goal-obstacle model to mitigate Information Uncertainty in requirement debts in the early stages of the requirements modelling and in situations where system goals might not have been finalised. However, it can be argued that our work can help requirements engineers to integrate TD under uncertainty practices as part of the requirements engineering processes. To mitigate this, one future investigation will evaluate shortcomings when applying our contributions to specific software engineering stages and their corresponding TD types. An investigation may explore different decision-making techniques which can be relevant to specific phases and can benefit specific TD types when handling uncertainties in TDM decision-making. One possibility is to consider financial aspects (e.g. net present value (NPV), ROI, real options, etc.) to help in prioritising uncertainties manifested in TD types that can be difficult to precisely estimate in a technical aspect (e.g. requirement debt, architectural debt, test debt).

7.1.2 Application to Different TDM activities

We formulated uncertainties in TDM decision-making as a problem, and we investigated its relevance to different TDM activities. The scope of this thesis was limited to selected TDM activities as proof of concept. Further research and investigation are needed to understand the presence of uncertainties, their manifestation in different TDM activities, and their effects throughout the TDM lifecycle. One of several promising research routes is to investigate the presence of Deep Uncertainty in TD monitoring and in relation to monitorable and its criteria. For instance, decision-makers may have to choose TD monitoring criteria to be utilised under project constraints. Different TD monitoring criteria can emphasise different TD aspects,

leading to disagreements between decision-makers on possible Deep Uncertainty in TD monitoring.

Another research direction to consider is the presence of Erratic Uncertainty in TD identification and measurement, as existing identification models and measurement methods might be insufficient to encapsulate the erratic behaviour of the system and its components in TDM and decision-making.

7.2 Improvement of Existing Contributions

Another research directions to consider are related to further refinements and enhancements of our contributions to mitigate their shortcomings/limitations. The directions are discussed by mapping to relevant thesis chapters:

7.2.1 Automated Scenario Categorisation

In Chapter 3 and 4, we leverage requirements engineering techniques to produce the list of scenarios. These techniques rely on practitioners and/or software documentation to profile the system scenarios. One future work to mitigate this is to establish scenarios based on historical data, loosening the reliance on software documentation. Another possibility is to utilise clustering methods such as hierarchical clustering or k-means to profile run-time variables into scenarios. We believe that the application of clustering methods can reduce the dependence on practitioners and software documentation altogether.

7.2.2 Automated Decision Selection

In Chapter 4 and 5, practitioners are required to select a set of decisions by themselves. Chapter 4, decision-makers choose which solutions should be included in the archive and to be utilised when there is an environmental change. Similarly, in Chapter 5, decision-makers are required to select solutions that will later undergo Replicator Dynamics and be employed in scenarios in which automated TD monitoring decisions can be applied. This process can suffer from stakeholder turnover, leading to inconsistencies in the effectiveness of associated contributions. To mitigate this issue, an automated selection approach could be implemented, which would provide a more consistent and systematic way to select candidate solutions while minimising the influence of stakeholder turnovers.

To implement this concept, Reinforcement Learning [52] can be utilised during both the design and runtime phases. During the design phase, practitioners complete a questionnaire regarding their priorities in terms of solution quality (e.g., security, reliability, performance), associated TD parameters (e.g. security rating, interest, SQALE rating, etc.), and common strategies for selecting solutions (e.g. Critical-oriented, cost-effectiveness oriented, etc.). The insights gained from this process help to identify TD parameters/features that can be considered as reward or penalty functions. This information is then used during the runtime phase, where automation offers a selection of solutions or alerts decision-makers when they may not conform to the priority of quality aspects stated prior. Feedback from these prompts can be used to fine-tune penalties and reward functions for future iterations.

7.2.3 Strategies Generation

Our disagreement resolution framework in Chapter 5 relies on MOEAs to generate TD rankings and practitioners to select them. Those factors can be resource-intensive depending on the number of objective functions. One future work is to investigate decision-making techniques/methods to mitigate this shortcoming. Promising methods include a method based on a machine learning framework in a class of generative adversarial network [31] and inverse reinforcement learning [87]. Another technique is to codify our disagreement resolution framework into a mathematical model tailored to a specific problem context. This can be done with the use of historical data and data analysis. We believe that the implementation of strategy generation can reduce practitioners' workload and can improve the practicality of the MOEAs framework.

Bibliography

[1] Z. S. H. Abad and G. Ruhe. "Using real options to manage technical debt in requirements engineering". In: *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE. 2015, pp. 230–235.

[2] N. Ahmadi et al. "A survey of social software engineering". In: *2008 23rd IEEE/ACM International Conference on Automated Software Engineering-Workshops*. IEEE. 2008, pp. 1–12.

[3] S. Akbarinasaji, A. Bener, and A. Neal. "A heuristic for estimating the impact of lingering defects: can debt analogy be used as a metric?" In: *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)*. IEEE. 2017, pp. 36–42.

[4] A. Aldaeej and C. Seaman. "From lasagna to spaghetti, a decision model to manage defect debt". In: *Proceedings of the 2018 International Conference on Technical Debt*. 2018, pp. 67–71.

[5] R. Alfayez et al. "A systematic literature review of technical debt prioritization". In: *Proceedings of the 3rd International Conference on Technical Debt*. 2020, pp. 1–10.

[6] E. Alzaghoul and R. Bahsoon. "CloudMTD: Using real options to manage technical debt in cloud-based service selection". In: *2013 4th International Workshop on Managing Technical Debt (MTD)*. IEEE. 2013, pp. 55–62.

[7] Apache. *Understanding Apache Cocoon*. 2019. URL: https://cocoon.apache.org/2.1/userdocs/concepts/index.html.

[8] Apache. *User Manual Apache Accumulo*. 2022. URL: https://accumulo.apache.org/1.3/user_manual/.

[9] A. Arcuri and L. Briand. "A practical guide for using statistical tests to assess randomized algorithms in software engineering". In: *Proceedings of the 33rd international conference on software engineering*. 2011, pp. 1–10.

[10] K. J. Arrow. "Uncertainty and the welfare economics of medical care". In: *Uncertainty in Economics*. Elsevier, 1978, pp. 345–375.

[11] A. Author(s). *TDMwSKN Repository*. 2022. URL: https://github.com/TDMwSKN/TDMwSKNrepo.

[12] P. Avgeriou et al. "Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162)". In: *In Dagstuhl Reports (Vol. 6,No. 4), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik* (2016).

[13] P. C. Avgeriou et al. "An overview and comparison of technical debt measurement tools". In: *IEEE Software* (2020).

[14] B. M. Ayyub. *Elicitation of expert opinions for uncertainty and risks*. CRC press, 2001.

[15] J. Bailey et al. "Evidence relating to Object-Oriented software design: A survey". In: *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. IEEE. 2007, pp. 482–484.

[16] G. Baio. *Bayesian methods in health economics*. Chapman and Hall/CRC, 2012.

[17] S. Bellomo, R. L. Nord, and I. Ozkaya. "A study of enabling factors for rapid fielding combined practices to balance speed and stability". In: *2013*

35th International Conference on Software Engineering (ICSE). IEEE. 2013, pp. 982–991.

[18] J. Z. Bergman et al. "The shared leadership process in decision-making teams". In: *The Journal of social psychology* 152.1 (2012), pp. 17–42.

[19] T. Besker, A. Martini, and J. Bosch. "Technical debt triage in backlog management". In: *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*. IEEE. 2019, pp. 13–22.

[20] J. Bohnet and J. Döllner. "Monitoring code quality and development activity by software maps". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. 2011, pp. 9–16.

[21] G. E. Box. "Robustness in the strategy of scientific model building". In: *Robustness in statistics*. Elsevier, 1979, pp. 201–236.

[22] N. Brown et al. "Managing technical debt in software-reliant systems". In: *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM. 2010, pp. 47–52.

[23] J. E. Burge et al. "Decision-making in software engineering". In: *Rationale-Based Software Engineering* (2008), pp. 67–76.

[24] S. A. Busari and E. Letier. "Radar: A lightweight tool for requirements and architecture decision analysis". In: *Proceedings of the 39th International Conference on Software Engineering*. IEEE Press. 2017, pp. 552–562.

[25] F. Buschmann. "To pay or not to pay technical debt". In: *IEEE software* 28.6 (2011), pp. 29–31.

[26] A. Cailliau and A. V. Lamsweerde. "A probabilistic framework for goal-oriented risk analysis". In: *2012 20th IEEE International Requirements Engineering Conference (RE)* (2012). DOI: 10.1109/re.2012.6345805.

[27] G. Catolino et al. “Gender diversity and women in software teams: How do they affect community smells?” In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE. 2019, pp. 11–20.

[28] L. Chen, M. A. Babar, and H. Zhang. “Towards an evidence-based understanding of electronic data sources”. In: *14th International conference on evaluation and assessment in software engineering (EASE)*. 2010, pp. 1–4.

[29] C. L. Citroen. “The role of information in strategic decision-making”. In: *International journal of information management* 31.6 (2011), pp. 493–501.

[30] K. Claxton et al. “A pilot study on the use of decision theory and value of information analysis as part of the NHS Health Technology Assessment programme”. In: *Health Technol Assess* 8.31 (2004), pp. 1–103.

[31] A. Creswell et al. “Generative adversarial networks: An overview”. In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[32] W. Cunningham. “The WyCash portfolio management system”. In: *Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum) - OOPSLA 92* (1992). DOI: 10.1145/157709.157715.

[33] R. Darimont and A. Van Lamsweerde. “Formal refinement patterns for goal-driven requirements elaboration”. In: *ACM SIGSOFT Software Engineering Notes*. Vol. 21/6. ACM. 1996, pp. 179–190.

[34] D. Das et al. “Technical debt resulting from architectural degradation and code smells: a systematic mapping study”. In: *ACM SIGAPP Applied Computing Review* 21.4 (2022), pp. 20–36.

[35] D. De Rigo. “Software uncertainty in integrated environmental modelling: the role of semantics and open science”. In: (2013).

[36] R. J. Eisenberg. "A threshold based approach to technical debt". In: *ACM SIGSOFT Software Engineering Notes* 37.2 (2012), pp. 1–6.

[37] A. Ekárt and S. Z. Németh. "Stability analysis of tree structured decision functions". In: *European Journal of Operational Research* 160.3 (2005), pp. 676–695.

[38] Z. Gong and F. Lyu. *Technical debt management in a large-scale distributed project: an Ericsson case study*. 2017.

[39] GuidingMetrics. *The E-Commerce Industry's 12 Most Critical Metrics*. `https://guidingmetrics.com/content/ecommerce-industry-most-critical-metrics-kpis/`, last accessed on 25/11/20. 2019.

[40] Y. Guo and C. Seaman. "A portfolio approach to technical debt management". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. ACM. 2011, pp. 31–34.

[41] Y. Guo, C. Seaman, and F. Q. da Silva. "Costs and obstacles encountered in technical debt management–A case study". In: *Journal of Systems and Software* 120 (2016), pp. 156–169.

[42] Y. Guo, R. O. Spínola, and C. Seaman. "Exploring the costs of technical debt management – a case study". In: *Empirical Software Engineering* 21.1 (2014), pp. 159–182. DOI: `10.1007/s10664-014-9351-7`.

[43] Y. Guo et al. "Tracking technical debt—An exploratory case study". In: *2011 27th IEEE International Conference on Software Maintenance (ICSM)*. IEEE. 2011, pp. 528–531.

[44] M. Harman and B. F. Jones. "Search-based software engineering". In: *Information and software Technology* 43.14 (2001), pp. 833–839.

[45] R. A. Howard. "Information value theory". In: *IEEE Transactions on systems science and cybernetics* 2.1 (1966), pp. 22–26.

[46] P. Hsia et al. “Formal approach to scenario analysis”. In: *IEEE Software* 11.2 (1994), pp. 33–41.

[47] Z. Hu and W. Wu. “A game theoretic model of software crowdsourcing”. In: *2014 IEEE 8th International Symposium on Service Oriented System Engineering*. IEEE. 2014, pp. 446–453.

[48] M. Imranur, S. Panichella, D. Taibi, et al. “A curated dataset of microservices-based systems”. In: *arXiv preprint arXiv:1909.03249* (2019).

[49] A. Indrayan and R. K. Malhotra. *Medical biostatistics*. Chapman and Hall/CRC, 2017.

[50] C. Izurieta et al. “On the uncertainty of technical debt measurements”. In: *2013 International Conference on Information Science and Applications (ICISA)*. IEEE. 2013, pp. 1–4.

[51] C. Izurieta et al. “A position study to investigate technical debt associated with security weaknesses”. In: *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*. IEEE. 2018, pp. 138–142.

[52] L. P. Kaelbling, M. L. Littman, and A. W. Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[53] J. R. Kasprzyk et al. “Many objective robust decision making for complex environmental systems undergoing change”. In: *Environmental Modelling & Software* 42 (2013), pp. 55–71.

[54] A. Katin et al. “On the Technical Debt Prioritization and Cost Estimation with SonarQube tool”. In: *18th International Scientific Conference on Industrial Systems - Industrial Innovation in Digital Age* (2020).

[55] R. Kazman, M. Klein, and P. Clements. *ATAM: Method for architecture evaluation*. Tech. rep. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2000.

[56] M. C. Kennedy and A. O'Hagan. "Bayesian calibration of computer models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.3 (2001), pp. 425–464.

[57] B. Kitchenham and S. Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007).

[58] T. Klinger et al. "An enterprise perspective on technical debt". In: *Proceedings of the 2nd Workshop on managing technical debt*. 2011, pp. 35–38.

[59] G. J. Klir. "Uncertainty and information". In: *Foundations of Generalized Information Theory* (2006).

[60] F. H. Knight. *Risk, uncertainty and profit*. Vol. 31. Houghton Mifflin, 1921.

[61] P. Kruchten, R. L. Nord, and I. Ozkaya. "Technical debt: From metaphor to theory and practice". In: *Ieee software* 29.6 (2012), pp. 18–21.

[62] P. Kruchten, R. L. Nord, and I. Ozkaya. "Technical Debt: From Metaphor to Theory and Practice". In: *IEEE Software* 29.6 (2012), pp. 18–21. DOI: 10.1109/ms.2012.167.

[63] O. Ktata and G. Lévesque. "Designing and Implementing a Measurement Program for Scrum Teams: What do agile developers really need and want?" In: *Proceedings of the Third C* Conference on Computer Science and Software Engineering*. 2010, pp. 101–107.

[64] A. Lapouchnian. "Goal-oriented requirements engineering: An overview of the current research". In: *University of Toronto* 32 (2005).

[65] L. Lavazza, S. Morasca, and D. Tosi. "Technical debt as an external software attribute". In: *Proceedings of the 2018 International Conference on Technical Debt*. 2018, pp. 21–30.

[66] R. J. Lempert. *Shaping the next one hundred years: new methods for quantitative, long-term policy analysis*. Rand Corporation, 2003.

[67] R. J. Lempert, M. E. Schlesinger, and S. C. Bankes. "When we don't know the costs or the benefits: Adaptive strategies for abating climate change". In: *Climatic Change* 33.2 (1996), pp. 235–274.

[68] V. Lenarduzzi, N. Saarimäki, and D. Taibi. "The Technical Debt Dataset". In: *15th Conference on Predictive Models and Data Analytics in Software Engineering*. Jan. 2019.

[69] V. Lenarduzzi et al. "A systematic literature review on technical debt prioritization: Strategies, processes, factors, and tools". In: *Journal of Systems and Software* 171 (2021), p. 110827.

[70] E. Letier, D. Stefan, and E. T. Barr. "Uncertainty, risk, and information value in software requirements and architecture". In: *Proceedings of the 36th International Conference on Software Engineering*. ACM. 2014, pp. 883–894.

[71] J.-L. Letouzey. "The SQALE method for evaluating technical debt". In: *2012 Third International Workshop on Managing Technical Debt (MTD)*. IEEE. 2012, pp. 31–36.

[72] J.-L. Letouzey and M. Ilkiewicz. "Managing technical debt with the sqale method". In: *IEEE software* 29.6 (2012), pp. 44–51.

[73] J. Lewis and M. Fowler. *Microservices*. `http://www.martinfowler.com/articles/microservices.html`. Mar. 2014.

[74] Z. Li, P. Avgeriou, and P. Liang. "A systematic mapping study on technical debt and its management". In: *Journal of Systems and Software* 101 (2015), pp. 193–220.

[75] Z. Li, P. Avgeriou, and P. Liang. "A systematic mapping study on technical debt and its management". In: *Journal of Systems and Software* 101 (2015), pp. 193–220. DOI: `10.1016/j.jss.2014.12.027`.

[76] E. Lim, N. Taksande, and C. Seaman. "A balancing act: What software practitioners have to say about technical debt". In: *IEEE software* 29.6 (2012), pp. 22–27.

[77] V. Madhivanan and Z. Godandapani. "Technical debt reduction using epsilon-Nash equilibrium for the perturbed software refactor game model". In: *International Journal of Information and Decision Sciences* 13.3 (2021), pp. 275–295.

[78] R. Marinescu. "Assessing technical debt by identifying design flaws in software systems". In: *IBM Journal of Research and Development* 56.5 (2012), pp. 9–1.

[79] A. Martini, T. Besker, and J. Bosch. "Process debt: A first exploration". In: *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE. 2020, pp. 316–325.

[80] A. Martini, J. Bosch, and M. Chaudron. "Architecture Technical Debt: Understanding Causes and a Qualitative Model". In: *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications* (2014). DOI: 10.1109/seaa.2014.65.

[81] A. Martini, J. Bosch, and M. Chaudron. "Architecture technical debt: Understanding causes and a qualitative model". In: *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE. 2014, pp. 85–92.

[82] S. McConnell. "Managing technical debt". In: *Construx Software Builders, Inc* (2008), pp. 1–14.

[83] M. Mohan, D. Greer, and P. McMullan. "Technical debt reduction using search based automated refactoring". In: *Journal of Systems and Software* 120 (2016), pp. 183–194.

[84] Monetate. *Monetate Ecommerce Benchmark Report*. `https://get.monetate.com/eq2-2019-benchmark-report/`, last accessed on 25/11/20. 2019.

[85] S. Morris, N. Devlin, and D. Parkin. *Economic analysis in health care*. John Wiley & Sons, 2007.

[86] J. Nash. "Non-cooperative games". In: *Annals of mathematics* (1951), pp. 286–295.

[87] A. Y. Ng, S. Russell, et al. "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2000, p. 2.

[88] A. Nugroho, J. Visser, and T. Kuipers. "An empirical model of technical debt and interest". In: *Proceedings of the 2nd workshop on managing technical debt*. 2011, pp. 1–8.

[89] J. Olsson et al. "Measuring affective states from technical debt: A psychoempirical software engineering experiment". In: *Empirical Software Engineering* 26 (2021), pp. 1–40.

[90] F. Palomba et al. "Beyond technical aspects: How do community smells influence the intensity of code smells?" In: *IEEE transactions on software engineering* 47.1 (2018), pp. 108–129.

[91] N. Paul. *Generative Adversarial Imitation Learning for gym environments*. 2018. URL: `https://github.com/navuboy/gail_gym`.

[92] B. Pérez, D. Correal, and H. Astudillo. "A proposed model-driven approach to manage architectural technical debt life cycle". In: *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*. IEEE. 2019, pp. 73–77.

[93] K. Petersen et al. "Systematic mapping studies in software engineering." In: *Ease*. Vol. 8. 2008, pp. 68–77.

[94] D. A. Pomerleau. "Alvinn: An autonomous land vehicle in a neural network". In: *Advances in neural information processing systems* 1 (1988).

[95] A. Ramirez, J. R. Romero, and S. Ventura. "A survey of many-objective optimisation in search-based software engineering". In: *Journal of Systems and Software* (2018).

[96] L. F. Ribeiro et al. "A strategy based on multiple decision criteria to support technical debt management". In: *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE. 2017, pp. 334–341.

[97] K. Rindell and J. Holvitie. "Security risk assessment and management as technical debt". In: *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE. 2019, pp. 1–8.

[98] N. Rios, M. G. de Mendonca Neto, and R. O. Spínola. "A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners". In: *Information and Software Technology* (2018).

[99] M. Sadatsafavi et al. "Need for speed: an efficient algorithm for calculation of single-parameter expected value of partial perfect information". In: *Value in Health* 16.2 (2013), pp. 438–448.

[100] W. D. Savedoff. "Kenneth Arrow and the birth of health economics". In: *Bulletin of the World Health Organization* 82 (2004), pp. 139–140.

[101] K. Schmid. "A formal approach to technical debt decision making". In: *Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures*. ACM. 2013, pp. 153–162.

[102] P. Schuster and K. Sigmund. "Replicator dynamics". In: *Journal of theoretical biology* 100.3 (1983), pp. 533–538.

[103] C. Seaman and Y. Guo. "Measuring and monitoring technical debt". In: *Advances in Computers*. Vol. 82. Elsevier, 2011, pp. 25–46.

[104] C. Seaman et al. “Using technical debt data in decision making: Potential decision approaches”. In: *2012 Third International Workshop on Managing Technical Debt (MTD)*. IEEE. 2012, pp. 45–48.

[105] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[106] C. A. Siebra et al. “Applying metrics to identify and monitor technical debt items during software evolution”. In: *2014 IEEE International Symposium on Software Reliability Engineering Workshops*. IEEE. 2014, pp. 92–95.

[107] SonarQube. *Metric Definition*. 2020. URL: `https://docs.sonarqube.org/latest/user-guide/metric-definitions`.

[108] D. Stefan and E. Letier. “Supporting sustainability decisions in large organisations”. In: *ICT for Sustainability 2014 (ICT4S-14)*. Atlantis Press. 2014.

[109] J. E. Stiglitz. “The contributions of the economics of information to twentieth century economics”. In: *The quarterly journal of economics* 115.4 (2000), pp. 1441–1478.

[110] A. Sutcliffe. “Scenario-based requirements engineering”. In: *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.* IEEE. 2003, pp. 320–329.

[111] A. G. Sutcliffe et al. “Supporting scenario-based requirements engineering”. In: *IEEE Transactions on software engineering* 24.12 (1998), pp. 1072–1088.

[112] N. N. Taleb. *The black swan: The impact of the highly improbable*. Vol. 2. Random House Incorporated, 2007.

[113] N. N. Taleb. *Antifragile: Things that gain from disorder*. Vol. 3. Random House Incorporated, 2012.

[114] N. N. Taleb. *Skin in the game: Hidden asymmetries in daily life*. Random House Trade Paperbacks, 2020.

[115] D. A. Tamburri. "Software architecture social debt: Managing the incommunicability factor". In: *IEEE Transactions on Computational Social Systems* 6.1 (2019), pp. 20–37.

[116] D. A. Tamburri, R. Kazman, and H. Fahimi. "The architect's role in community shepherding". In: *IEEE Software* 33.6 (2016), pp. 70–79.

[117] D. A. Tamburri, F. Palomba, and R. Kazman. "Exploring community smells in open-source: An automated approach". In: *IEEE Transactions on software Engineering* 47.3 (2019), pp. 630–652.

[118] D. A. Tamburri et al. "Social debt in software engineering: insights from industry". In: *Journal of Internet Services and Applications* 6.1 (2015), p. 10.

[119] T. Team. *TDMuUI Repository*. 2023. URL: `https://github.com/TDMuUI/TDMuUI`.

[120] Z. Tong et al. "Greening Software Requirements Change Management Strategy Based on Nash Equilibrium". In: *Wireless Communications and Mobile Computing* 2017 (2017).

[121] A.-A. Tsintzira et al. "Applying machine learning in technical debt management: Future opportunities and challenges". In: *International Conference on the Quality of Information and Communications Technology*. Springer. 2020, pp. 53–67.

[122] H. Tu and T. Menzies. "DebtFree: minimizing labeling cost in self-admitted technical debt identification using semi-supervised learning". In: *Empirical Software Engineering* 27.4 (2022), pp. 1–37.

[123] A. Van Lamsweerde. "Goal-oriented requirements engineering: A guided tour". In: *Proceedings fifth ieee international symposium on requirements engineering*. IEEE. 2001, pp. 249–262.

[124] A. Van Lamsweerde and E. Letier. "Handling obstacles in goal-oriented requirements engineering". In: *IEEE Transactions on software engineering* 26.10 (2000), pp. 978–1005.

[125] S. H. Vathsavayi and K. Systä. "Technical Debt Management with Genetic Algorithms". In: *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE. 2016, pp. 50–53.

[126] W. E. Walker, R. J. Lempert, and J. H. Kwakkel. "Deep uncertainty". In: *Delft University of Technology* 1 (2012), p. 2.

[127] W. E. Walker et al. "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support". In: *Integrated assessment* 4.1 (2003), pp. 5–17.

[128] M. Weber. "Decision making with incomplete information". In: *European journal of operational research* 28.1 (1987), pp. 44–57.

[129] R. Wieringa et al. "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion". In: *Requirements engineering* 11 (2006), pp. 102–107.

[130] C. Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10.

[131] WordStream. *Google Ads Benchmarks for Your Industry*. `https://www.wordstream.com/blog/ws/2016/02/29/google-adwords-industry-benchmarks`, last accessed on 25/11/20. 2019.

[132] L. Xiao et al. "Identifying and quantifying architectural debt". In: *Proceedings of the 38th International Conference on Software Engineering*. ACM. 2016, pp. 488–498.

[133] J. Yli-Huumo, A. Maglyas, and K. Smolander. "How do software development teams manage technical debt?–An empirical study". In: *Journal of Systems and Software* 120 (2016), pp. 195–218.

[134] F. Yokota and K. M. Thompson. "Value of information literature analysis: a review of applications in health risk management". In: *Medical Decision Making* 24.3 (2004), pp. 287–298.

[135] F. Zampetti, A. Serebrenik, and M. Di Penta. "Was self-admitted technical debt removal a real removal? an in-depth perspective". In: *Proceedings of the 15th international conference on mining software repositories*. 2018, pp. 526–536.

[136] N. Zazworka, C. Seaman, and F. Shull. "Prioritizing design debt investment opportunities". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. ACM. 2011, pp. 39–42.

[137] P. Zeephongsekul and C. Chiera. "Optimal software release policy based on a two-person game of timing". In: *Journal of applied probability* 32.2 (1995), pp. 470–481.

[138] Y. Zhang, M. Harman, and A. Mansouri. *The SBSE Repository: A repository and analysis of authors and research articles on Search Based Software Engineering*. `http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/`.

Appendix A

Selected Studies in SLR

[S1] I. Gat and M. Coté. "Implications of Technical Debt Uncertainty for Software Licensing Negotiations". In: (2010). URL: https://theagileexecutive.com/2010/10/19/implications-of-technical-debt-uncertainty-for-software-licensing-negotiations/.

[S2] A. Omeyer. "What is Technical Debt in Agile?" In: (2019). URL: https://stepsize.com/blog/broader-definition-of-technical-debt/.

[S3] P. Golofit. "Technical Debt - the silent villain of web development". In: (2020). URL: https://accesto.com/blog/technical-debt-the-silent-villain-of-web-development/.

[S4] S. Johann and E. Wolff. "Managing Technical Debt". In: (2013). URL: https://www.infoq.com/articles/managing-technical-debt/.

[S5] D. O'Neill. "Technical debt in the code". In: *Defense aquisition, technology and logistics, XLII No* 2 (2013), pp. 35–38.

[S6] L. Gautam. "The Hidden Cost Of Technical Debt". In: (2020). URL: https://medium.com/serious-scrum/the-hidden-cost-of-technical-debt-1963b958e5ed.

[S7] C. Izurieta et al. "On the uncertainty of technical debt measurements". In: *2013 International Conference on Information Science and Applications (ICISA)*. IEEE. 2013, pp. 1–4.

[S8] Z. S. H. Abad and G. Ruhe. "Using real options to manage technical debt in requirements engineering". In: *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE. 2015, pp. 230–235.

[S9] Y. Guo and C. Seaman. "A portfolio approach to technical debt management". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. 2011, pp. 31–34.

[S10] N. Brown et al. "Managing technical debt in software-reliant systems". In: *Proceedings of the FSE/SDP workshop on Future of software engineering research*. 2010, pp. 47–52.

[S11] Y. Guo, C. Seaman, and F. Q. da Silva. "Costs and obstacles encountered in technical debt management–A case study". In: *Journal of Systems and Software* 120 (2016), pp. 156–169.

[S12] Y. Guo et al. "Tracking technical debt—An exploratory case study". In: *2011 27th IEEE international conference on software maintenance (ICSM)*. IEEE. 2011, pp. 528–531.

[S13] D. Falessi et al. "Practical considerations, challenges, and requirements of tool-support for managing technical debt". In: *2013 4th International Workshop on Managing Technical Debt (MTD)*. IEEE. 2013, pp. 16–19.

[S14] Y. Guo, R. O. Spínola, and C. Seaman. "Exploring the costs of technical debt management–a case study". In: *Empirical Software Engineering* 21 (2016), pp. 159–182.

[S15] A. Martini, J. Bosch, and M. Chaudron. "Architecture technical debt: Understanding causes and a qualitative model". In: *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE. 2014, pp. 85–92.

[S16] E. Tom, A. Aurum, and R. Vidgen. "An exploration of technical debt". In: *Journal of Systems and Software* 86.6 (2013), pp. 1498–1516.

[S17] E. Alzaghoul and R. Bahsoon. "CloudMTD: Using real options to manage technical debt in cloud-based service selection". In: *2013 4th International Workshop on Managing Technical Debt (MTD)*. IEEE. 2013, pp. 55–62.

[S18] C. Fernández Sánchez et al. "A Cost-Benefit analysis model for technical debt management considering uncertainty and time". In: (2013).

[S19] E. Alzaghoul and R. Bahsoon. "Evaluating technical debt in cloud-based architectures using real options". In: *2014 23rd Australian Software Engineering Conference*. IEEE. 2014, pp. 1–10.

[S20] T. Chen et al. "To adapt or not to adapt? technical debt and learning driven self-adaptation for managing runtime performance". In: *Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering*. 2018, pp. 48–55.

[S21] A. Martini, J. Bosch, and M. Chaudron. "Investigating architectural technical debt accumulation and refactoring over time: A multiple-case study". In: *Information and Software Technology* 67 (2015), pp. 237–253.

[S22] K. Rindell and J. Holvitie. "Security risk assessment and management as technical debt". In: *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE. 2019, pp. 1–8.

[S23] A. Martini and J. Bosch. "The danger of architectural technical debt: Contagious debt and vicious circles". In: *2015 12th Working IEEE/IFIP Conference on Software Architecture*. IEEE. 2015, pp. 1–10.

[S24] T. T. Ho and G. Ruhe. "When-to-release decisions in consideration of technical debt". In: *2014 Sixth International Workshop on Managing Technical Debt*. IEEE. 2014, pp. 31–34.

[S25] M. A. de Freitas Farias et al. "Identifying self-admitted technical debt through code comment analysis with a contextualized vocabulary". In: *Information and Software Technology* 121 (2020), p. 106270.

[S26] E. Alzaghoul and R. Bahsoon. "Economics-driven approach for managing technical debt in cloud-based architectures". In: *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE. 2013, pp. 239–242.

[S27] B. Vogel-Heuser and S. Røsch. "Applicability of technical debt as a concept to understand obstacles for evolution of automated production systems". In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE. 2015, pp. 127–132.

[S28] J. Olsson et al. "Measuring affective states from technical debt: A psychoempirical software engineering experiment". In: *Empirical Software Engineering* 26 (2021), pp. 1–40.

[S29] A. Martini and J. Bosch. "On the interest of architectural technical debt: Uncovering the contagious debt phenomenon". In: *Journal of Software: Evolution and Process* 29.10 (2017), e1877.

[S30] L. Rantala. "Towards better technical debt detection with nlp and machine learning methods". In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*. 2020, pp. 242–245.

[S31] R. Verdecchia et al. "Building and evaluating a theory of architectural technical debt in software-intensive systems". In: *Journal of Systems and Software* 176 (2021), p. 110925.

[S32] M. Albarak et al. “Managing technical debt in database normalization”. In: *IEEE Transactions on Software Engineering* 48.3 (2020), pp. 755–772.

[S33] T. Besker, A. Martini, and J. Bosch. “Technical debt triage in backlog management”. In: *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*. IEEE. 2019, pp. 13–22.

[S34] D. Tsoukalas et al. “Technical debt forecasting: an empirical study on open-source repositories”. In: *Journal of Systems and Software* 170 (2020), p. 110777.

[S35] G. K. Hanssen, G. Brataas, and A. Martini. “Identifying scalability debt in open systems”. In: *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*. IEEE. 2019, pp. 48–52.

[S36] S. Kumar et al. “Identifying and estimating technical debt for service composition in SaaS cloud”. In: *2019 IEEE International Conference on Web Services (ICWS)*. IEEE. 2019, pp. 121–125.

[S37] L. B. Foganholi et al. “Supporting technical debt cataloging with TD-Tracker tool”. In: *Advances in Software Engineering* 2015 (2015), pp. 4–4.

[S38] A. Martini and J. Bosch. “Architectural technical debt in embedded systems”. In: *Systems Engineering in the Fourth Industrial Revolution* (2019), pp. 77–103.

[S39] M. A. Al Mamun, C. Berger, and J. Hansson. “Explicating, understanding, and managing technical debt from self-driving miniature car projects”. In: *2014 sixth international workshop on managing technical debt*. IEEE. 2014, pp. 11–18.

[S40] L. Lavazza, S. Morasca, and D. Tosi. “Technical debt as an external software attribute”. In: *Proceedings of the 2018 International Conference on Technical Debt*. 2018, pp. 21–30.

[S41] A. Aldaeej and C. Seaman. "From lasagna to spaghetti, a decision model to manage defect debt". In: *Proceedings of the 2018 International Conference on Technical Debt*. 2018, pp. 67–71.

[S42] S. Hacks et al. "Towards the definition of enterprise architecture debts". In: *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW)*. IEEE. 2019, pp. 9–16.

[S43] B. Ojameruaye and R. Bahsoon. "Systematic elaboration of compliance requirements using compliance debt and portfolio theory". In: *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings 20*. Springer. 2014, pp. 152–167.

[S44] C. Mera-Gómez et al. "Elasticity debt: a debt-aware approach to reason about elasticity decisions in the cloud". In: *Proceedings of the 9th International Conference on Utility and Cloud Computing*. 2016, pp. 79–88.

[S45] M. Al-Barak and R. Bahsoon. "Database design debts through examining schema evolution". In: *2016 IEEE 8th International Workshop on Managing Technical Debt (MTD)*. IEEE. 2016, pp. 17–23.

[S46] C. Petrozzino. "Who pays for ethical debt in AI?" In: *AI and Ethics* 1.3 (2021), pp. 205–208.

[S47] A. Martini, T. Besker, and J. Bosch. "Process debt: A first exploration". In: *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE. 2020, pp. 316–325.

[S48] B. Ojameruaye, R. Bahsoon, and L. Duboc. "Sustainability debt: A portfolio-based approach for evaluating sustainability requirements in architectures". In: *Proceedings of the 38th International Conference on Software Engineering Companion*. 2016, pp. 543–552.

[S49] Y. C. Yeong, S. Hacks, and H. Lichter. "Prioritization of EA debts facilitating portfolio theory". In: *QuASoQ@ APSEC*. 2019, pp. 45–52.

[S50] N. Ramasubbu and C. F. Kemerer. "Controlling technical debt remediation in outsourced enterprise systems maintenance: an empirical analysis". In: *Journal of Management Information Systems* 38.1 (2021), pp. 4–28.

[S51] A. M. Ebrahimi, G. A. Oliva, and A. E. Hassan. "Self-Admitted Technical Debt in Ethereum Smart Contracts: A Large-Scale Exploratory Study". In: *IEEE Transactions on Software Engineering* (2023).

[S52] M. I. Murillo et al. "Common Causes and Effects of Technical Debt in Costa Rica: InsighTD Survey Replication". In: *2021 XLVII Latin American Computing Conference (CLEI)*. IEEE. 2021, pp. 1–9.

[S53] S. S. De Toledo et al. "Accumulation and prioritization of architectural debt in three companies migrating to microservices". In: *IEEE Access* 10 (2022), pp. 37422–37445.

[S54] A. Martini et al. "Process Debt: Definition, Risks and Management". In: *Risks and Management* ().

[S55] M. Ristimaki. "Gradual Rewrite for Reducing Technical Debt". In: (2023). URL: `https://developer.disqo.com/blog/gradual-rewrite-for-reducing-technical-debt/`.

[S56] D. Stolyar. "How to Manage Technical Debt (and Why It Matters)". In: (2021). URL: `https://rubyroidlabs.com/blog/2021/06/how-to-manage-technical-debt-and-why-it-matters/`.

Table A.1: A Table of Selected Studies and their SLR facets

ID	Research Facets	TD activities facets	Contributions Facets	Domain Facets	Uncertainty Facets
S1	Opinion Paper	TD Representation	Method	Software Licensing	Information Deficiency (Lack of technical debt backlog from software vendor)
S2	Opinion Paper	TD Negotiation	Process	Agile	Erratic Behaviour (Ripple Effect)
S3	Experience Report	TD Measurement	Process	Web App	Undefined
S4	Experience Report	TD Measurement	Method	TDM	Undefined
S5	Philosophical Paper	TD Measurement	Other (Classification)	TDM	Undefined
S6	Experience Report	TD Measurement	Method	TDM	Erratic Behaviour (Ripple Effect)
S7	Solution Proposal	TD Measurement	Method	TDM	Information Deficiency (Lack of standardised TD measurement/ground-truth)
S8	Solution Proposal	TD Measurement	Method	TDM	Erratic Behaviour (Customer & Market Requirements)
S9	Solution Proposal	TD Measurement	Process	TDM	Erratic Behaviour (The growth of interest)
S10	Philosophical Paper	TD Measurement	Research Guideline	TDM	Erratic Behaviour (Impact Estimation)
S11	Experience Report	TD Measurement	Report	TDM	Erratic Behaviour (Growth of Interest)
S12	Experience Report	TD Measurement	Report	TDM	Erratic Behaviour (Growth of Interest)
S13	Experience Report	TD Measurement	Report	TDM	Erratic Behaviour (Parameters Quantification)
S14	Experience Report	TD Measurement	Report	TDM	Erratic Behaviour (Parameters Quantification)
S15	Experience Report	TD Identification, TD Measurement	Model	TDM	Information Deficiency (Lack of Clear Documentation), Erratic Behaviour (Ripple Effect)
S16	Experience Report	TD Measurement	Process	TDM	Erratic Behaviour (Risk impact)
S17	Solution Proposal	TD Representation	Method	Cloud	Conflicting Perspectives
S18	Solution Proposal	TD Measurement	Model	TDM	Erratic Behaviour (Growth of Interest)
S19	Solution Proposal	TD Representation	Method	Cloud	Conflicting Perspectives
S20	Solution Proposal	TD Representation	Process	Self-Adaptive System	Conflicting Perspectives
S21	Experience Report	TD Identification, TD Measurement	Taxonomy	TDM	Information Deficiency (Lack of clear documentation), Erratic Behaviour (Ripple Effect)
S22	Solution Proposal	TD Measurement	Process	Security	Erratic Behaviour (Growth of interest & Ripple Effect of Security Risk)
S23	Experience Report	TD Identification, TD Measurement	Taxonomy	TDM	Information Deficiency (Lack of clear documentation), Erratic Behaviour (Ripple Effect)
S24	Solution Proposal	TD Measurement	Model	Software Releases	Information Deficiency (Lack of available data for TD interest quantification), Erratic Behaviour (Complexity)
S25	Experience Report	TD Identification	Report	TDM	Information Deficiency (Uncleared Information)
S26	Solution Proposal	TD Representation	Model	Cloud	Conflicting Perspectives
S27	Experience Report	TD Measurement	Report	Automated production systems	Erratic Behaviour (Ripple Effects)
S28	Experience Report	TD Identification, TD Measurement	Report	TDM	Erratic Behaviour (Ripple Effects)

Table A.2: A Table of Selected Studies and their SLR facets (continued)

S29	Experience Report	TD Measurement	Report	TDM	Erratic Behaviour (Growth of Interest)
S30	Solution Proposal	TD Identification	Other (Hypotheses)	TDM	Conflicting Perspectives
S31	Experience Report	TD Repayment	Report	TDM	Erratic Behaviour (Unpredictability of outcomes)
S32	Solution Proposal	TD Representation	Process	Database	Conflicting Perspectives
S33	Experience Report	TD Prioritisation	Report	TDM	Erratic Behaviour (Future of the software)
S34	Solution Proposal	TD Representation	Model	TDM	Erratic Behaviour (TD Evolution)
S35	Solution Proposal	TD Representation	Process	TDM	Erratic Behaviour (System Scalability)
S36	Solution Proposal	TD Representation	Method	Cloud	Conflicting Perspectives
S37	Solution Proposal	TD Measurement	Process	TDM	Erratic Behaviour (Growth of Interest)
S38	Experience Report	TD Measurement	Report	Embedded Systems	Information Deficiency (Unclear Document), Erratic Behaviour (Growth of Interest)
S39	Experience Report	TD Identification	Report	Self-Driving Automobile	Information Deficiency (Uncleared Information), Erratic Behaviour (Ripple Effect)
S40	Philosophical Paper	TD Repayment	Research Guideline	TDM	Erratic Behaviour (Repayment Return)
S41	Solution Proposal	TD Measurement	Model	TDM	Erratic Behaviour (Ripple Effect)
S42	Philosophical Paper	TD Representation	Other (Research plan)	Enterprise Architecture	Erratic Behaviour (System Complexity)
S43	Solution Proposal	TD Representation	Method	Compliance	Erratic Behaviour (Complexity of Goal Compliance)
S44	Solution Proposal	TD Representation	Tool	Cloud	Erratic Behaviour (Complexity of Elasticity Management)
S45	Philosophical Paper	TD Representation	Taxonomy	Database	Conflicting Perspectives
S46	Opinion Paper	TD Repayment	Other (Discussion)	AI	Erratic Behaviour (Ethical Uncertainty)
S47	Experience Report	TD Representation	Process	Process Debt	Information Deficiency (Uncleared Information)
S48	Solution Proposal	TD Representation	Method	Architectural Sustainability	Information Deficiency (Uncleared Information)
S49	Solution Proposal	TD Representation	Method	Enterprise Architecture	Erratic Behaviour (System Complexity)
S50	Experience Report	TD Representation	Report	Enterprise Systems	Erratic Behaviour (Environmental Uncertainties)
S51	Experience Report	TD Representation	Report	Smart Contracts	Conflicting Perspectives (Design Choices)
S52	Experience Report	TD Identification	Report	TDM	Erratic Behaviour (Time Estimate Ambiguity)
S53	Solution Proposal	TD Representation	Method	Microservices	Conflicting Perspectives (on metadata)
S54	Experience Report	TD Representation	Report	Software Process	Information Deficiency (Lack of clear document/documentation)
S55	Experience Report	TD Identification, TD measurement	Report	TDM	Information Deficiency (Unclear Requirement), Erratic Behaviour (Ripple Effects)
S56	Experience Report	TD Identification	Report	TDM	Information Deficiency (Unclear Requirement)