THE PRIORITIZATION OF TECHNICAL DEBT TYPES IN SOUTH AFRICAN FINANCIAL SERVICES

by

Amos Maphosa

Student number: 295120

A thesis submitted in fulfilment of the requirements for the degree of Master of Business Administration (MBA) by coursework and research in the Wits Business School at the University of the Witwatersrand

Johannesburg, July 2019

Protocol number: WBS/BA295120/585

DECLARATION

I, Amos Ngongolo Maphosa declare that this research project is my own work. The research project is submitted in fulfilment of the requirements for the degree of Master of Business Administration (MBA) by coursework and research in the Wits Business School at the University of the Witwatersrand. The research thesis has not been submitted to any examination or degree at any University. I further declare I have obtained the necessary ethical clearance, title approval and supervisor consent to carry out this research.

Researcher: Amos Maphosa

Date: 29 July 2019

ACKNOWLEDGEMENTS

First, I would like to thank **God** for giving me the opportunity to study MBA at Wits University and undertake this research and complete it to the end, without His blessings this wouldn’t have been possible. Victory belongs to Jesus.

I would like to thank my research supervisor Sandile Selemi. He was always open to assist whatever I needed to bounce ideas and steered me in the right direction.

I would like to thank all the respondents that participated during my pilot study to validate the questions, without their input and time, the validation survey would have not be successfully concluded.

Finally, I would like to thank my late parents for showing me the key to life is hard work and commitment.

SUPPLEMENTARY INFORMATION

Journal: Journal of Systems and Software

Title: A systematic mapping study on technical debt and its management

Authors: ZengyangLi, Paris Avgeriou and Peng Liang

Supervisor/Co-author: Sandile Selemi

Word count: 12 439

Supplementary files:

1. Technical Debt types ANOVA Statistical Output
2. Budget Estimate TD types ANOVA Statistical Output
3. Quality Attribute ANOVA Statistical Output
4. IT Risk Framework variables (ANOVA) Statistical Output
5. Research Questions file
6. Data analysis file
7. Plagiarism form

ABSTRACT

Technical Debt hinders software development projects by reducing the speed of IT teams during development cycles. Although technical debt is assessed on either software artifacts or the entire application, the craftsmanship of IT resources causes the accumulation of technical debt.

In this paper, I investigate two research questions related to the prioritization of technical debt among the respondents in the South African financial services.

The results of 72 recorded respondents were analyzed using a quantitative data analysis approach. The analysis revealed that the Architecture Technical Debt type has the highest level of contribution to Technical Debt with 23.44% of respondents stating so. The analysis further revealed that the System Quality attribute should not be compromised with 17.42% of the respondents attesting to that. Lastly, IT strategy was deemed by 15.04% of the respondents to be the aspect of the IT risk framework to use in prioritizing the Architecture Technical Debt type.

I. INTRODUCTION

The South African financial services industry runs financial applications that are required to be exceptionally secure so that, should they fail, business operations is more likely to be threatened. Incurring debt causes an increase in maintenance obligations and often forces the organisation to relook potential risks that can become a threat to the business (Ramasubbu & Kemerer, 2016).

A. Purpose of study

Financial applications are prone to defects. One of the main failures is the design process. Building complex systems using the Waterfall model implies that as requirements are defined upfront all outstanding development activities are established on initial requirements (Guo,Spínola & Seaman,2014). However, technology changes rapidly, and when development groups deviate from the Waterfall process it gives rise to agile methodologies where changes are embraced, rather than rejected and this results in high rates of change iterations (Codabux & Williams, 2013).

The purpose of this research is firstly, to examine different types of technical debt and how these can be prioritised in the financial services industry. The second purpose is to develop a model to prioritise technical debt types using aspects of the IT risks framework in terms of its budget and quality attributes.

B. Context of the study

Management decisions are important to choose which debt defects to fix, defer or classify since there are multiple dimensions in which to categorise technical debt. According to McConnell (as cited in Snipes, Robinson, Guo, & Seaman, 2012) debt can be intentional or unintentional. The intentional debt can be broken down further into short- and long-term debt classification (Alzaghoul & Bahsoon, 2014).

Technical debt management has showed to measure, reduce and prioritise technical debt in the IT development lifecycle (Allman, 2012). The prioritisation of technical debt consists of techniques, tools and procedures to use in the IT lifecycle. The current and existing literature on technical debt prioritisation has developed models and frameworks. Prioritisation workshops of technical debt in past years has gathered multiple of studies. However, prioritisation techniques are challenging to implement for managers and managers(Alzaghoul & Bahsoon, 2013).

According to Power (2003) there are seven challenges regarding the technical prioritisation: (1) measuring technical debt; (2) agreeing what items should be classified as technical debt; (3) prioritisation of technical debt over time; (4) understand the delay with associated costs and accrued interest; (5) the root cause of defects that created the technical debt; (6) the consequences of ignoring technical debt while multiple releases are implemented and (7) the technical debt needs to be visualized to get a clearer picture(Dos Santos, Varella,Dantas& Borges,2013).

Different management activities have been studied and gathered in the area of technical debt. According to Li, Avgeriou and Liang (2015) these activities may be divided into the eight categories as follows:

- Repayment
- Measurement
- Prioritization
- Identification
- Prevention
- Representation
- Documentation
- Monitoring

This study mainly focuses on only one of the eight management activities, namely, Prioritization (Li et al., 2015).

Li, et al define their prioritization by the ranking of technical debt items according to predefined rules that identify which of the items are:

- Deemed as priority and to be repaid first;
- Classified as least priority; or
- If they should be deferred to later release (Li et al., 2015).

Various other practitioners with different approaches, such as formal approximations and the finance domain itself (Izurieta, Vetrò, Zazworka, Cai, Seaman & Shull, 2012), have also recommended the prioritization of technical debt. In their case study survey conducted of 15 large organizations, they predicted the amount of time spent managing technical debt to be 25%, although with variance.

The tracking of technical debt is part of the developer's responsibility, acting independently using various tools such as backlogs and static analysers to manage the overhead of approximately 7%. However, 26% of the practitioners in this study used tools to track technical debt, while 7.2% opted for a systematic approach. Lack of knowledge contributes to these low numbers (Besker et al, 2018). The breakdown of this research is shown in Fig 1.1.

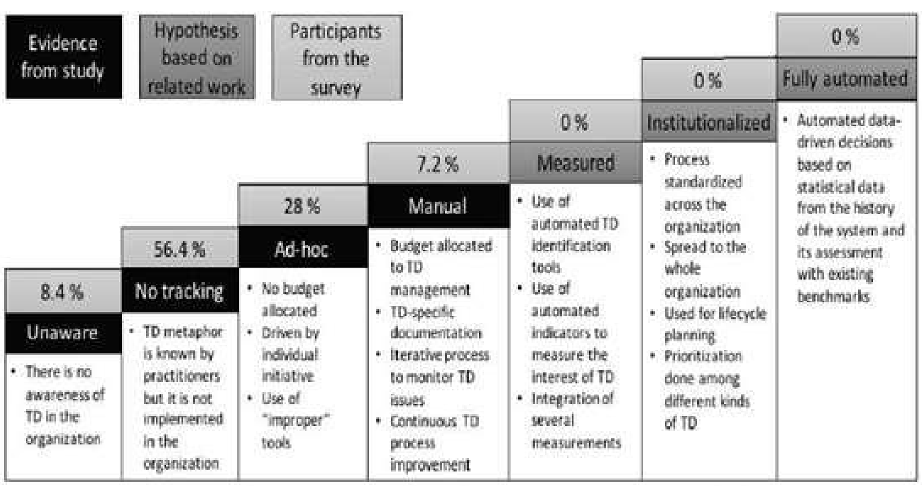

Fig 1.1. The Strategic Adoption model for tracking technical debt: the main milestones and the state of practice (Besker et al, 2018).

C. Definition of Terms
The following table outlines the terminology used in this research

Table I. Definition of terms and abbreviations

Term	Definition
TD	Technical debt
ICT	Information Communication and Technology
Principal	Effort estimation to resolve this debt item
Interest	Extra effort estimation to resolve this debt item
Liabilities	The state of being liable for something or an obligation arising from a past business transaction

II. RELATED WORK

Software development has become the focal point of interest for organisations to compete with competitors and manage customer expectations. The dependency on software systems through a range of devices, applications and wearable technology has increased exponentially with time (Ho & Ruhe, 2014). Owing to their sensitivity and interconnectedness, customers (key stakeholders) expect these systems to be highly available and reliable since any financial and operational issues will create significant problems for them (Zablah & Murphy, 2015).

In 2013, one study (Cambridge University, 2013) found that fixing software errors and the cost of debugging was estimated at USD$312 billion. The statistic indicate that for every dollar spent is used to fix the code. The industry, therefore, needs to reform and improve systems in their organisations and improve how they design and develop new software. The challenge is that technical debt activities are ongoing while the environment is constantly changing and it, therefore, becomes difficult to model real-world problems with immediate solutions (Zablah & Murphy, 2015).

A. Background and Discussion

Technical debt grows over time because firstly, different technical resources maintain financial systems; secondly, assumptions are made about the direction of the code or design philosophy; and inevitably, new defects are introduced (Ammerlaan, Veninga & Zaidman, 2013). The code needs to be maintained, irrespective of the magnitude of changes, so that short- or long-term solutions remain a concern as well the associated risks that accrue for every additional defect initiated with those changes (Kalm, 2018).

B. Defining Technical Debt (TD)

The concept of TD was borrowed from financial debt circles as it defines the achievement of long-term goals with sub-optimal solutions that have negative effects arising from the implementation of short-term benefits (Vogel-Heuser, Rösch, Martini & Tichy, 2015).The term has been widely studied in pure systems, despite a lack of research in the financial system industry as a whole. According to Martini, Besker and Bosch (2018) one of the examples of TD is that poor software quality does not adhere to standards and deliberate feature development.

The Fowler Definition of TD

As indicated in Figure 2.1 below, Martin Fowler came up with four quadrants by applying McConnell's (Fowler, 1999) concepts further. Fowler's TD Quadrant attempts to classify or source TD categories based on context and intent. Fowler argues that debt can be defined by intent: deliberate, or inadvertent. Further, he also categorises deliberate debt as reckless or prudent as indicated in Fig 2.1

TD needs to be prioritised and managed rather than ignored hence the Quadrant addresses symptoms of TD early as something that cannot be avoided.

Prudent debt. This relates to not paying TD debt if the interest is sufficiently small. **Reckless debt.** The effect is a longer payback period to pay the principal amount or incurring some crippling interest.

Fig. 2.1 The Technical Debt Quadrant by Martin Fowler (1999)

The Curtis definition of TD

Bill Curtis provides three definitions concerning TD as follows: technical and principal debt and interest (Curtis, 2012).

Technical debt - This is when future costs are known as result of structural flaws in the system. The cost includes both the interest and principal amount. The known TD item is included in the calculation of TD as the owners of the application they believe it must be fixed. The primary component of TD is the cost of the application (Curtis, 2012).

Principal debt - This is the cost of redressing the problem in the application. At minimum the cost are calculated from the number of hours required to remediate the bug, multiplied by other resources hourly rate to be involved in designing, testing of fixes and implementing (Curtis, 2012).

Interest - This continuing cost is a result of fixing the problem in the production code. It is worthwhile to note that modifying complex code produces excessive effort as it take up resource usage (Curtis, 2012).

C. Technical Debt Taxonomy

Since the TD comes from different stages of the development and has a variety of forms it becomes impossible to diagnose problems without understanding the root cause and risk factors (Holvitie & Leppanen, 2013). Often, the business case is misunderstood during the design phase where different roles that actors have in the design process (namely, architects, development team, and analysts) may affect the agile or normative Waterfall model as the project evolves (Kalm, 2018). The Table II below shows the eight TD types that have been studied in previous research.

Table II: TD taxonomy

Technical debt types	Definitions
Code Debt	Source code is often so poorly described that it requires refactoring and does not follow best practices of programming and business rules. Examples include bad style (which makes readability an issue) and the complex code (which makes the logic difficult to follow). (Leppänen, Lahtinen, Kuusinen, Mäkinen, Männistö, Itkonen 2015).
Design Debt	This takes shortcuts to deviate from good principles of design to under-focus qualities such as adaptability and maintenance (Reimanis & Izurieta, 2016).
Environmental Debt	Software development, infrastructure and hardware tend to create environmental debt. Postponing upgrade changes to the infrastructure, hardware and software are some examples of this. All these activities can increase productivity if they are completed timeously (Tom, Aurum & Vidgen, 2013).

Architecture Debt	Substandard solutions that lack long-term sustainability because internal quality is compromised such as: performance, modularity reuse, robustness, among others. Architecture debt type requires a lot of activities to fix structural defects caused by bad decisions to create the foundation (Alves, Mendes, de Mendonça, Spínola, Shull & Seaman, 2016).
Knowledge Distribution Debt	Knowledge of code that is not documented with a long history has an effect on new members of the team especially if the original members do not want to share or have left the organization (Tom et al. 2013).
Test Debt	This is a lack of proper unit testing that does not include scenarios; acceptance tests that do not stick to the business rules; a failure to do comprehensive testing, and taking shortcuts (Szabados & Kovács, 2015).
Documentation Debt	This is inadequate documentation that is incomplete and outdated (Alves et al. 2016).
Infrastructure Debt	These are the costs involved to refresh components of a system such as storage, servers, anti-virus, operating system, middleware applications by patching or fixing them rather than by basing them on the IT and business strategy (Alves et al. 2016).

D. Different Approaches to Prioritise TD

The research community is well represented on the TD topic from a theoretical and practical perspective. However, there is limited research on the prioritization of TD types in the South African financial services context and how this approach is carried out practically by organizations(De Groot, Nugroho, Back & Visser, 2012).

Various researchers in the literature suggest numerous ways to arrive at the decision-making process when TD is prioritised (Falessi, Kruchten,Nord & Ozkaya, 2014).

Further, the industries are not unique as to how business functions and most of the research publications that prioritise TD consider different aspects for the process they recommend management should adopt. The most frequent approach is a cost-benefit analysis as the preferred method although other approaches are also acceptable within the TD community (Eisenberg, 2012).

Falessi et al. (2013) approach looks at quality rules that are violated when prioritization is practiced. Another common approach, is the payment of debt as a result of TD impact on software influence from a customer journey, and the severity and localization of TD items (Holvitie, 2014).

In another publication, the strategy prioritised TD by identifying break points (timing) to repay the debt where the principal amount corresponds or below the TD interest accumulated. According to Snipes et al. (2012) a cost benefit analysis factors-in financial decisions when TD is prioritised (Chatzigeorgiou et al. 2019).

Seaman adopted a different strategy to decide what TD items should be held back or accrued as an investment strategy(Seaman,2012).To do this involves all stakeholder perspectives into the decision-making process when prioritising TD.

Another study defined key factor variables such as maintenance costs, risk and lead-time as important factors. Architects and project owners are able to provide this information during the prioritization process or when activity architecture is practiced (Martini & Bosch, 2019).

There is not much difference between the above research in the literature and the approach chosen for this study in that it is not far from the Martin and Bosch method that considers in-risk as a vital aspect when prioritising TD. Another difference is that this study's focus is only on the South African financial services.

E. Conceptual Technical Debt Framework

The TD framework exists in the literature to give an overview of technical components that define the overall phenomenon. The systematic framework presented in Figure 2.2 below consists of three main areas: Precedents, TD key drivers and intended outcomes (Tom et al. 2013).

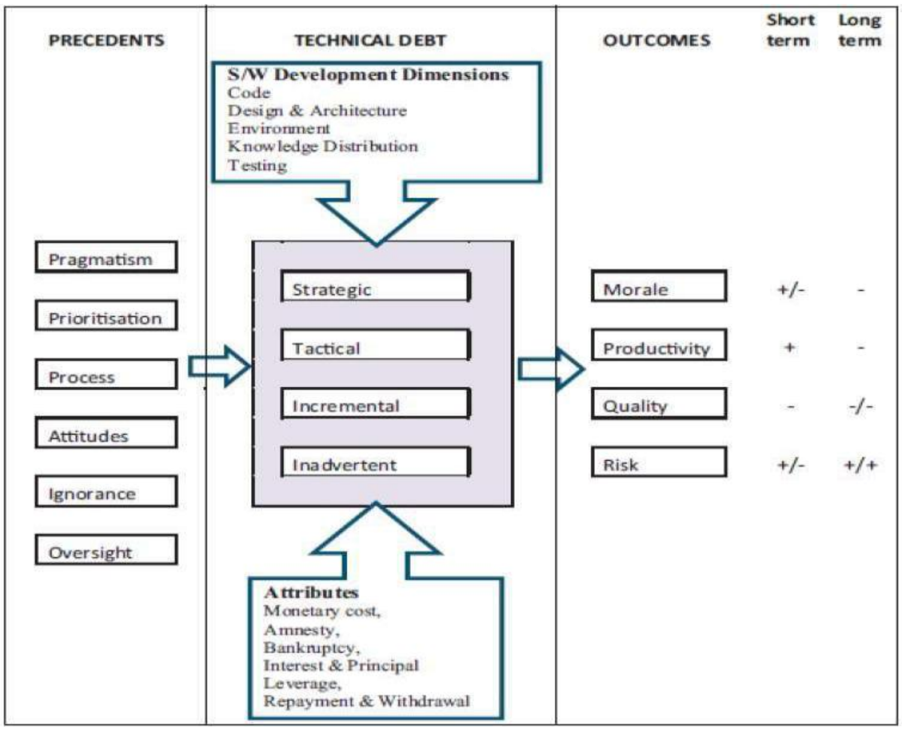

Fig. 2.2. Technical Debt Framework (Tom et al., 2013)

The framework model addresses different precedents and outcomes of TD as described above in Figure 2.2. The framework model is comprehensive and summarises key concepts of TD, therefore it has been adopted to answer the research question and is used in conjunction with the IT risk framework in the financial service industry in South Africa.

F. TD on the Balance Sheet and Budget Estimate

In financial services, management always needs to understand the financial position of the organisation, and TD is therefore appended to financial liability on the balance sheet. The TD line item on the balance sheet can capitalise software and create a credit limit when TD reaches a certain level; however, too many TDs will reflect a financial risk (Akbarinasaji & Bener, 2016).

Organisations with a low liability on their current balance sheet tend to reflect accurate ratios and a healthy financial prospects (Holvitie, Leppanen & Hyrynsalmi, 2014). Incurred interest in a form of increased cost exhibits the financial debt as it conforms to the TD metaphor and the potential bankruptcy when the system is rewritten or new platform introduced (Tom et al., 2013).

G. TD Study Overview

Surveys in the literature of various companies on their current state of TD gives an overview of the following issues found (Besker et al. 2018).

These were categorised as:

- Unaware of TD
- No tracking
- Ad hoc application
- Use of incorrect technology
- Low quality maintenance data
- Lack of documentation
- Incomplete system testing
- Manual Tracking

These may be described as follows:

- **Unaware of TD** - According to the survey data, only 8.4% were able to manage TD. The lack of awareness of TD, and zero management capabilities were reported in high numbers (Besker et al. 2018).

- **No Tracking -** Software engineers seemed generally aware of the TD notion and the negative effects were known to the system but they reported that there was no traction to track it. The majority of respondents (56.4%) were reported to be in the category that disagreed with tracking TD (Fraser, Mancl, Opdyke, Bishop,Kathail, Lacar, Ozkaya & Szynkarski, 2013).

- **Ad hoc application -** According to the survey data TD awareness exists, since some technical software developers track or have started to track it, which then makes it an ad hoc activity for management. With no assigned budget for TD application, the technical resources use what is available in terms of processes and tools (Guo, Seaman, Zazworka & Shull, 2010).

- **Incorrect use of technology -** When technology is used that does not support business objectives or that is not the best-fit technology (due to minimum viable product strategies, fewer features and applications) it does not convey the best solution. Only iterative agile prototyping reduces cost overrun (Berenbach, 2014).

- **Low quality maintenance data -** Non-existent documentation of overtime contributes to the cost of TD when the system needs to be maintained. Technical resources interpret functional code with their limited knowledge of the system and perform maintenance activities if and when they see bugs without following a prescribed documentation (Alves et al. 2014).

- **Incomplete System Testing -** Testing of the system is another component that creates TD by being incomplete or test pack deferred. Ineffective testing increases the risk of potential failures, security concerns and other defects (Szabados & Kovács, 2015).

- **Manual Tracking -** There were some organizations that acknowledged the support of TD and their management's use of such initiatives as the budget allocated. According to one case study, the estimated budget allocated was between 10% and 30%. The analysis from the case study listed specific TD values with their principal amount, risk and interest. Here, an iterative process is in place to monitor TD (estimate, repay and identify) and this is subject to continuous improvement (Griffith, Reimanis, Izurieta, Codabux, Deo & Williams, 2014).

H. Significance of TD against quality attributes

The quality attributes in embedded software development have also been examined in the literature. The study by Ampatzoglou et al. (2016) highlights a different approach for how these attributes are prioritised. The trade-off among the quality attributes in embedded software. The attributes of interest are performance, security, understandability, functionality, effectiveness and flexibility (Guo & Seaman, 2011).

The researchers provide some of this relationship to quality attributes as the dimension of software quality and TD have direct relationship with each other. Essentially, TD can be seen as a priority in software quality because workarounds tend to increase complexity over a period and short-cuts have a negative impact on quality attributes (Avgeriou et al. 2016).

I. The IT Risks Framework in Financial Services

The Deloitte's IT Risk Management framework (see Figure 2.3 below) where each domain or component is ranked relative to each TD.

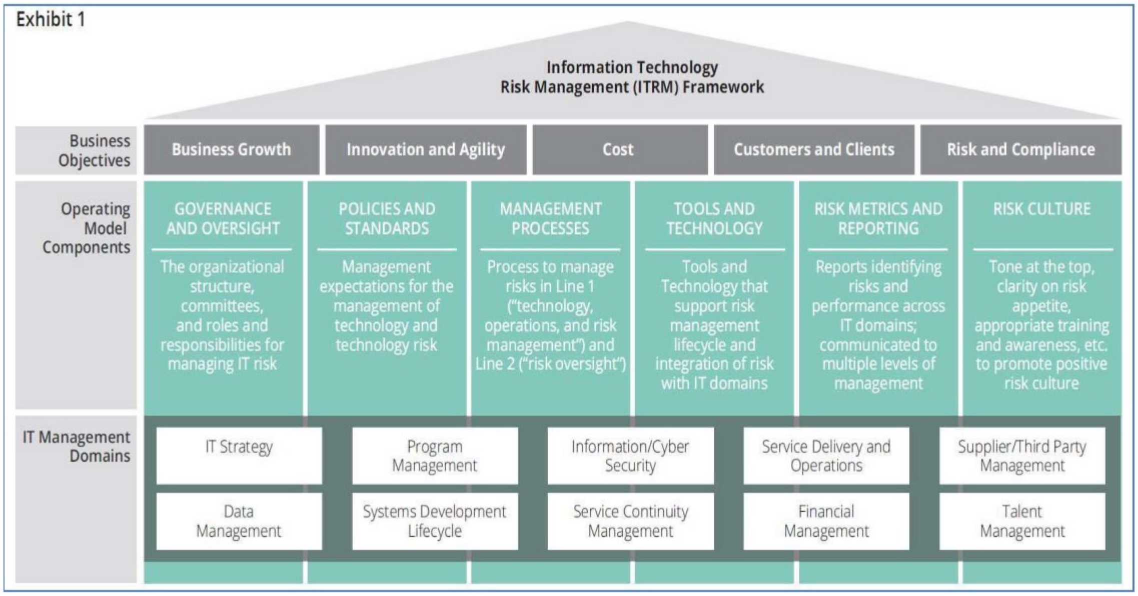

Fig 2.3 IT Risk Management Framework

Financial institutions face different types of IT risk particularly when business is not aligned to the IT strategies. This occurs in the following circumstances:
When management continues to increase costs; complicate the environment; and mismatch IT talent. The consequences are that the organisation will be disrupted and therefore hinder business agility (Board, 2004).

When management does not budget for and prioritise TD properly. Many technology "FinTech" companies are redefining the business models and processes without carrying TD. If management teams prioritise TD based on technical and business risks assessment it enables a swift response to focus on new requirements that are relevant to the customer (Deloitte, 2018).

The factors to measure TD are difficult to gauge as they are not always numeric and because the context of each organization differs (Griffith, Izurieta,Taffahi & Claudio,2014). However, this study has examined current management practices in financial services to recommend improvements to the quality attributes of maintenance, customer satisfaction and systems performance (Falessi et al. 2013).

Strategic risks of IT

The IT environment is constantly changing and risks derived from ineffective strategies include embracing new technology or ignoring it due to too much debt. This implies that financial services must balance the risks and prioritise accordingly. Recently, the FinTech solutions have shifted their innovation in the financial services industry by disrupting the status quo with cloud solutions which now promise greater benefits but which carry significant risks (Deloitte, 2018).

Cybersecurity

In 2018, many reports have surfaced about the need for financial institutions to tighten their privacy and security. The hacking of data has significant breach on financial services. Since then, the directors and boards are always mindful and have made it part of their agenda to push management to review cyber risk as a serious priority. This implies, therefore, that if TD is related to security it must be a priority.

Vendor and third-party risk

Financial services organizations with joint ventures and vendors need to arrange to do their internal risks with applications they support. However, the risks and TD can be obscure. For example, the vendor can make software updates as part of a maintenance support contract but could then become incompatible with implementation by clients as other risks could be hidden(Izurieta,Griffith,Reimanis & Luhr, 2013).

Similarly, the technical developers can take shortcuts without understanding the holistic view of the system and potentially damage the core architecture of the underlying system. This would then introduce more risks to the maintenance plans of the vendor (Ramasubbu & Kemerer, 2016).

Data management risk

The key is for effective data management in the financial institution to avoid loss of trust from stakeholders and customers (Izurieta, Griffith, Reimanis & Luhr, 2013). Regulatory agencies have a stake in the accounting and data management capabilities as the risk and financial management require accurate data and timely responses. However, new risks can manifest themselves when new layers of rules are added with the combination of internal and external data (Deloitte, 2018).

Technology risk management

With the technological environment constantly changing, the risk is that new technology could become unusable in the future for several reasons, namely:

- Lack of sufficient testing (Kim, Zimmermann & Nagappan, 2012).
- An inappropriate release without taking care of the consequences arising from the introduction of change.

Both reasons pose a significant risk to the environment (Deloitte, 2018).

IT Program execution risk

Financial service industry organizations usually have multiple IT software systems across different business units and geographic locations. This software includes finance systems, enterprise resource planning (ERP), and customer relationship management (CRM), billing and commission systems, among others. All these systems are prone to risk, such as: misalignment of strategies and execution; delays to serve the business; poor governance that has failed to accommodate risks; and lack of communication to address all business units with one voice (Lane,Koolmanojwong & Boehm, 2013).

J. Conclusion

This section has looked at existing literature in relation to different types of TD debt and management approaches to prioritise. The section has discussed different definitions of TD and used the conceptual framework of TD outcomes such as risk, management morale, productivity and quality to base the research objectives (Kim, Cai & Kim 2011).

III. RESEARCH STUDY DESIGN

The objective of this research is to gain an insight into the understanding of how the South African financial services industry views and prioritises TD types and accordingly, how this is linked to the related risks. Therefore, the research questions look at five different TD types in accordance to those of Alves et al. In addition, the research direction for these TD types are particularly focused on the South African financial services industry.

While two research questions are used to solicit more understanding of TD and IT risk factors, the questionnaire has various other questions to gather more information on the perception of TD types and references. It also has an IT risk framework as a guide to rate them. In addition, it assesses how management treats these factors in the context of the environment (Fernández-Sánchez, Garbajosa, & Yagüe, 2015).

A. Objective and research question

A quantitative methodology was used as the primary method to determine the prioritization of TD types perceived by participants of this research. The objective was to analyse the sample population data collected using the Qualtrics survey tool and to determine which of the IT risks correlate to TD types more at a statistically significant level.

The quantitative approach made it easier to collect the data unlike the time-consuming data collected for a qualitative study. The anonymity of the questionnaire distributed also gave the respondents freedom to reply truthfully about their understanding of TD within their organizations as sensitive information that may hamper prospects of the business is not to be disclosed in this study (Apuke, 2017).

B. Problem Statement

The main problem is to determine which technical debt types should be prioritised in the financial service industry, and to make suggestions to management for the development of strategies that can be applied to existing systems and current projects.

Sub-question

Research Question 1: Which TD types are most urgent to resolve in the financial service industry within the South African context?

> **Hypothesis 1:** Evaluates the means of selected TD types by respondents if they are significant.
>
> **Hypothesis 2:** Seeks to determine if a relationship exists on budget estimate means.
>
> **Hypothesis 3:** Seeks to determine that a relationship exists across the quality attribute values.

Research Question 2: Which aspects of risk management framework will be useful to prioritise TD types?

Hypothesis 4: Seeks to determine if IT risk variable values are significant to each other.

C. Research design

This study's quantitative research design approach also ensured the following participants were not manipulated when providing feedback. The research investigated the relationship variables of TD types against an IT risks framework and budget estimates and quality attributes were to be prioritised by the respondents.

Another rationale behind the research approach is that it is less time-consuming to conduct and distribute the questionnaire online rather than using the face to face interview approach. This ensured that the participants were able to answer questions willingly in the comfort of their environment without being under pressure. The final survey recorded responses (n = 72) with only 18 questions incomplete tends to confirm this view.

Prior to the distribution of the questionnaire, the variables were identified as part of the data analysis section. This gave insights for how they were defined or grouped. The questionnaire comprised various questions divided into five sections as follows:

- Introduction about study and TD concepts.
  Demographics.
- Open-ended questions to build up the understanding of the TD concept for those participants not familiar with it.
- The discrete data section to perform effective quantitative analysis (below is TD research model as illustrated by phase 1).
- The budget estimate of TD as a supporting variable to quantify the market share of budget assigned per TD type.

As shown in Table III below the study is divided into five phases whereby each phase and the action items to be achieved are tabulated.

Table III: TD research design phase approach

	Design phase	Action items
Phase 1	Preparation	Draft survey questions into Qualtrics tool, test with pilot study and delivery of questionnaire to target group
Phase 2	Data collection	Send the survey link to recipients to collect the data and monitor response rate
Phase 3	Data cleaning	Recode, remove incomplete responses to the data set and approve the data set, and rename column headings where they need appropriate changes for easy identification. Group related data columns that have numeric value to be used in the analysis methods.
Phase 4	Analysis and synthesis	Perform ANOVA data analysis as part of Step 1 and make decisions on critical values of ANOVA section if p-value is statistically significant at 95% interval test.
Phase 5	Analysis and synthesis	Continue Step 2 by using Phase 4 data from ANOVA to do further tabular analysis and correlation of selected variables.

As can be seen from the above table, the phases comprised the following activities:

Phase 1: The study was presented and discussed with this researcher's supervisor during numerous workshops and meetings. The final questionnaire was then emailed to the supervisor as well as those participants in the study who had a basic understanding of the TD concept.

Phase 2: The data collection was collected using Qualtrics tool by capturing the questions pertaining to the study. All the recorded and in-progress answers were collected as part of the sample size. All the questions were closed-ended questions to select a choice of TD preferred by the participant for how their companies should prioritise TD. The survey questions listed below were used to solicit answers pertaining to the two research questions mentioned in the literature.

Table IV: Research question statements

ID	Statement	RQ
ST1	Is your company prioritising Technical Debt?	1 and 2
ST2	A company's IT backlog priority are a combination of their strategies, new initiatives and future projects. Together, all of these things form a company's Technical Debt (TD). What do you think is the most important TD type that a company should prioritise (ranging from 1 for least important to 5 for most important)?	1 and 2
ST3	Using the previous answer TD type chosen, to what extent will the elements of the Risk Management Framework be useful in prioritising TD from a management perspective (ranging from 1 for least important to 8 for most important)?	2
ST4	Please rate these qualities attributes that should not be compromised when TD type is prioritised (ranging from 1 for least important to 7 for most important from a management perspective).	1
ST4	Please estimate the share of budget in your company that is allocated per TD. (Total must be 100%)	1

Phase 3: The subset of the analysis process that was performed when decoding or mapping different codes into numerical interpretation levels (this can be referenced in data analysis section).

Phase 4 & 5: In-depth analysis was performed with different types of measurement methods. An ANOVA was performed on the variables to ask the basic question if the variables chosen are different from one another. This answers the question by returning a p-value as the hypotheses created form part of the test using the ANOVA data analysis.

D. Pilot Study

The draft questionnaire was distributed selectively to assess errors for adjustment on the questions. This assessment consisted of a mini-pilot exercise of four individuals to test the validity of the survey and to check if the captured data filtered through the repository storage. The purpose of any pilot study is also to identify flaws, ambiguities, and to assess if any there was any discomfort or embarrassment the participants might have experienced. The pre-test also helps to minimize fundamental errors and make sure the target audience understands the objective of the research without having doubts or questions (van Teijlingen & Hundley, 2002).

All the errors picked up during the pilot exercise were amended on the questionnaire; the link was also tested to address recently published questions.

E. Population and Sample Size

The target audience for the survey were individuals involved in IT; namely, professionals, system integrators, project managers, contracted third party providers, and independent contractors in the financial services industry in South Africa. The participants received the questionnaire’s first page (introduction) which defined the meaning of TD and had examples. In the 70 questionnaires distributed, the participants were also instructed to forward data to other recipients whom they thought might fit the criteria to answer questions of high-level technical depth.

The sample population was derived from the financial services industry; particularly those with prior experience of the subject who have worked for small-to-large enterprises (start-up or mature organizations) in the sector.

The final sample size was based on the criteria of previous research published in the literature. However, since the topic has not been well studied in the South African context, the sample was further determined by two other factors:

The estimated response rate;

The approach followed for the performance of the analysis

F. Data Sample and Synthesis

The final survey had 72 recorded responses from respondents were assumed to have acquired at least an understanding of Technical Debt (TD) during their careers. These respondents comprised current employees at this researcher's workplace, other postgraduate students, and employees in the general IT sector or financial services industry in South Africa.

The data was filtered to the technical debt type that resonates with respondents and which followed the research design method. In the total number of recorded answers (beginning to end) opted for different technical debt selections.

The purpose of the questionnaire was to get an insight into how the financial services in South African prioritise TD to make further decisions. The respondents who had incomplete responses (n = 15) or some missing data in the questionnaire were removed since all data was necessary to form the basis of the analysis. The TD rankings were critical to indicate respondents' perceptions for how TD is prioritised by their management. This was also done to make the demographics of the respondents meaningful for a sound conclusion. The incomplete responses had a progress status less than 89%.

No outliers were removed that were biased towards a particular TD type, as questions were fair. In two responses on the age category question the response 'other' was selected but these were kept as part of final dataset.

Further, the TD rankings indirectly indicated what other important issues there were for the company if management complied or aligned with industry practices of TD prioritization.

The TD type constructs were all defined upfront as part of the questionnaire in accordance with what was popular in the literature and with the TD community, namely:

- Code Debt
- Design Debt
- Architecture Debt
- Infrastructure Debt
- Documentation Debt

It was easier to analyse a narrowed list of TD types as 'other' if they were not popular in other industries or not studied frequently. The questions were closed-ended selection questions. This made it easier to select predetermined answers on the form as opposed to open questions which are difficult to map and which may reflect bias towards a particular TD type.

G. Research Instrument

The survey questionnaire was used to gather the necessary insights to answer the research questions. Qualtrics generated the survey link via email and captured the questions of the study since it was expected that participants had internet access or a mobile device to accommodate the Qualtrics questionnaire structure. The survey used the referral approach that did not restrict individuals passing questionnaire data to their friends and colleagues who had a basic understanding of TD and computer literacy and were within the sector.

The questionnaire had 18 question items that were designed specifically for the TD prioritization in the study. Adjustments were necessary to fit in with the research context of the study in the sense that a five to seven scale was adapted for the questions. The scales were rated from 1 (the lowest) to 8 (the highest) to indicate the preferred choice.

IV. Data Collection

Data was collected by creating an online survey using Qualtrics. The period given for the online survey was from 15 June to 30th June 2019 after which it was made inactive. The responses were then stored on Qualtrics repository for further analysis.

An introduction to TD was given in the initial of the questionnaire and affirmed that the collected information and data would be unlinked once the purpose of the study was achieved. The questionnaire was self-administered to ensure there was no external or direct interaction with respondents when they completed the survey.

A. Measures of TD prioritization

The analysis focused the approach on the prioritization of the TD types as per Research Questions 1 and 2:

The respondents' answers to the most important TD type questions in the questionnaire requested them to prioritise and rank the technical debt types by using their previous or current experience on the TD subject. An understanding of this was also given at the start of the survey.

The main question was restricted to five TD types: Design, Infrastructure, Document, Code and Architecture to match those mostly studied in the literature. The selection of TD type for each respondent was limited to a maximum value of these five options available. Then the ANOVA statistical analysis was performed for all the TD types to get a better understanding of how respondents in the South African financial services industry prioritise TD. This was done by using other variables such as quality attributes, aspects of IT risk framework, and budget estimation.

A numerical quality attribute (QA) question of the questionnaire asked the respondents to allocate the rank of importance for attributes that should not be compromised for TD prioritization. The results were summarised and grouped according to the TD type perceived according to priority. For each of these scores a summary analysis was used to gain an insight into the respondents' perceived prioritization preferences.

B. Steps to correlate TD type and variables

Correlation was performed for the Research Questions 1 and 2 to understand the relationship between variable 'x' (budget estimate, quality attribute and IT risk factor) and the selected TD type to be prioritised in the following ways:

- ANOVA data analysis was performed on each variable that had the highest value to ascertain the most dominate choice to be used on the correlation analysis.
- The variables with the highest value was further analysed against the architecture debt type by correlating the relationship between the two variables.
- The processes were repeated for each variable with highest value as part of the study.

C. Technical debt data points

The variables been selected and the corresponding data analysis approach are shown in Table IX.

[V1] Selected Technical Debt Type - The types of TD presented as of part of the questionnaire component were referenced from the work by Li et al. and listed as follows: Architecture, Design, Documentation, Infrastructure and Code. The TD community recognises the existence of these TD types.

[V2] Technical Debt Budget - This variable represents how of the Technical Debt is estimated based on respondents' observation or experiences in their environment. The input percentage field was presented as a closed ended selective answer by asking respondents to indicate the budget an organization should be allocated per TD type when prioritising. The questionnaire form indicated that the total to sum had to add up to 100 percent. ANOVA statistical analysis was performed to get a better understanding of each respondent's mean and p-values.

[V3] Quality Attribute - A scale of 1 (lowest) to 5 (highest) was used to rate the following quality TD attributes according to their importance: Usability, Functionality, Reliability, Performance, Security, Compatibility and Portability.

[V4] IT Risk Domain - This variable was used in the context of IT risk framework in the financial services of South Africa as an approach to prioritise TD types. The IT risk factors are as follows: IT Strategy, Project Management, Information/Cyber Security, Service Delivery and Operations, Systems Development and lifecycle, Service Continuity Management, Financial Management and Talent Management.

This section presented the analysis of the collected data. The respondents were asked their basic understanding of TD and allowed them to prioritise their choice through the questionnaire with other supporting variables. The data was analysed, themes and observations were present. The findings are discussed in results section.

V. DATA ANALYSIS

The study is premised on two research questions. An ANOVA statistical analysis and hypothesis was performed to answer these questions. The analysis was carried out for each research question as presented in Table V. In addition, a correlation was performed related to the variables contained in these research questions. Regarding research question 1, the quality attribute and budget were unveil to see if there is a relationship against the selected TD type by respondents. Finally, research question 2 explored the IT risk framework to see which of the factors should be used when TD is prioritised.

Table V: Technical debt data points

Research Question	ANALYSIS PLAN	
	Used Variables	**Analysis**
1	**V1** (TD selected) **V2** (Technical Debt Budget) **V3** (System Quality Attribute)	ANOVA single factor Cross-Tabulation Hypothesis testing Correlation analysis
2	**V4** (IT risk factors)	ANOVA single factor Cross-Tabulation Hypothesis testing Correlation analysis

VI. RESULTS

Project demographics

The final dataset was verified through different stages of preparation by removing unnecessary biases to keep it clean. Cross-tabular and pivot grouping analyses were used on the demographics.

A. Age of respondents

As indicated in Table VI below, all the respondents agreed to supply their age as part of the questionnaire.

Table VI: Age of respondents

Age category	Count	Percentage
18-25	19	26.39%
26-35	16	22.22%
36- 45	33	45.83%
46-55	4	5.56%
Grand Total	72	100.00%

The age question analysis was coded into five categories and the results showed no selections above 56 years. An age category of 36-45(45.83%) was recorded as the majority while the minority category had 5.35%. Two respondents did not select any age category.

B. Education of respondents

The table analysis shown in Table VII below reveals that the majority of respondents (50%) had graduated with a degree from an academic institution, with the least percentage (1.39%) are doctoral degree and less high school.

Table VII: Education of respondents

Qualification Type	Count	Percentage
Less than High School	1	1.39%
Doctoral Degree	1	1.39%
High School	2	2.78%
Some University or College	5	6.94%
Diploma	12	16.67%
Masters Degree	15	20.83%
Degree	36	50%
Grand Total	72	100%

C. Gender of respondents

The gender of respondents had 29 females, 43 males and all the respondents answered the gender question rather than the option for "Prefer not answer".

Table VIII: Gender of respondents

Gender Category	**Count**	**Percentage**
Female	29	40.28%
Male	43	59.72%
Total	72	100%

D. Prior knowledge of technical debt

The confident answers made on knowledge statistics indicated how well the participants understood the concepts of TD consistently for these questions and throughout the rest of the survey. The Figure 4.1 shows the majority of respondents had an understanding of TD by scoring more than 50% (shared between "Well knowledgeable" and "Adequately knowledgeable"). Those who had no prior knowledge of the subject accounted for 17.19%

while 15.63% reported “Poorly knowledgeable” even though an introduction was given with two examples at the beginning of the survey.

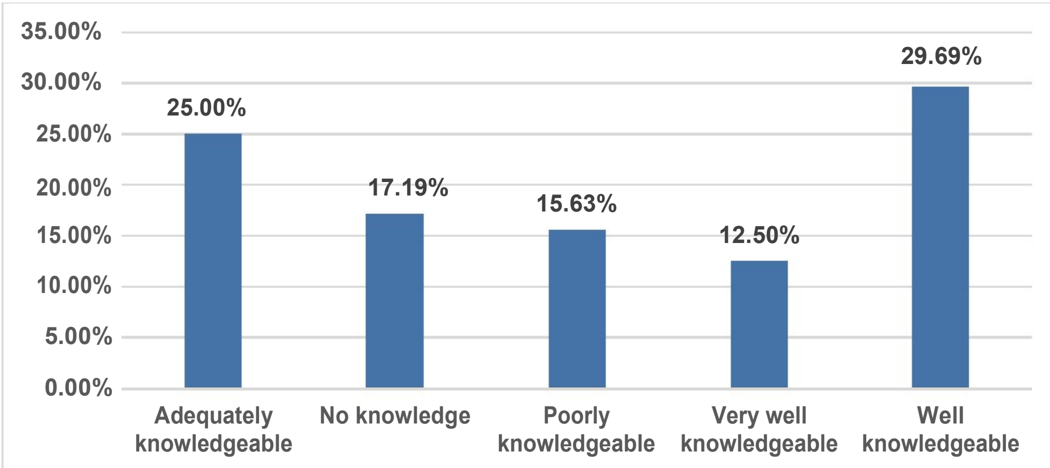

Fig 4.1 Prior Knowledge of Technical Debt

E. Working experience of respondents

The following graph (Fig. 4.2) depicts the work experience of the respondents. The highest percentage of 34.85% for those who had less than 5 years work experience can be attributed to the group that reported 17.19% (no knowledge of technical debt) in Figure 4.1. The significance of the relationship between the outcome for these variables is not so important as part of the study. The second and third group (above 40%) reported their work experience ranged between 5 – 15 years.

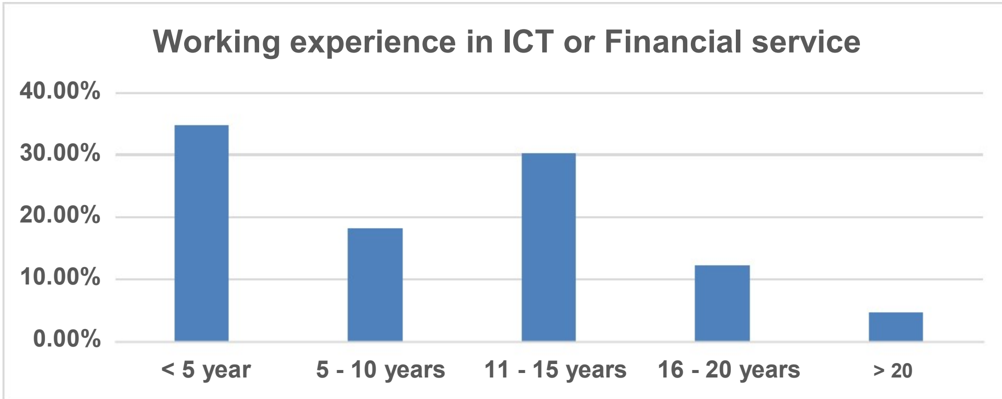

Fig. 4.2 Working experience of participants in ICT or Financial Services

F. Stakeholder's prioritization of technical debt

The following graph (Fig. 4.3) indicates majority of the respondents (49%) reported where TD is prioritised within their organization. This was followed by a combination (31.37%) of participants reporting IT vendor and in-house partnerships.

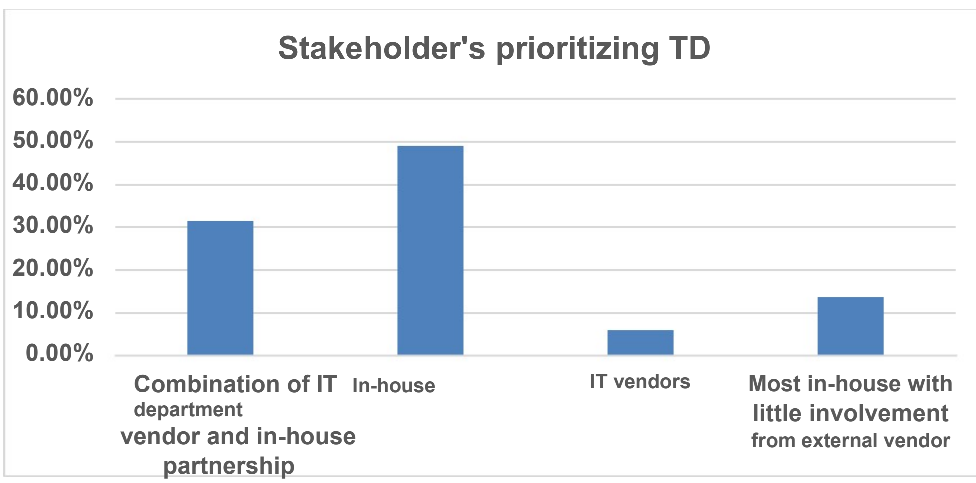

Fig 4.3 Stakeholder's prioritization of TD

G. Statistical analysis

The purpose of the statistical analysis is to answer the research questions given. Two research questions were presented as part of the study. Furthermore, the preferred TD type (V1) against V2 (budget estimate) and V3 (quality attribute) to see if a relationship exists by applying ANOVA data analysis as part of research RQ1 and then to test the null hypothesis if it is accepted or rejected.

For RQ2 an ANOVA data analysis is performed on TD type selected (V1) and IT risk domain (V4) to unveil the relationship as to what participants perceived the most important TD to be prioritised. The hypothesis informed the decision making to reject or accept the null hypothesis at its level of significance.

H. Research Question 1

RQ1: Which TD type is most urgent to resolve in the financial services industry in the South African context?

Hypothesis 1: The hypothesis evaluates the means of selected TD types by respondents if they are significant.

> **H0:** There are no differences in mean values on selected TD types
>
> **H1:** There are differences in mean values on selected TD types

The rationale to prioritise the most frequently used TD type in the literature with the most frequently used or reported by Li et al. (2005).

The ANOVA statistical analysis was then performed. Figure 5.1 below shows the bar chart on the most selected TD types (V1) prioritised by respondents.in financial services.

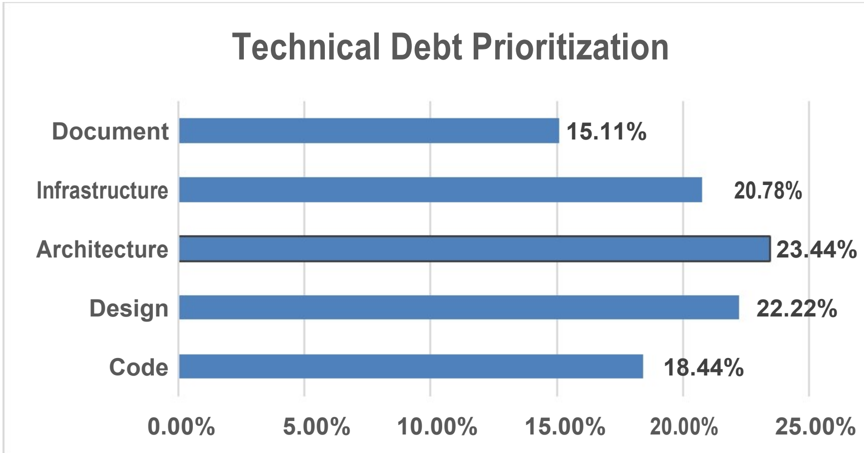

Fig 5.1 Technical debt prioritization

The TD types given in the questionnaire were mostly researched in the TD literature (namely, architecture, design and documentation). Other technical debt types were either not included in this study or understudied by the TD community; namely, versioning, and requirements. The architecture debt type was chosen by the majority of respondents after performing a ANOVA analysis for each TD type. The Table IX contains the summary of this exercise.

I. Prioritization of Technical Debt type by respondents (V1)

One-way ANOVA was conducted and the output was then divided into summary and ANOVA in the supplementary files (**Technical Debt types ANOVA Statistical Output A**). The statistic was run as a single factor. The ANOVA portion of the results then showed the between groups and within groups. The between groups (TD types) results were then interpreted.

The F value is 8.04 and F critical value of 2.40. It can be seen that F value is greater than F critical value, therefore, this means the findings are statistically significant between the technical debt types.

The p-value is 3.59E-06 which indicates there is only a small percentage chance the observation could have been through random error. The p-value is below alpha of 0.05, therefore the null hypothesis (H0) that states that there are no differences in mean values on TD types is rejected and the alternative hypothesis (H1) that the TD types are different as observed by the means is accepted.

The ANOVA summary

The TD types hypothesis results shown in Table IX have five variables: code debt, design debt, architecture debt, infrastructure debt and document debt.

Table IX: Analysis of TD types and related hypothesis summary

	Code	Design	Architecture	Infrastructure	Document
Mean	2.77	3.33	3.52	3.12	2.27
Variance	2.05	1.44	1.58	1.6	2.5
Sum	166	200	211	187	136
Overall Percentage	18.44%	22.22%	23.44%	20.78%	15.11%

The architecture debt type had the highest mean of 3.52 out of the five TD types selected. Likewise, design debt selection became the 2nd and 3rd infrastructure most popular. This was, followed by Code Debt and Document Debt respectively. From the summary above, it is apparent that the majority (23.44%) of respondents selected the Architecture Debt type to be prioritised in the respondents' sample.

J. The TD Budget allocated (V2)

Hypothesis 2: The hypothesis seeks to determine if a relationship exists on budget estimate means:

H0: There are no differences on TD budget estimate mean values

H1: There are differences in mean values on the TD budget estimate

The ANOVA single factor was performed as output under supplementary files (**Budget Estimate TD types ANOVA Statistical Output**) with summary and ANOVA section. The values between the technical debt types (groups) were interpreted as follows:

It can be observed that the F value of 2.4235 is greater than the p-value of 0.0484, therefore the findings confirm that the values are statistical significant. The P value of 0.0484, has a small percentage error and is well below 0.05. The null hypothesis (H0) is rejected that states that there are no differences on TD budget estimate mean values and accept the alternative hypothesis(H1) that the TD types are different as observed by the mean calculated values.

TD Budget type ANOVA summary

The results for the hypothesis on TD budget type are shown in Table X below.

Table X: Analysis of TD budget estimate and related hypothesis

	Code	Design	Architecture	Infrastructure	Document
Mean	22.53	21.03	22.81	17.33	16.29
Confidence Level (95.0%)	298.71	123.47	256.16	173.0	233.82
Sum	1307	1220	1323	1005	945
Overall Percentage	22.53%	21.03%	22.81%	17.33%	16.29%

The architecture debt type had the highest mean of 22.81% out of the five TD types selected. Likewise, Code Debt answers were rated as second and Design Debt third. This was followed by Infrastructure Debt and Document Debt respectively. From the summary, therefore, it is apparent that the vast majority of respondents chose architecture debt type to be prioritised over other types.

K. Correlation between architecture and TD budget type

The results for the correlation on TD budget type are shown in Table XI below; Pearson's R is 0.07, which reflects a correlation between budget (architecture) estimated, and architecture debt type chosen. This indicates a weak correlation.

Table XI: Architecture and budget type correlation results

	Architecture Debt	Budget - Architecture Debt
Architecture Debt	1	
Budget - Architecture Debt	0.070298597	1

L. Quality Attribute (V3)

Hypothesis 3:

This hypothesis seeks to determine that a relationship exists among the quality attribute values.

H0: There are no differences in quality attribute means on the data sample

H1: There are differences in quality attribute means on the data sample

The ANOVA single factor was performed as output under supplementary files (**Quality Attribute ANOVA Statistical Output**) with summary and ANOVA section. The values between the technical debt types (groups) are interpreted as follows:

The F value (6.89) is greater than F critical value (2.12) and it can be observed from the findings that the values are statistical significant. The P value of 5.7410E-07 has a small percentage error and is well below 0.05. The null hypothesis (H0) is rejected that states that there are no differences among quality attribute mean values and accept the alternative hypothesis(H1) that quality attributes are different as observed in the summary shown in Table XII.

ANOVA summary for Quality Attributes (QA)

The results for Budget TD type hypothesis are shown in Table XII below. The summary has seven factors: Usability, Functionality, Reliability, Performance, Security, Compatibility and Portability.

Table XII: Analysis of Quality Attributes and related hypothesis

	Usability	Functionality	Reliability	Performance	Security	Compatibility	Portability
Mean	3.75	4.09	4.44	4.40	4.88	3.58	2.86
Confidence Level (95.0%)	5.44	3.05	2.32	3.20	2.47	2.47	5.59
Sum	2.14	233	253	251	278	204	163
Overall Percentage	13.41%	14.6%	15.85%	15.73%	17.42%	12.78%	10.21%

The cross-tabular analysis confirms Security (17.42%) as an important attribute that should not be compromised when TD type is prioritised. This is followed by Reliability (15.85%) and Performance (15.73%) respectively. The top three quality attributes have relatively the same means of 4.57 combined.

M. Correlation between architecture and security type

Table XIII below shows Pearson's R is 0.18 in respect of the correlation between budget (architecture) estimated and architecture debt type chosen. This indicates a weak correlation.

Table XIII: Architecture type and security factor correlation

	Security	Architecture Debt
Security	1	
Architecture Debt	0.18242705	1

N. Research Question 2

Which aspects of the Risk Management Framework will be useful to prioritise TD types?
Hypothesis 4: This hypothesis sought to determine if IT risk variable values are significant to each other.

H0: There are no differences in mean values for IT risk variable
H1: There are differences in mean values for IT risk variable

The ANOVA single factor was performed as output under supplementary files (**IT Risk Framework variables (ANOVA) Statistical Output)** with summary and ANOVA section. The values between the technical debt types (groups) are interpreted as follows:

The F value 4.82 is greater than F critical value (2.03) and it can be observed from the findings that the values are statistically significant. The P value of 0.000029577 has a small percentage error and is well below 0.05. The null hypothesis (H0) is rejected that states that there are no differences in mean values for the IT risk variable and accepts the alternative hypothesis (H1) that the mean values are different as observed.

The ANOVA summary

The IT risks framework hypothesis summary results are shown in Table XIV below. The summary has eight factors that are considered such as: IT Strategy, Project Management, Information/Cyber Security, Service Delivery and Operations, Systems Development Lifecycle, Service Continuity Management, Financial management and Talent Management.

Table XIV: Analysis of IT Risks factors and related hypotheses

	IT Strategy	Project Management	Informat-ion/Cyber Security	Service Delivery and Operations	Systems Development Lifecycle	Service Continuity Management	Financial Management	Talent Management
Mean	5.41	4.41	4.83	4.71	5.03	3.41	4.33	3.86
Confidence Level (95.0%)	6.91	5.09	4.25	4.04	3.40	3.83	4.93	7.35
Sum	314	256	280	273	292	198	251	224
Overall Percentage	15.04%	12.26%	13.41%	13.07%	13.98%	9.48%	12.02%	10.73%

The above cross-table analysis was used to interpret the numerical values of IT risks scores to prioritise TD. The summary table confirms that IT Strategy will be a useful risk to prioritise TD with 15.4% as a leader from the analysis. The second and third position was Systems Development Lifecycle (13.98%) and Information/Cyber Security (13.41%) respectively.

O. Correlation between architecture type and IT Strategy

Pearson's R is 0.09 on the correlation below (Table XV) between budget (architecture) estimated and architecture debt type chosen. This indicates a weak correlation.

Table XV: Architecture type and IT strategy correlation

	Architecture Debt	IT Strategy
Architecture Debt	**1**	
IT Strategy	**0.086882153**	**1**

VII. Delimitation of the Research

As indicated above, the research is firstly limited to the financial services industry and secondly, only gives focus to the technical debt in a South African context. South African financial services have sufficiently large and diversified financial service providers for the research and this study is generalized to fit technical debt.

VIII. Assumptions

The following assumptions were made regarding this study:

- Risk is universal in the financial service industry and is relative to the size of the organisation (for example, start-up or mature) and the portfolio of IT projects (small or large, for example).

- Technical debt is also known by alternative words that refer to either backlog, code debt or design debt.

- There is no secret or sensitive information pertaining to the research as the participants who consulted decision-makers in each organisation early in the research process have cleared all such areas of concern.

- All the respondents in the study were assumed to have some understanding of technical debt along the lines of the introduction given at the beginning of the questionnaire.

IX. IMPLICATIONS

This study offers a two-pronged approach with both academic and practical components. The literature shares some academic light on TD definition, TD taxonomy and context on different industries. However, it is noteworthy that most of the TD research has been done on developed countries with only scant research on the South African financial services industry (Kruchten, Nord,Ozkaya & Falessi, 2013). This study, therefore, has aimed to raise awareness and fill the void by expanding the existing literature to African countries.

The practical component has implications for management and employees who deal with technical debt based on their strategic objectives. The implication of this study is that they need to prioritise architecture items in their organizations by incorporating IT strategy as a supporting driver to ensure that security as a quality attribute is not compromised(Lane,Koolmanojwong & Boehm, 2013). Managements need to promote and encourage architectural resources to encourage a culture of treating TD as a priority. Therefore, this implies holistic review of current strategies is made by all management (Kruchten,Nord,Ozkaya & Visser,2012).

Teamwork and cooperation can increase efficiencies if the TD prioritization is done in the right way. Incorporating this hypothesized model to prioritise TD brings about incremental and transformational change in the organization for positive results. There is also a need to look at other values across all the variables that are highly significant but were not the overall majority choice.

X. VALIDITY

Four concerns were addressed to ensure that the results of the TD prioritization would not be affected by the biased selection of specific TD types. The four concerns were as follows:

- Incompleteness of TD study search - One concern for this study was that there may have been some relevant literature that had not been read or retrieved which could affect the completion of the selected TD types. In mitigation of this concern, all popular journals were searched, as well as relevant book chapters on the electronic database. Google scholar was used to validate the completeness of what was found. When some literature termed TD differently (e.g. backlog or refactoring) this data was duly followed up (Kruchten, 2012).
- Bias of TD selection - Some researchers in the literature may have been biased towards the most popular TD selection e.g 'code', 'design' and 'architecture'. It appears that other researchers have some differences in understanding criteria but predominantly it was found that the above three terms were favoured (Kruchten, Nord & Ozkaya, 2012).
- Data extraction inaccuracy - The classification and analysis phase of data extraction may often show incorrect results. To reduce the bias, a measure was employed by performing a data extraction pilot with this researcher's supervisor to root out possible disagreements before proceeding further with the analysis.
- Bias on data synthesis - Most of the studies in work related section did not disclose all the detailed information from the start. i.e some studies mentioned prioritization of TD using IT risks or quality attributes without providing descriptions, the method applied or any definitions(Lassenius, 2014).

In summary, this research can be defined as quantitative study as the aim was to prioritise different TD with an assistance of hypothesis defined in the research design section under sub-questions. Data collection method was employed was a questionnaire using Qualtrics tool to capture and store the data.

XI. CONCLUSIONS AND FUTURE WORK

The research questions and sub-questions defined in the research study design directed the research process to evaluate how TD can be prioritised in the South African financial services.

This study concludes by acknowledging other prioritization approaches outlined in work related section. This section therefore makes recommendations for management to enable new insights on aspects that can be used to prioritise TD. Suggestions then follow for future study avenues worthy of research by using this study as a reference.

This study postulates a positive relationship between chosen variables to prioritise TD in all the four hypotheses as follows:

(H1) There is a statistical significance to the selected TD types from respondents' answers and there is a small error percentage on the observed sample using the ANOVA analysis. From the results, architecture debt type is more dominant because the p-value calculates less than 0.05 and has higher mean and percentage value compared to other TD types.

(H2) There is a statistical significant relationship among the TD budget types from the respondents' answers using the ANOVA analysis. From the p-value of TD budget, the null hypothesis was rejected, as there are different mean values of budget estimated on the variable. The p-value is less than 0.05, which indicates a relatively strong relationship.

(H3) There is a statistical significant relationship among the Quality Attributes (QA) from the respondents' answers using the ANOVA analysis. From the p-value of QA, the null hypothesis was rejected, as there are different mean values of QA variable. The p-value is less than 0.05, which indicates a relatively strong relationship.

(H4) There is a statistical significant relationship among the IT risk factors from the respondents' answers using the ANOVA analysis. From the p-value of IT risk factors, the null hypothesis was rejected, as there are different mean values. The p-value is less than 0.05, which indicates a relatively strong relationship.

The results support all the hypotheses (H1, H2, H3 and H4). Further correlation was performed on the variables with the highest value against architecture debt type. All the correlation values for these variables showed a weak correlation.

Management will need to find other supporting creative ways of introducing methods or aspects of prioritising TD in order to reduce the backlog strategically (Johansson & Widell, 2014).

Suggestions for Further Research

This study makes a significant contribution to management practice in the South Africa financial services for how TD is prioritised by adding new dimensions (IT risks, quality attributes and budget). There are, however, limitations to the study which can further open up other avenues of research. Firstly, the data is gathered from different employees in the financial services. In this regard, management and IT employees could have been narrowed down for more informative results (Klinger, Tarr, Wagstrom & Williams, 2011).

Employees need to adopt a framework model at their level. This would prioritise well related TD activities suggested by management, who are major stakeholders in the value chain of TD activities.

Another area to explore would be to collect data from more practised shareholder respondents of TD instead of those with only a general understanding.

Lastly, future research could also look into the alternative TD types that were least prioritised as they also have an impact to some degree on how the organization manages its day-to-day activities.

Recommendations

According to the respondent's data collected, the results from the study show that architecture debt is by far the most prioritised. The prioritization process of TD can be carried out by different approaches as listed in work related section. In addition, a plethora of other aspects exist that can be considered for prioritising TD. The purpose of this study only focused on IT risk factors and quality attributes, therefore, an uptake on this aspect could be taken in the future in the South African financial services or other sectors.

If management is to benefit from the outcome of this study, they will need to mobilise various departments in their organization to assist in the prioritization of key technical debt types across the business area. In addition, they should further adopt automated technical innovation to track and monitor such prioritised items (Krishna & Basu 2012).

REFERENCES

Akbarinasaji, S., & Bener, A. (2016, 4-4 Oct. 2016). Adjusting the balance sheet by appending technical debt. Paper presented at the 2016 IEEE 8th International Workshop on Managing Technical Debt (MTD).

Allman, E., 2012. Managing Technical Debt. Commun.ACM, 55(5), pp. 50-55.

Alves, N.S., Ribeiro, L.F., Caires, V., Mendes, T.S. and Spínola, R.O., 2014. Towards an Ontology of Terms on Technical Debt, Managing Technical Debt (MTD), 2014 Sixth International Workshop on 2014, IEEE, pp. 1-7.

Alves, N. S. R., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F., & Seaman, C. (2016). Identification and management of technical debt: A systematic mapping study. Information and Software Technology, 70, 100-121. doi:10.1016/j.infsof.2015.10.008

Alzaghoul, E. and Bahsoon, R., 2013. CloudMTD: Using real options to manage technical debt in cloud-based service selection, Managing Technical Debt (MTD), 2013 4th International Workshop on 2013, pp. 55-62.

Alzaghoul, E. and Bahsoon, R., 2013. Economics-Driven Approach for Managing Technical Debt in Cloud-Based Architectures, Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on 2013, pp. 239-242.

Alzaghoul, E. and Bahsoon, R., 2014. Evaluating Technical Debt in Cloud-Based Architectures Using Real Options, Software Engineering Conference (ASWEC), 2014 23rd Australian 2014, pp. 1-10.

Ammerlaan, E., Veninga, W. and Zaidman, A., 2013. Old Habits Die Hard: Why Refactoring for Understandability Does Not Give Immediate Benefits}, Proceedings of the 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER)} 2013.

Apuke, O. D. (2017). Quantitative research methods: A synopsis approach. Arabian Journal of Business and Management Review, 6(11), 40-47. doi:10.12816/0040336

Berenbach, B., 2014. On Technical Credit. Procedia Computer Science, 28(0), pp. 505-512.

Besker T, Martini A, Lokuge R. E., Blincoe K., Bosch J., Embracing technical debt, from a startup company perspective, in: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME),pp. 415{425.

Board, E.-S. S. (2004). Standard for software engineering - Software life cycle processes - Risk management. ISO/IEC 16085:2004 (E) IEEE STD 1540-2001, 1-30. doi:10.1109/IEEESTD.2004.6298075

Cambridge University study states software bugs cost economy $312 billion per year. [Online].
Available:http://www.prweb.com/releases/2013/1/prweb10298185.htm

Charalampidou, S. (2019). Managing technical debt through software metrics, refactoring and traceability. [Groningen]: University of Groningen.

Codabux, Z. and Williams, B., 2013. Managing technical debt: An industrial case study, Managing Technical Debt (MTD), 2013 4th International Workshop on 2013, pp. 8-15.

Codabux, Z., & Williams, B. (2013, 20-20 May 2013). Managing technical debt: An industrial case study. Paper presented at the 2013 4th International Workshop on Managing Technical Debt (MTD).

Curtis Bill,Sappidi Jay,Szynkarski Alexandra: Estimating the Principal of an Application's Technical Debt. IEEE Software 29(6): 34-42 (2012)

Deloitte. (2018). What board members need to know — and do. In Information technology risks in financial services, 1, 2-8.

De Groot, J., Nugroho, A., Back, T. and Visser, J., 2012. What is the value of your software? Managing Technical Debt (MTD), 2012 Third International Workshop on 2012, pp. 37-44.

Dos Santos, P., Varella, A., Dantas, C. and Borges, D., 2013. Visualizing and Managing Technical Debt in Agile Development: An Experience Report. 149, pp. 121-134.

Lim, E., Taksande, N. and Seaman, C. B. "A Balancing Act: What Software Practitioners Have to Say about Technical Debt," IEEE Software, vol.29(6), 2012.

Eisenberg, R.J., 2012. A Threshold Based Approach to Technical Debt. SIGSOFT Softw.Eng.Notes, 37(2), pp. 1-6.

Falessi, D., Shaw, M. A., Shull, F., Mullen, K., & Keymind, M. S. (2013, 20-20 May 2013). Practical considerations, challenges, and requirements of tool-support for managing technical debt. Paper presented at the 2013 4th International Workshop on Managing Technical Debt (MTD).

Falessi, D., Kruchten, P., Nord, R.L. and Ozkaya, I., 2014. Technical Debt at the Crossroads of Research and Practice: Report on the Fifth International Workshop on Managing Technical Debt. SIGSOFT Softw.Eng.Notes,39(2), pp. 31-33.

Fernández-Sánchez, C., Garbajosa, J., & Yagüe, A. (2015, 2-2 Oct. 2015). A framework to aid in decision making for technical debt management. Paper presented at the 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD).

Fowler, M. (1999). Refactoring: Improving the Design of Existing Code. Boston, MA, USA: Addison-Wesley. ISBN: 0-201-48567-2

Fraser, S., Mancl, D., Opdyke, B., Bishop, J., Kathail, P., Lacar, J., Ozkaya, I. and Szynkarski, A., 2013. Technical Debt: From Source to Mitigation, Proceedings of the 2013 Companion Publication for Conference on Systems, Programming, \&\#38; Applications: Software for Humanity 2013, ACM, pp. 67-70.

Griffith, I., Izurieta, C., Taffahi, H. and Claudio, D., 2014. A Simulation Study of Practical Methods for Technical Debt Management in Agile Software Development, Proceedings of the 2014 Winter Simulation Conference 2014, IEEE Press, pp. 1014-1025.

Griffith, I., Reimanis, D., Izurieta, C., Codabux, Z., Deo, A. and Williams, B., 2014. The Correspondence between Software Quality Models and Technical Debt Estimation Approaches, Managing Technical Debt (MTD), 2014 Sixth International Workshop on 2014, IEEE, pp. 19-26.

Guo, Y., Seaman, C., Zazworka, N. and SHULL, F., 2010. Domain-specific Tailoring of Code Smells: An Empirical Study, Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 22010, ACM, pp. 167-170.

Guo, Y. and Seaman, C., 2011. A Portfolio Approach to Technical Debt Management, Proceedings of the 2Nd Workshop on Managing Technical Debt 2011, ACM, pp. 31-34.

Guo, Y., Spínola, R.O. and Seaman, C., 2014. Exploring the costs of technical debt management–a case study. Empirical Software Engineering, , pp. 1-24.

Ho, T.T. and Ruhe, G., 2014. When-to-release decisions in consideration of technical debt, Managing Technical Debt (MTD), 2014 Sixth International Workshop on 2014, IEEE, pp. 31-34.

Holvitie, J. and Leppanen, V., 2013. DebtFlag: Technical debt management with a development environment integrated tool, Managing Technical Debt (MTD), 2013 4th International Workshop on 2013, pp. 20-27.

Holvitie, J., 2014. Software implementation knowledge management with technical debt and network analysis, Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on 2014, IEEE, pp. 1-6.

Holvitie, J., Leppanen, V. and Hyrynsalmi, S., 2014. Technical Debt and the Effect of Agile Software Development Practices on It-An Industry Practitioner Survey, Managing Technical Debt (MTD), 2014 Sixth International Workshop on 2014, IEEE, pp. 35-42.

Izurieta, C., Griffith, I., Reimanis, D. and Luhr, R., 2013. On the Uncertainty of Technical Debt Measurements, Information Science and Applications (ICISA), 2013 International Conference on 2013, pp. 1-4.

Izurieta, C., Vetrò, A., Zazworka, N., Cai, Y., Seaman, C., & Shull, F. (2012, 5-5 June 2012). Organizing the technical debt landscape. Paper presented at the 2012 Third International Workshop on Managing Technical Debt (MTD).

Johansson, A. and Widell, C., 2014. Evaluating future IT-investment options through and theory testing approach on Company AB. 58 technology debt-A case study

Kalm, D. P. (2018). Technical debt - What is it costing your company. 1(1), 3-10.

Kar, S. S. (2013). Is-30-the-magic-number-issues-in-sample-size-estimation(1),175-179.

Kim, M., Cai, D. and Kim, S., 2011. An Empirical Investigation into the Role of API-level Refactorings During Software Evolution, Proceedings of the 33rd International Conference on Software Engineering 2011, ACM, pp. 151-160.

Kim, M., Zimmermann, T. and Nagappan, N., 2012. A Field Study of Refactoring Challenges and Benefits, Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering 2012, ACM, pp. 50:1-50:11.

Klinger, T., Tarr, P., Wagstrom, P. and Williams, C., 2011. An Enterprise Perspective on Technical Debt, Proceedings of the 2Nd Workshop on Managing Technical Debt 2011, ACM, pp. 35-38.

Krishna, V. and Basu, A., 2012. Minimizing Technical Debt: Developer's viewpoint, Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012), International Conference on 2012, pp. 1-5.

Krishna, V. and Basu, A., 2013. Software Engineering Practices for Minimizing Technical Debt.

Kruchten, P., 2012. Strategic Management of Technical Debt: Tutorial Synopsis, Quality Software (QSIC), 2012 12th International Conference on 2012, pp. 282-284.

Kruchten, P., Nord, R.L. and Ozkaya, I., 2012. Technical Debt: From Metaphor to Theory and Practice. Software, IEEE, 29(6), pp. 18-21.

Kruchten, P., Nord, R.L., Ozkaya, I. and Falessi, D., 2013. Technical Debt: Towards a Crisper Definition Report on the 4th International Workshop on Managing Technical Debt. SIGSOFT Softw.Eng.Notes, 38(5), pp. 51-54.

Kruchten, P., Nord, R.L., Ozkaya, I. and VISSER, J., 2012. Technical Debt in Software Development: From Metaphor to Theory Report on the Third International Workshop on Managing Technical Debt. SIGSOFT Softw.Eng.Notes, 37(5), pp. 36-38.

Lane, J.A., Koolmanojwong, S. and Boehm, B., 2013. 4.6. 3 Affordable Systems: Balancing the Capability, Schedule, Flexibility, and Technical Debt Tradespace, INCOSE International Symposium 2013, Wiley Online Library, pp. 1385-1399. 59

Lassenius, K.R.C., 2014. PACING SOFTWARE PRODUCT DEVELOPMENT: A Framework and Practical Implementation Guidelines.

Leppänen, M., Lahtinen, S., Kuusinen, K., Mäkinen, S., Männistö, T., Itkonen, J. et al.. (2015, 2-2 Oct. 2015). Decision-making framework for refactoring. Paper presented at the 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD).

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. Journal of Systems and Software, 101, 193-220. doi:10.1016/j.jss.2014.12.027

Martini, A., Besker, T., & Bosch, J. (2018). Technical debt tracking: Current state of practice. Science of Computer Programming, 163, 42-61. doi:10.1016/j.scico.2018.03.007

Power, K., 2013. Understanding the impact of technical debt on the capacity and velocity of teams and organizations: viewing team and organization capacity as a portfolio of real options. In: 2013 4th International Workshop on Managing Technical Debt (MTD), pp. 28–31. doi: 10.1109/MTD.2013.6608675 .

Ramasubbu, N., & Kemerer, C.F. (2016). Technical debt and the reliability of enterprise software systems: A competing risks analysis. Management Science, 62(5), 1487-1510. doi:10.1287/mnsc.2015.2196

Reimanis, D., & Izurieta, C. (2016, 4-4 Oct. 2016). Towards assessing the technical debt of undesired software behaviors in design patterns. Paper presented at the 2016 IEEE 8th International Workshop on Managing Technical Debt (MTD).

Snipes, W., Robinson, B., Guo, Y., & Seaman, C. (2012, 5-5 June 2012). Defining the decision factors for managing defects: A technical debt perspective. Paper presented at the 2012 Third International Workshop on Managing Technical Debt (MTD).

Szabados, K., & Kovács, A. (2015, 2-2 Oct. 2015). Technical debt of standardized test software. Paper presented at the 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD).

Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt. Journal of Systems and Software, 86(6), 1498-1516. doi:10.1016/j.jss.2012.12.052

van Teijlingen, E., & Hundley, V. (2002). The importance of pilot studies. Nurs Stand, 16(40), 33-36. doi:10.7748/ns2002.06.16.40.33.c3214

Vogel-Heuser, B., Rösch, S., Martini, A., & Tichy, M. (2015, 2-2 Oct. 2015). Technical debt in Automated Production Systems. Paper presented at the 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD).

Zablah, R., & Murphy, C. (2015, 2-2 Oct. 2015). Restructuring and refinancing technical debt. Paper presented at the 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD).