Technical Debt Management in the public sector

By

Mille Edith Kjærsgaard Hansen
IT University of Copenhagen
Computer Science
2022

A dissertation submitted in compliance with the requirements for the degree of
DOCTOR OF PHILOSOPHY (Ph.D.)

Academic supervisors
Mircea Filip Lungu, IT University of Copenhagen
Christian Østergaard Madsen, IT University of Copenhagen

Committee
Hanne Westh Nicolajsen, IT University of Copenhagen
Helle Zinner Henriksen, Copenhagen Business School
Johan Magnusson, Gothenburg University

IT UNIVERSITY OF COPENHAGEN

Acknowledgments

A Ph.D. dissertation is not created in a vacuum; I have many people to thank for their help along the way. First, I would like to thank Jens Schmidt, Head of the Research Center for Government IT, for creating the Center and gaining funding from the Digitization Agency, IT University of Copenhagen, and ATP. Thank you for giving me the opportunity to conduct a Ph.D. in your research center.

Thank you to my main supervisor, Mircea Filip Lungu. Your support, openness, and patience enabled me to explore the “Technical Debt” concept from a different perspective. A special thank you to my co-supervisor, Christian Østergaard Madsen, for introducing me to the digital government field, for our many inspiring conversations, and for pushing, encouraging, and believing in me. Moreover, thank you for our collaboration and sharing your research experiences: writing is indeed rewriting.

I want to thank the anonymous Agency X for agreeing to be a case in my research and to the employees who participated and were curious and open to the research. I look forward to the conversations to come.

I am also much indebted to the rest of the Research Center for Government IT for their support; I cherished your help, conversations, and reflections. Also, thank you to my colleague and co-author Søren Skaarup for our conversations. Thank you to the midway committee: Paolo Tell, Oliver Krancher, and Lars Rune Christensen, for their thoughtful comments and suggestions. Your feedback has contributed significantly to the development of my arguments and the shaping of the project. Thank you to Yvonne Dittrich for our initial talks encouraging me to find the right topic and perspective. Thank you to my fellow Ph.D. students, Ph.D. support staff, and the Ph.D. school at ITU for your support and for providing a virtual community in the (sometimes) lonely COVID times.

Thank you to the digital government community; I am grateful for the support from and warm welcome to the community. Thank you to my colleagues at Linköping University in Sweden for hosting my stay abroad: Ulf Melin, Ida Lindgren, and Jonathan Crusoe. Thank you for your engaging conversations, collaboration, and feedback. Thank you, Johan Magnusson, for offering to collaborate on technical debt; if not for COVID, I am sure we could have written a fascinating article.

I want to extend my deepest thanks to my close friends, who listened and kept me sane during this process, and to my dear family for their support and encouragement. A heartfelt thanks to Jacob, who was immensely supportive and kept me grounded during challenging periods, and ensured every accomplishment was celebrated.

Finally, I want to apologize to everyone else I should have thanked but who escaped my mind in the writing these paragraphs. You know who you are. Thank you.

I dedicate this dissertation to my grandparents, Dr. Jens Kristian Nielsen (deceased 2020) and Grethe Nielsen. I follow their example of creating opportunities, taking chances, and letting events unfold to see where they might lead.

Abstract

In the past decades the government has focused on digitalizing government services leading to numerous IT systems. The society is increasingly dependent on IT systems within the public sector, creating a growing need to ensure those systems perform successfully. When features must be made available under time pressure, shortcuts and temporary fixes are likely to be introduced to new and existing IT systems; this phenomenon is called technical debt (TD). Introducing TD saves time and resources in the short run but may cause problems and require additional resources in the long run. Therefore, TD management is essential to prevent costly IT problems, and a decline in the quality of IT systems. Knowledge about TD in the public sector is limited, however, as previous research on TD management has primarily focused on the private sector and has taken a software engineering approach, where source code is the primary focus.

This dissertation comprises four research articles and a summary (kappa). It focuses on how the concept of TD can be operationalized to improve the management of IT systems in the public sector. The core of this dissertation is an embedded case study that took place over seven months at a Danish agency. I adopted an engaged scholarship approach which closely relates research to practice. Thus, this dissertation contributes to practical problem solving and developing new theoretical insights. The empirical material was collected and analyzed by applying three perspectives: stakeholder theory, IT portfolio management literature, and metaphor theory.

I found a lack of TD research in the public sector and which I address by focusing on the organizational aspects of TD and TD management in the public sector. The overarching findings are 1) TD is affected by multiple kinds of stakeholders, however the stakeholders vary and depend on the IT system; 2) TD is not affected only by internal stakeholders, but external and non-technical stakeholders also affect TD activities (e.g., TD creation and TD repayment); 3) TD can be created and resolved on an IT portfolio management level; 4) five factors which increase the likelihood of TD creation; 5) metaphors can be combined and used to comprehend and communicate a new phenomenon.

This dissertation contributes new empirical insight into TD management in the public sector, explains and predicts TD activities, and presents recommendations for practice. I find that analyzing TD on a higher level of abstraction is beneficial to understanding TD at an organizational level. I propose two frameworks: the first framework describes the development of a metaphor, and the second framework integrates TD with IT portfolio management. I reveal five factors of TD creation and recommendations for TD management. The dissertation contributes empirical insights into IT maintenance within the government, models of TD processes, and practical recommendations for how to manage TD in the public sector. Finally, it encourages future studies to 1) replicate this study in private and public organizations, 2) compare quality assessments tools' evaluation of source code with IT professionals' experiences of the system, 3) conduct a longitudinal study examining TD processes and exploring TD management strategies, and 4) explore how TD management strategies can be implemented in policy documents.

Sammenfatning

I de seneste årtier har der været et stort fokus på digitalisering af offentlige ydelser, hvilket har ført til adskillige it-systemer. Som samfund er vi afhængige af den offentlige sektors it-systemer, og der er derfor behov for at sikre at disse it-systemer er velfungerende. Der er risiko for, at der bliver taget smutveje og lavet midlertidige løsninger til nye såvel som eksisterende it-systemer. Dette fænomen kaldes teknisk gæld.

På kort sigt sparer man tid ved at introducere teknisk gæld i it-systemerne. Men på længere sigt kan teknisk gæld forårsage fejl og problemer, hvilket gør, at det samlet set kan blive dyrere at sikre velfungerende it-systemer. Det er afgørende, at man håndterer den tekniske gæld og forhindrer dyre løsninger, der kan føre til fejl, samt minimerer risikoen for, at kvaliteten i it-systemerne bliver forringet. Viden om teknisk gæld i den offentlige sektor er dog begrænset. Tidligere forskning i teknisk gæld har fokuseret på den private sektor og 'open source' projekter. Dertil er teknisk gæld primært undersøgt fra en softwareudviklings-vinkel, hvor selve kildekoden har været det primære fokus.

Denne afhandling består af fire forskningsartikler og en kappe, der både introducerer og binder de fire artikler sammen. Afhandlingen fokuserer på, hvordan konceptet teknisk gæld kan operationaliseres for at forbedre styringen af it-systemer i den offentlige sektor. Afhandlingens kerne består af et casestudie, der fandt sted i en dansk styrelse over en syv måneders periode. Jeg har valgt 'Engaged Scholarship'-tilgangen, hvor forskningen laves med praktikere. Dermed bidrager afhandlingen til praktisk problemløsning samt udvikling af teoretiske indsigter. Efter indsamlingen af det empiriske materiale, analyserede jeg det fra tre forskellige perspektiver: Stakeholderteori, it-porteføljestyringslitteratur og metafor teori.

I mit litteraturstudie identificerede jeg mangel på teknisk gæld forskning i den offentlige sektor. Jeg adresserer dette ved at fokusere på de organisatoriske aspekter af teknisk gæld og styring af teknisk gæld i den offentlige sektor. De overordnede resultater er 1) teknisk gæld er påvirket af forskellige typer af interessenter, dog varierer interessenterne fra it-system til it-system; 2) teknisk gæld er ikke kun påvirket af interne interessenter, men også af eksterne og ikke-tekniske interessenter; 3) teknisk gæld kan opstå og løses på et it-portefølje-styringsniveau; 4) fem faktorer som kan øge sandsynligheden for at skabe teknisk gæld; 5) metaforer kan kombineres og bruges til at forstå og kommunikere et nyt fænomen.

Denne afhandling bidrager med ny empirisk indsigt i styringen af teknisk gæld i den offentlige sektor, forklarer og forudsiger teknisk gæld aktiviteter, og præsenterer anbefalinger til praktikere. Jeg viser, at analyse af teknisk gæld på et højere abstraktionsniveau er gavnligt for at forstå teknisk gæld på et organisatorisk niveau. Jeg foreslår to rammeværk: Det første kategoriserer udviklingen af en metafor, og det andet integrerer teknisk gæld med IT-porteføljestyring. Jeg præsenterer årsager til opbygning af teknisk gæld og giver anbefalinger til ledelsen om håndtering af teknisk gæld. Afhandlingen bidrager med empirisk indsigt i it-vedligeholdelse i en dansk styrelse, modeller af teknisk gæld processer, og praktiske anbefalinger til, hvordan man kan styre teknisk gæld i den offentlige sektor. Til sidst, opfordrer jeg fremtidige forskere til at 1) gentage mit studie i private og offentlige organisationer, 2) sammenligne kodekvaliteten med IT professionelles erfaring med systemet, 3) gennemføre et tidsstudie om tekniske gæld processer og styringsstrategier, og 4) undersøge implementering af styring af teknisk gæld strategier i politiske dokumenter.

Overview of research articles

This is an article-based Ph.D. dissertation that consists of four articles and a summary (kappa) to the research conducted throughout the Ph.D. This section provides an overview of the articles and their status.

Article 1: Nielsen, M. E., Madsen, C. Ø., and Lungu, M. F. (2020, August). Technical debt management: A systematic literature review and research agenda for digital government. **Status: Published** in International Conference on Electronic Government (pp. 121-137). Springer, Cham.

Article 2: Nielsen, M. E., Crusoe, J., Melin, U. Exploration of metaphors as a way to understand socio-technical phenomena: An emergent analytical framework. **Status: 2nd round of revisions** at the Journal of Information Polity June 2022

Article 3: Nielsen, M. E., Skaarup, S. (2021, October). IT Portfolio management as a framework for managing technical debt: Theoretical framework applied on a case study. **Status: Published** in Proceedings of the 14th International Conference on Theory and Practice of Electronic Governance (ICEGOV 2021). ACM Press; New York.

Article 4: Nielsen, M. E., Madsen, C., Ø (2022). Stakeholder influence on technical debt management in the public sector: An embedded case study. **Status: Published** at Government Information Quarterly.

Table of Contents

1. Introduction

Digitizing the government has led to the creation of numerous IT systems. As the public sector has digitized its operations, IT systems have proliferated. At the same time, increased dependency on IT systems within the public sector creates a growing need to ensure the systems perform and can be maintained successfully (Danish Ministry of Finance, 2017a). In presenting new features to users, the software developer is likely to introduce temporary fixes to existing IT systems (Guo & Seaman, 2011). Systems must be maintained to prevent costly IT problems, and a decline in quality. Cunningham (1992) called the reoccurring phenomenon of shortcuts and quick fixes within IT development as technical debt (TD). Shortcuts and unsuitable choices during the development or maintenance of an IT system can have negative consequences such as inefficiency and instability and can complicate the system's future development and maintenance. TD saves time and resources in the short run but may cause problems and require additional resources in the long run (Cunningham, 1992). Managing TD is challenging, whether it is seen as a local phenomenon (residing in individual applications) or in an IT portfolio management perspective. An IT portfolio management perspective involves: *"balancing resources, technology, business needs, and changing situations while simultaneously maximizing returns and minimizing risk"* (Gliedman, 2004, p. 2).

I came across the phenomenon of TD when I was working as a practitioner. I was curious about how TD increased workloads and what could be done about it. From my practical experience, I knew TD was something organizations had to address and prioritize to avoid increased costs, erroneous data, and other problems. I assumed TD was barely researched and relatively new because the concept was limited to the very technical employees. The research has primarily focused on open-source projects and the private sector. Most TD studies have been conducted outside of the digital government field, primarily by researchers in the fields of software engineering and information systems. However, rapid digitalization has resulted in increased TD in the public sector as well. For instance, in 2015, a third of the Danish government's society or business-critical IT systems were found to be inadequate (Danish Ministry of Finance, 2017a). Similarly, the Swedish National Audit Office (2019) found that 70% of the IT systems in Sweden's government agencies were outdated. TD is a relevant research topic in digital government because it can hinder public organizations from taking full benefit from digitalization (Nielsen, Madsen & Lungu, 2020).

This Ph.D. project aims to explore TD management within the public sector in Denmark. This dissertation addresses the lack of TD research in the public sector by researching TD management in a Danish agency. Furthermore, I hope this research inspires other scholars to expand on my findings. Denmark is ranked among the top countries for digital government adoption (UNDESA, 2017; Tinholt et al., 2018). Therefore, it seems fitting that the Danish government should become a role model for other countries in addressing and managing TD. I hope this dissertation can be a step in this direction.

1.1 Research questions

Motivated by my practical experiences, I wanted to address *how* TD can managed in the public sector. As bringing about change is my intention with the dissertation, I posed "how" questions (Blaikie & Priest, 2019). Therefore, I chose a practical perspective and followed an engaged scholarship approach, which allows researchers to contribute to practical problem-solving (Mathiassen, 2017). The Ph.D. project was guided by the following overarching research question:

RQ1: How can the concept of technical debt be operationalized to improve the management of IT systems in the public sector?

To structure the research process, I developed four sub-questions that investigate various aspects of the overarching research question:

RQ2: How is technical debt defined and studied in the existent academic literature?
RQ3: How and why is technical debt created in a public organization?
RQ4: How does technical debt become visible in a public organization?
RQ5: How can technical debt management become part of IT portfolio management?

To investigate these research questions, I conducted longitudinal research adopting a qualitative approach. More specifically, I conducted an embedded case study over seven months, using a combination of interviews, observation, and study of documents related to IT management. These techniques enabled me to explore TD management from various perspectives ranging from IT portfolio management to software development.

As I conducted a single case study, this dissertation does not seek to make generalizations, but future researchers may repeat this study, as I will return to in section 10.4. My objective was to provide empirical insight into the complexity of TD management in the public sector and propose recommendations for practice and research. My research is positioned within the field of digital government for multiple reasons, most importantly because TD management is necessary for evolving existing and adding new systems to society.

1.2 Reading guide

This dissertation consists of four articles, and a summary (kappa). It has the following structure. Chapter 2, introduces the setting of the research, followed by a background chapter of TD literature and digital government field, partially based on the work of Article 1. Chapter 4 presents the theoretical frameworks of the dissertation by introducing engaged scholarship, and the key concepts from stakeholder theory, IT portfolio management, and metaphor theory. Chapter 5, on method, elaborates on engaged scholarship, embedded case study, and the data collection techniques applied in this research. Chapter 6 gives a brief note on the development of the articles. Chapter 7 presents articles 2, 3, and 4, which I advise the reader to read in full. Additionally, it presents an analysis of two IT systems that are not covered in Article 4. Chapter 8 provides overarching findings of the research and general tendencies of TD processes. Chapter 9 emphasizes the empirical, theoretical, and practical contributions and relate them to previous literature. Lastly, Chapter 10 covers the conclusions and limitations of this study and offers suggestions for future work.

2. Research setting

This chapter presents an overview of the research setting, including: an overview of the Danish public sector (2.1); a historical description of how the Danish public sector has been modernized and digitized (2.2); an overview of digitization in the public sector (2.3); the difference between public and private organizations (2.4); a case description of Agency X (2.5); and lastly, the consequences TD can have on the public sector (2.6).

2.1 Overview of the Danish public sector

This subchapter provides an overview of the Danish public sector and how it is divided into parliaments, ministries, agencies, public organizations, municipalities, and regions. I developed figure 1 to display the different layers of the public sector and illustrates the complexity of its different institutions and stakeholders. The first level shows the Danish and European parliaments; they make the legislation that the public organizations follow. The second level displays the ministries, municipalities, and regions; and below the ministries, we find agencies and public organizations. Certain agencies and public organizations answer to a specific ministry. However, interactions also take place across ministries, municipalities, and regions. Furthermore, some agencies are more affected by the European Parliament than others.

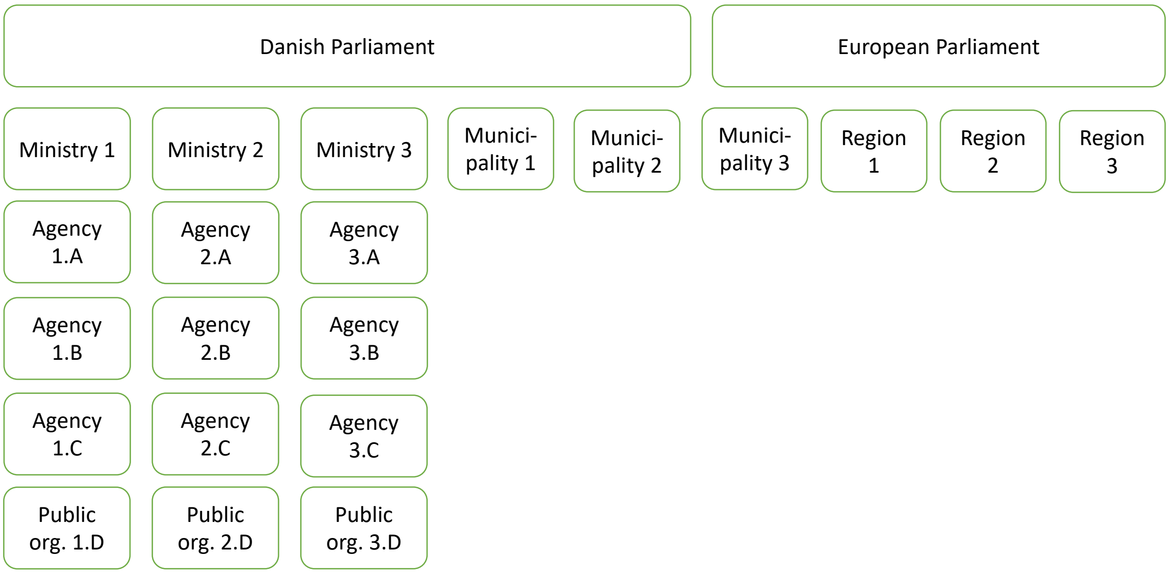


Figure 1: Overall organization of the Danish public sector

In 2007, the Danish public sector underwent a municipality amalgamation reform, which reduced the number of municipalities and changed their service responsibilities (Blom-Hansen, 2012). Because digitization is perceived as a collaborative effort, the Danish central government, regions, and local municipalities have developed digitization strategies in a cross-sectional manner (Jæger & Löfgren, 2010). This will be further elaborated in the next section.

2.2 Modernizing and digitizing the Danish public sector: a historical perspective

In this section, I will describe the history of digitization in the Danish public sector. The main characteristics of this process are top-down management and collaboration across the public sector (i.e., municipalities, agencies, and regions) (Jæger, 2020). The country has had several digitization strategies; the first launched in 2001 (Agency for Digitisation, n.d.). In line with the digitization programs seen in other countries, the overall purpose of these strategies has been to increase efficiency and reduce costs (Jæger & Löfgren, 2010; Schou & Hjelholt, 2019).

At the end of 1950s, the Danish government created the Huldatacentralen, later known simply as the Datacentralen ('Data Center') (Ejersbo & Greve, 2014). Subsequently, in 1968, the government digitized Danish citizens' CPR numbers (ID numbers). This had a minor impact on citizens, as the recorded data was limited. Furthermore, it was challenging at the time to move data and perform data exchange (Jensen, 2021). The Data Center was converted into a state-owned stock company to compete with other suppliers (Ejersbo & Greve, 2014). Mærsk (a Danish shipping firm) bought a percentage of the stock company, and in the 1990s, the Data Center was completely privatized. In 1996, the American concern Computer Science Association—known today as CSC Denmark—bought the residual company. The government refrained from developing IT systems and programs because private actors were thought to be more qualified (Ejersbo & Greve, 2014). After privatizing the Data Center, the government stopped being the owner of organizations in this area (Ejersbo & Greve, 2014).

The Danish public sector has been through several modernization programs since the 1980s. First, in 1983, there was a need to get the public finances in order. Then, in the mid-1980s, the focus shifted to IT, which was perceived as a tool for modernizing the public sector. At the time, public sector reforms entailed centralizing and merging larger tasks to achieve efficiency and look ahead to the future. As part of these efforts, Denmark invested heavily in bringing public administration into the Information Age and creating digital administration. Between 1980 and 2013, the central aim of public sector management was efficiency, which was complimented by a focus on sound finances and robustness against future challenges (Ejersbo & Greve, 2014). During this period, public administration policy primarily took the form of modernization initiatives. As public sector management developed, policy gradually became more sophisticated and nuanced (Ejersbo & Greve, 2014).

In the 1990s, telecommunication was liberalized, which opened up possibilities for the development of IT (Ejersbo & Greve, 2014). The emergence of the Internet and other digital technologies further contributed to the growing importance of IT in the modernization agenda (Ejersbo & Greve, 2014). This period was marked by the report *Info-Society 2000* (also known as the Dybkjær report), which was published in 1994. The report confirmed the political significance of IT and outlined some of its economic implications (Jæger & Löfgren, 2010). In the interest of creating a general framework for digitization, the report also proposed several strategies for incorporating IT into the public sector and society as a whole (Jæger, 2020).

Since 2000, there has been broad agreement that the purpose of public sector digitization is to create efficiency and improve government services (Jæger, 2020; Hjelholt & Schou, 2019). In 2002, the Ministry of Finance announced that the government would transform Denmark into a modern information society and, as part of this process, would apply new technology to change work practices in the public sector. This approach had clear parallels with previous reform initiatives and programs in the Danish public sector. In 2000, a digital task force was created and placed in the Ministry of Finance. Its objective was to develop a general digitization strategy for the public sector. In 2002, there was a shift in government, with socialist parties losing power to liberal and conservative parties. This shift changed IT politics and digitization strategy in Denmark. Greater emphasis was placed on preserving Danish values, generating growth and profit, and lowering administrative costs in the public sector. The task force thus became a hardline *tæskehold* (literally, a "thug team"); this approach lasted until 2011 (Jæger, 2020). Every few years, the task force released new digitization strategies for the public sector, which significantly impacted public organizations. The Agency for Digitization argues that these strategies are the reason why Denmark has succeeded in public sector digitization (Jæger, 2020).

Digitization has been managed centrally in the Danish public sector: first by the Research Ministry, then the Ministry of Finance, and now by the Agency for Digitization (which is part of the Ministry of Finance). In general, the focus has been on increasing efficiency. For example, the government has created a new public organization, Statens IT ("State IT"), in order to centralize IT procurement and operations in the public sector. Recently, however, the focus of digitization has shifted from IT acquisitions and operations to the relation between IT and individual organizations.

In 2014, it became mandatory for all citizens to have Digital Post (formerly known as e-boks), a kind of digital postbox. People without a computer, or who lack the skills to use one, were given the option to apply for an exemption. At this point, it was already mandatory for organizations to have Digital Post (Henriksen, 2015). The extension of the program marked a shift from digitization to digital transformation as IT began to impact work processes and welfare benefits. This gave greater scope for efficiency and, in turn, financial savings. In addition to generating new welfare technologies and automating work processes, large data volumes have the potential to create new knowledge. Therefore, we can expect artificial intelligence (AI) and algorithms to increasingly support political decision-making. In 2016, the government published its latest digitization strategy. This focused on citizens' data, how the public uses it, and its possibilities (Danish Ministry of Finance et al., 2016).

In recent years, the Ministry of Finance has published two reports that give an overview of IT use in the public sector (Danish Ministry of Finance, 2017a; Danish Ministry of Finance, 2018). This overview included information on collective IT costs, IT projects with a budget over 5 million DKK, and the critical IT systems within government. The Finance Ministry published a strategy to address the issues raised in the reports, above all that critical IT systems were not adequately maintained (Danish Ministry of Finance, 2017b). In 2021, the government created a digitization council with private sector experts. This body makes recommendations on key issues like the future of the public sector and the emergence of a data-driven society (Danish Ministry of Finance, 2021).

In 2022, 92% of all Danes have access to the Internet and use it daily (Aagaard & Pedersen, 2022; Statistics Denmark, 2020). Thus, Denmark has a strong digital infrastructure, which is essential for digitizing government (Aagaard & Pedersen, 2022). Digital technologies play a key role in public sector administration, helping citizens to access and use public services, while at the same time reducing costs for the state.

To sum up, modern technology is associated with digital technologies (e.g., a computer program) and technological tools (e.g., a screen for self-service) (Høybye-Mortensen, 2011). Technology is a key part of most tasks performed by public organizations. In Denmark, digitization is essential to the work of both ministries and municipalities (Høybye-Mortensen, 2011). For over a decade, the country has been ranked as one of the most digitized in the world (Aagaard & Pedersen, 2022). Across different governments, the central ambition has been to maintain Denmark's position as a digital government frontrunner (Aagaard & Pedersen, 2022). To this end, the Danish state has used the high level of information it possesses on its citizens to move toward personalized public services. The information enables the government to use digital infrastructure for "nudging" citizens in their daily lives (Aagaard & Pedersen, 2022).

2.3 The importance of digitization in the Danish public sector

This section seeks to understand how digital administration became so important in the Danish public sector and how critical it is to the country's infrastructure. Digital administration refers to the systematic application of digital technologies to organizations and work processes in order to improve efficiency. These technologies include electronic databases (e.g. case and document databases), the Internet, e-mail, online self-service systems, intranets, and digital signatures (Ejersbo & Greve, 2014). The Danish government expects digitization to (1) contribute to horizontal coordination (i.e., better connected public services); (2) make public sector decision-making more transparent; and (3) facilitate the overall management of the public sector (Høybye-Mortensen, 2011).

Digital administration is thus crucial for the Danish government, which has launched significant planning and management initiatives to oversee its implementation (Ejersbo & Greve, 2014). However, digitization has also been accompanied by marketization (e.g., the privatization of the Data Center). Indeed, leading IT companies such as IBM, Accenture, and Navision have cooperated with the Danish government in delivering hardware and software, thereby supporting its vision of a digitized public sector (Ejersbo & Greve, 2014).

Despite their importance for the Danish government, digitization reforms have lacked systematic monitoring of their results and benefits, and it is unclear how this monitoring should take place. This is related to how digitization of the Danish public sector is limited to improving efficiency. There has not been a proper debate about how it could increase communication with citizens and facilitate a broader transformation of society (Ejersbo & Greve, 2014). Yet as digitization moves closer to citizens, public interest increases, making the legitimacy and accountability of digitization more politically important (Aagaard & Pedersen, 2022). Technology must be legitimate in the eyes of citizens and professionals in order to have a positive impact (Aagaard & Pedersen, 2022).

Digitization is fundamental to the public sector's governance agenda. At the central, municipal, and regional levels of government, many concrete initiatives have been made under different digitization strategies, boards, councils, and taskforces (Ejersbo & Greve, 2014). However, the digitization of workflows and processes has varied in degree and largely been directed from above (Ejersbo & Greve, 2014; The Public Accounts Committee & The Auditor General's Office, 2020). The top-down approach to digitization has been key to its success and led to Internet portals for citizens (borger.dk), the health service (sundhed.dk and sundhedsplatformen.dk), and businesses (virk.dk) (Aagaard & Pedersen, 2022). Several services provided by the Danish welfare system have also been digitized, including taxation, medical records, and public transport payments (Aagaard & Pedersen, 2022).

From the Danish model, we learn that digitization of the public sector requires political and administrative management (Jæger, 2020). A clear vision for digitization is essential for the creation of digital administration (Jæger, 2020). To this end, the Danish government has continuously and systematically made an effort to enhance public sector management. It chose the Ministry of Finance as the central coordinator of digitization, with finance and efficiency the guiding stars. Additionally, the government has followed a dynamic approach to digitization, enabling various interest groups and stakeholders to participate in reforms. These reforms have generated concrete results, at least in the short term, and go beyond what has been achieved by Denmark's Scandinavian neighbors, Sweden and Norway (as the amalgamation policy is unique). As a result, the reforms have attracted international attention. Most pertinently, the reforms have strengthened management capacity in the Danish public sector (Ejersbo & Greve, 2014). This is important because there are indications that the management system is decaying (Ejersbo & Greve, 2014). For example, as contracts gradually expand to include new goals and demands, they become more of a management ritual and thus begin to lose their purpose. Management impacts TD and can cause TD creation, which in turn can damage the public IT infrastructure. Therefore, it is essential to explore how the management of IT systems can be improved, particularly in terms of TD.

2.4 Differences between public and private organizations

This subchapter outlines the differences between public and private organizations, using first Berg-Sørensen et al. (2011), and then Bozeman and Bretschneider's (1994) four dimensions of public organizations.

Public and private organizations can be seen as different in several respects (Berg-Sørensen et al., 2011). Berg-Sørensen et al. (2011) state that public organizations are politically driven, whereas private organizations are typically driven by profit. As public organizations are funded by taxpayers' money, they face greater demands for transparency and the equal treatment of citizens. Despite these apparent differences, there is in fact no simple distinction between public and private organizations, particularly in light of recent developments that have brought the two closer together (Berg-Sørensen et al., 2011). The public sector has imported numerous management terms from private organizations; for example, a government body is considered an enterprise. Furthermore, in Denmark, there are hybrids like the Arbejdsmarkedets Tillægspension (ATP or, in English, the Workers' Supplementary Pension), which provide essential public services to citizens with the involvement of the private sector. Nonetheless, Berg-Sørensen et al. (2011) stress the importance of keeping the public context in mind when considering public administration.

Public organizations gain the public's attention through politicians and the media. There is a focus on response time, budget overruns, cost reductions, and the violation of rules. Typically, a public organization must solve tasks involving a range of stakeholders. For example, a government ministry must support the relevant minister's political goals, ensure legality and efficiency, be transparent in its work, and be attentive to both its employees and the citizenry (Berg-Sørensen et al., 2011). Public organizations must also manage uncertainty and change, which can be caused by the formation of a new government, the emergence of new technologies, and other factors.

Bozeman and Bretschneider (1994) argue that certain private or semi-private organizations have similar characteristics to public organizations. For example, financial organizations are strongly regulated and expected to uphold a high level of professionalism. The authors identify four dimensions that can be used to analyze the characteristics of an organization. In Section 2.5, I apply these dimensions in the description of Agency X. The dimensions are as follows:

(1) **Origin of the resources**. This depends on three variables: the extent to which resources come from government contracts and grants; the extent to which resources come from government appropriations; and the extent to which the government finances some equipment.
(2) **Control over the agenda** (e.g., through financing). If the government finances the organization, how does a lack of finance change the organization's focus and goals?
(3) **Amount of public communication** (and communication with external government actors, e.g., personnel or agencies).
(4) **Ownership of the organization**. This relates to the legal owners of an organization. Bozeman and Bretschneider (1994) argue that this classification is sometimes meaningless. They give the example of a private university: despite it being owned privately, it still has more in common with a public university than, for instance, private industrial research units.

These four dimensions can help us to identify whether the government and/or the public sector is a key stakeholder in an organization. It is important to consider this because an organizations' stakeholders may be cross-sectional. For example, Denmark's financial sector is regulated by a public agency, national legislation, and EU standards. I apply the four dimensions to ease comparison between Agency X and other organizations (see section 2.5.1).

2.5 Case description of Agency X

This section first describes Agency X on a general level: its size, collaborations, and goals. The case description is elaborated in three sub-chapters: (1) the characteristics of the Agency (2.5.1), (2) digitization's impact on the Agency's work (2.5.2), and (3) a description of the Agency's approach to IT development (2.5.3).

A mature IT organization in the Danish public sector, Agency X is considered one of the best government bodies in terms of developing and operating IT systems. The Agency maintains the stable operation of its digital services by focusing on high user satisfaction, efficient execution of the political agenda, and being open, trustworthy, and innovative.

Agency X is a large organization with between 700 and 1,100 employees (not including external consultants), who are located in different parts of Denmark. The Agency is responsible for a digital portal that displays the services offered by various government agencies in one place. Additionally, the Agency is responsible for customer service and receives instant feedback from its users. Any problems are, therefore, quickly reported to the appropriate people within the organization.

The Agency supports a specific ministerial area. However, it has a broad range of responsibilities and collaborates with other public organizations. Therefore, it has numerous stakeholders, including the Danish Parliament, businesses, citizens, and other agencies and ministries.

2.5.1 Characteristics of Agency X as a public organization

To further describe the frame of Agency X, I apply Bozeman and Bretschneider's (1994) four dimensions (as presented in Section 2.4):

(1) Origin of the resources: Agency X's resources are primarily from the Danish government. They are provided in exchange for specific public services, or according to tasks carried out on behalf of other public organizations (e.g., the REPORT system, which I analyze in Article 4). Agency X also offers fee-based services (e.g., the SERVICE system, which I analyze in sections 7 and 8). In the Danish public sector, fees are standard for certain services; for example, if you wish to change your name or acquire a passport.

(2) Control over the agenda: Overall, the Danish government is in charge of Agency X's agenda. The minister specifically responsible for the Agency uses it to carry out the Danish government's decisions and policies, as ratified by the Danish Parliament. In turn, the Agency supports the minister and, by extension, the government. As Denmark is subject to EU legislation and initiatives, the EU Parliament also has an impact on the Agency's agenda.

(3) Amount of public communication: Agency X communicates with other public organizations and private stakeholders, for example, companies hired to assist in IT development and maintenance. The Agency also communicates with citizens and enterprises via its online services and customer support. Additionally, it supports the responsible minister by providing them with relevant information when they speak to the press or Parliament.

(4) Ownership of the organization: The Danish government owns the Agency. However, there is a tradition among civil servants of being politically neutral. Thus, they remain in their posts after a change in government (Damgaard, 2004). The Danish Parliament sets the level of funding for the Agency and decides how it should be organized; for example, whether it should be merged with another agency.

Finally, as Berg-Sørensen et al. (2011) point out, several similarities exist between public organizations and large enterprises. For example, Agency X is organized similarly to a private business, with a director supported by managers. The director's responsibility is to ensure that the Agency adheres to legislation and provides the necessary services at a professional level. Like the different branches of a large enterprise, the agency must follow the general rules and guidelines set by an overarching authority—in its case, the Danish Parliament. For example, when Parliament decides that payments to external consultants need to be reduced, the Agency must implement this decision. Lastly—and unlike a private enterprise—the agency should be non-profit, so user fees should only cover the cost of administering specific services.

2.5.2 Digitization's impact on Agency X

Digitization has greatly impacted Agency X's casework and services. The Agency began by digitizing its primary services, leading to the development of back-end IT systems (e.g., the TEXT system). As legislation and ambitions changed and new technologies arose, the Agency expanded its IT landscape by digitizing other services and by modernizing its existing IT systems. Today, the agency is a frontrunner within the public sector in terms of IT development, to the extent that it provides IT development and maintenance on behalf of other public organizations. Within the Agency, digitization has led to greater efficiency and improvements in casework. For instance, it has enabled the Agency to employ new technologies, such as machine learning, in order to increase the number and effectiveness of controls in certain areas. Furthermore, digitization has eased data sharing and sped up casework across public organizations in general (e.g., REPORT, SERVICE). As more services are digitized, digital infrastructure becomes increasingly important. Therefore, the Agency has focused on stabilizing the operation of its digital services.

2.5.3 Description of Agency X's approach to IT development

The Agency has not employed its own developers; instead, it has created partnerships with software companies. IT development is carried out through collaboration between the Agency and on-site developers provided by the companies. The Agency has an extensive IT portfolio containing internally developed and off-the-shelf applications.

Agency X works actively on IT portfolio management. There are monthly meetings between the IT architects and top management, as well as quarterly two-day workshops with the IT department, top management, and business managers. The workshops are inspired by SAFe Program Increment (PI) planning, focusing on IT development and the interdependencies between projects and maintenance (Scale Agile Inc., 2021). The purpose of the workshops is to plan the next three months of IT development and maintenance and ensure dependencies between tasks, projects, and systems are considered to mitigate potential risks. The business managers are present in these workshops to help prioritize tasks, clarify needs, and answer funding questions. The workshops primarily focus on internally developed systems. The PI planning format was implemented shortly before the data collection took place, and the Agency was working toward a final format of these planning workshops.

Agency X also develops and maintains IT systems for other agencies. These agencies cover the costs, define the needs, and are responsible for the underlying legislation. The external systems become part of Agency X's IT landscape and use its back-end systems. In the thesis, I investigated four internally developed IT systems; two are presented in Article 4 (REPORT and TEXT), and two are presented in Section 7.4 (CASE and SERVICE). Internally developed IT systems can more easily be altered compared to off-the-shelf applications (Al-Ibraheem & Ruël, 2011).

2.6 Public sector and technical debt

For several decades, digitization of the public sector has taken place with the aim of achieving modernization and efficiency. In the history of the Danish public sector, IT has been long been regarded as an effective solution, with the result that some public sector IT systems are now relatively old (Danish Ministry of Finance, 2017a). These systems are difficult to replace and require their surroundings to stay the same in order to continue to work (Matthiesen & Bjørn, 2015).

The pursuit of greater efficiency, financial savings, and top-down management can cause TD creation. This is because these goals can unintentionally lead to producing more at the expense of quality, reduced time to market, shorter deadlines, and a lack of concern about maintainability (Ramač et al., 2022, Freire et al., 2020).

Moreover, technical development is increasingly complex. Much has to be taken into account: political, technical, and ethical considerations, as well as factors related to efficiency, cost reductions, economic constraints, the interests of both the organization and citizens, legal restrictions, and the desire to be a digitization frontrunner within government (Aagaard & Pedersen, 2022). These issues make IT planning more challenging and may prevent it from taking place. Lack of IT planning is another known cause of TD creation (Ramač et al., 2022, Freire et al., 2020).

The Danish public sector has outsourced IT development to private organizations because they are perceived to have a higher level of knowledge and expertise (Ejersbo & Greve, 2014). Public organizations are more inclined to knowledge loss (a cause for TD creation) as a result of the tender requirement for IT development and maintenance (IT-Branchen, 2021). Furthermore, public resources have to await approval each year, in order to ensure that they are part of the Finance Act (Agency for Public Finance and Management, n.d.). Therefore, it can be difficult to plan and to make long-term decisions and investments. This further increases the risk of TD creation.

To sum up, TD can have several negative consequences for a public organization if it is not effectively managed. It can: (1) hinder the organization in providing services to citizens; (2) impede the organization in terms of adhering to legislation; (3) prevent the organization from moving forward with new technologies or exploiting the scalability of these technologies; (4) make IT security vulnerable and increase the risk of being hacked; (5) create complexity and thus hinder transparency in terms of how the IT system works, (6) increase the costs and resources of new IT development and maintenance; and (7) decrease the efficiency of, for example, casework. Having explored digitization in the Danish public sector and explained how TD poses a risk to public sector infrastructure and services, I now turn to previous studies on digital government and TD.

3. Previous studies on digital government and technical debt

Drawing on existing literature reviews, this chapter first introduces digital government as a research field. It then introduces TD and the TD literature, using as a basis my previous literature review (Nielsen et al., 2018).

3.1 Digital government as a research field

This section draws on literature reviews and overviews of the digital government research field. I introduce digital government as a field, its multi-disciplinary aspect, the first research phase (late 1990s–2010), and finally, the second research phase (2011–today).

What is digital government?

Research into digital public administration, or digital government (previously known as e-government), is a relatively new field (Jæger, 2020). Emerging in the late 1990s with the rise of the World Wide Web (Scholl, 2015a), the field was based on the expectation that IT and digitization would be beneficial to government (Heeks & Bailur, 2007). The field thus addresses “the use of information and mobile technology to support and improve public policies and government operations, engage citizens, and provide comprehensive and timely government services” (Scholl, 2007, as quoted in Scholl, 2015a, p. 6). From the field’s beginning, scholars have been interested in measuring the progress of digital government, recording the proliferation of information systems, increases in efficiency, and other transformational effects (Scholl, 2015a). The first decade of academic digital government research focused on observing and evaluating digital government outputs. This research was primarily descriptive (Yildiz, 2007). Now, digital government has passed into a second phase, in which the digital transformation of the public sector has begun to take place (Scholl, 2020).

Information communication technology (ICT) has been utilized in government for several decades (Andersen et al., 2010). In Denmark, the government has invested in ICT with the expectation that it will create a positive change (Andersen et al., 2010). Thus, to some extent, this development has been driven by rhetoric and hope, rather by empirical knowledge of its effects. There is a firm belief that ICT can transform the public sector in areas such as internal administration, regulation, and law enforcement (Andersen et al., 2010). It is also believed to increase effectiveness, improve information quality, and lead to efficiency gains (Andersen et al., 2010). Although its long-term impact remains to be seen, Andersen et al. (2010) find that ICT has an overall positive impact on digitizing government.

Digital government mainly concerns how public organizations can use IT systems to transform their work processes and services. Thus, IT scholars tend to dominate digital government research (Lindgren, et al., 2021). The central focus is a transition from physical, paper-based work and procedures to automated processes, digital communication, and publicly available online records (Lindgren, et al., 2021). The major themes of digital government include organization management and transformation, which relate to and overlap with a focus on infrastructure and interoperability (Scholl, 2015a). Furthermore, there is a strong interest in public policy and in governance issues such as participation, inclusion, and democracy. Several scholars have criticized digital government research for its persistent focus on affirmative and confirmatory research, where scholars seek to demonstrate the positive outcomes of digital government initiatives (Scholl, 2020, Yildiz, 2007).

Digital government as a multi-disciplinary field

Among scholars in the field, there is ongoing debate about how to interpret and analyze digital government and its overall direction (Scholl, 2015b). Heeks and Bailur (2007) and Grönlund (2004) criticize digital government research for being theoretically slight. However, Scholl (2015a) claims that different rules and standards apply to this field because its research falls outside the boundaries of a single discipline. Indeed, Scholl (2015a) argues that digital government research contains six major concepts, which overlap with aspects of other disciplines (Figure 2). Thus, the scope of digital government is too large to follow a mono-disciplinary approach (Scholl, 2015a).

Digital government research draws on various strands of computer science (e.g., human computer interaction), information sciences, information systems, administrative and organizational sciences, political science, sociology, psychology, and other disciplines (Scholl, 2015a). Because of the inter-disciplinary nature of the field, there is no overarching theory. Nonetheless, by connecting the objects of study with methods and procedures from different disciplines, the field is able to produce relevant research for practitioners (Scholl, 2006a; 2015a). Digital government research spans the full spectrum of hard and soft, pure and applied science (Scholl, 2006a). After wide recognition of its multi-disciplinary nature, the field's disciplinary diversity has been regarded as a strength rather than a weakness (Scholl, 2015a).

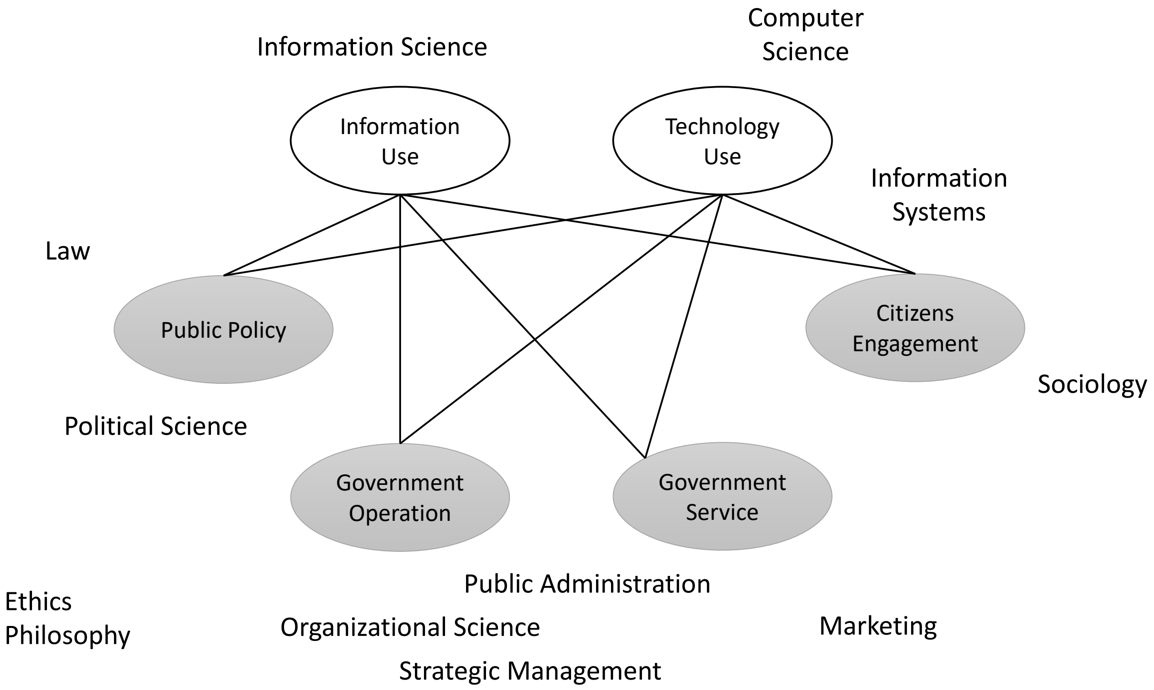


Figure 2: Relationships between the different disciplinary components of digital government (Scholl, 2007)

Digital government from its beginnings to 2010

The initial popularity of digital government has been attributed to the belief that it offered something radically new within the field of information systems (Scholl, 2006a). Yet by the early 21st century, its popularity had begun to wane, possibly due to the lack of a clear definition of what the field entailed (Scholl, 2006a). The hype surrounding digital government seemed to disappear, along with the technological optimism that originally inspired it (Heeks & Bailur, 2007).

Related to this is Scholl's (2006a) argument that academia's adoption of the terms "e-business", "e-commerce", and "e-government" has, in general, been unsuccessful, possibly because these terms have been defined retrospectively and surrounded by a lingering vagueness. Nonetheless, Anderson and Henriksen (2008) found that the discipline "IT in the public sector"—which we now know as digital government—was successfully revived after the bursting of the dot-com bubble at the start the new millennium.

Although digital government slowly became "mainstream," scholars criticized its theoretical weaknesses and questioned its long-term sustainability (Scholl, 2006a). For example, Anderson and Henriksen (2008) found that it lacked a clear research paradigm, while Yildiz (2007) argued that it was conceptually vague. Meanwhile, Grönlund (2004) argued that as a field, digital government was immature and undertheorized. However, Anderson and Henriksen challenged Grönlund's (2004) findings, as they found that a third of articles in the field of digital government were oriented toward its conceptualization. However, the issue of the field being undertheorized continued to emerge. Bannister and Connolly, for instance, repeated this accusation in 2015. Meanwhile, Yildiz (2007) highlighted a lack of in-depth analysis of the political nature of digital government processes, as well as a failure to clearly recognize complex political and institutional environments.

Anderson and Henriksen (2008) found numerous variations and conflicts across the field of digital government. They argued that this suggests more issues divide than unite the scholars and practitioners who use the term "digital government." Yildiz (2007) further argued that there are divergent conceptions of digital government's theoretical framework, on account of three factors: (1) digital government is defined by the objective of the activity rather than technology; (2) it means different things to different groups; (3) it is connected to hype and promotional efforts, which have led to unrealistic expectations and a lack of a clear vision.

As digital government is practice-oriented, external events can impact both its focus and those who conduct research in the field. For example, the terror attacks of September 11th, 2001 caused a change in the topics that digital government scholars addressed. The focus shifted to security and privacy, as digital government infrastructure became a potential target for terrorism (Yildiz, 2007). Furthermore, research is highly dependent on funding. This is evidenced by how, after the Obama administration decreased funding for digital government research, the amount of research coming from the United States declined (Scholl, 2015a). Lastly, the different levels of digitization across countries contributes to the diversity of the research field.

To sum up, in the first two decades, digital government merged into a solidly multi-disciplinary academic field at the intersection of research streams such as public administration, information systems, computer science, and political science (Scholl, 2014). It is difficult to limit research on the digitization of the public sector to a mono-disciplinary approach, as it involves several dimensions and takes place at different levels of public administration (Bannister & Connolly, 2015). The field is criticized for being fragmented and for lacking a solid theoretical foundation. However, Scholl (2015a) argues that digital government has been criticized because researchers assume that it should adhere to the standards and rules of their respective home disciplines (Scholl, 2015b).

Digital Government from 2010 onwards

Scholl (2022) argues that digital government research focuses on the benefits of emerging technologies, such as artificial intelligence, machine learning, and blockchain (Scholl, 2020). Within government, policymakers have intensified their calls for implementing these new technologies within public institutions (Jansson & Erlingsson, 2014 as cited in Schou & Pors, 2019). Recently, Schou and Pors (2019) examined how citizens and governments are influenced by the digitization of public services, while Skaarup (2020) has explored which skills digital public services require of citizens.

Recent digital government research has also focused on public sector values (Lindgren et al., 2021; Rose et al., 2015; Rose et al., 2018; Twizeyimana & Andersson, 2019). For example, Rose et al. (2018) explore the values embedded within digital government, while Twizeyimana and Andersson (2019) examine the value that it is supposed to yield. Public organizations should focus on providing value, not maximizing benefits for individuals (Lindgren, et al., 2021); this is one way in which they differ from private organizations. Lindgren et al. (2021) find that public organizations' IT values include transparency and privacy (Lindgren, et al., 2021). Meanwhile, Twizeyimana and Andersson (2019) identify three overarching and overlapping dimensions of the public value in digital government: improved public services, improved administration, and improved social value. More broadly, they argue that the public value of digital government should be understood as the impact it has on government operations, actions, policies, and services for citizens (Twizeyimana, & Andersson, 2019).

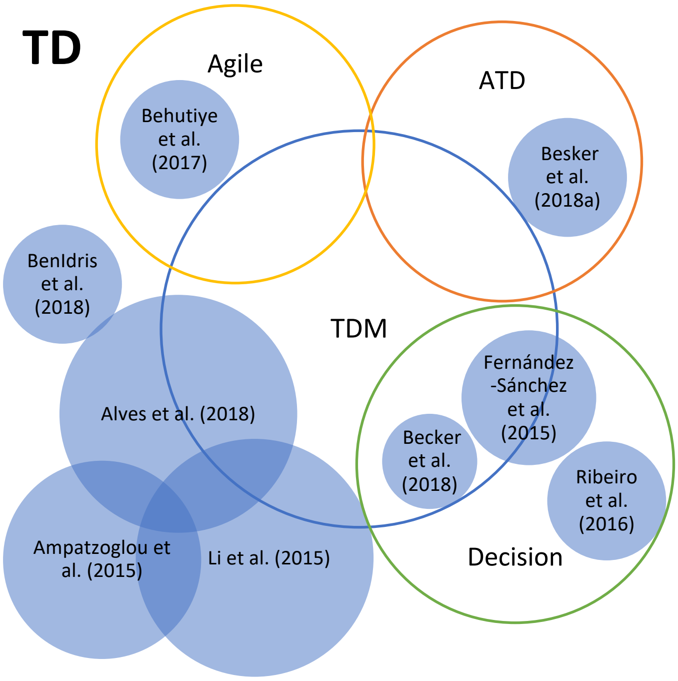
To sum up, digital government research is a relatively young field that emerged alongside the World Wide Web. The field arose out of disciplines such as information systems, computer science, and public administration, and scholars are still encouraged to follow a multi-disciplinary approach. Over time, researchers have accepted digital government as a field that encompasses a range of major issues, including new technologies, digitization and digital public services, and public sector reform. Therefore, I have chosen to position my PhD within this field and combine it with a term (TD) imported from computer science. The purpose is to create awareness, both among digital government scholars and practitioners of TD, of the benefits of digitization and the challenges that public organizations face in implementing it. I also seek to offer recommendations on how public organizations can overcome these challenges.

3.2 Previous literature reviews of TD

This section introduces TD, examines previous literature reviews of TD, describes the development of the literature review in this dissertation (Article 1), and summarizes the literature review in that article.

TD was first introduced in 1992 by Ward Cunningham (Rios et al., 2018). Since then, research interest has grown, particularly within the field of software engineering. Therefore, Fairbanks (2020) argues that TD meant something different in 1992 than it does today. Fairbanks (2020, p. 95) names the original TD for ur-technical debt and describes as "*… ur-technical debt arises when my ideas diverge from my code. That divergence is inevitable with an iterative process.*". Today Fairbanks (2020, p. 95) argues "*any code that a developer dislikes is branded as technical debt*.". Thus, TD is used in a broader sense today compared to 1992.

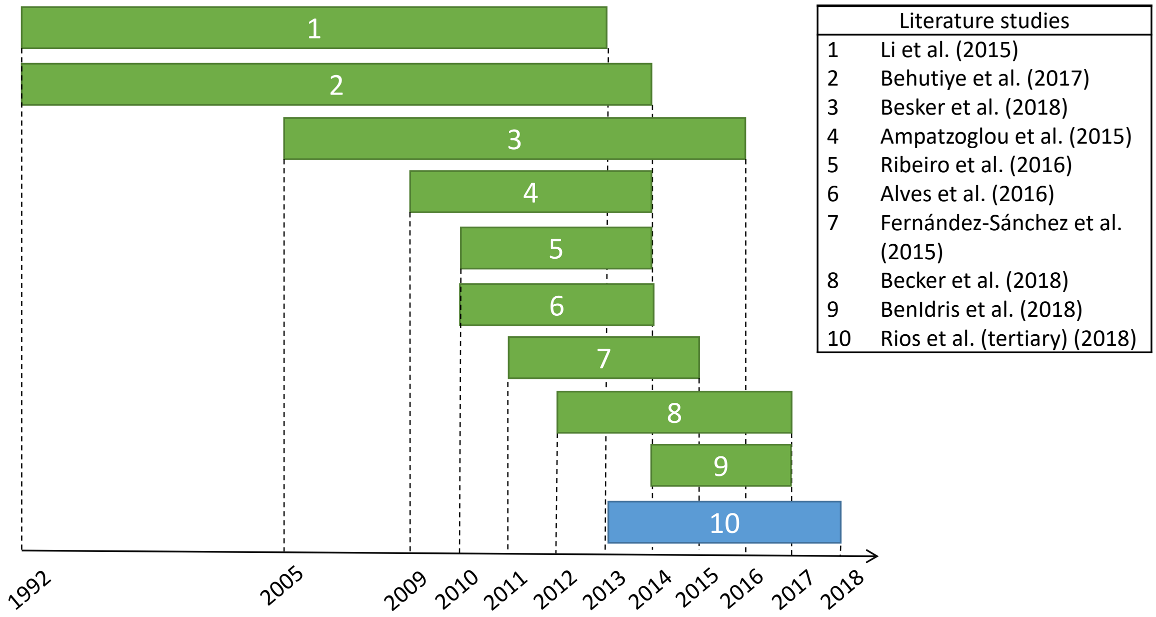
The TD metaphor uses financial vocabulary to enhance understanding for non-technical people of the technical challenges related to IT development (Guo & Seaman, 2011). Several different definitions have been suggested for the phenomenon (Krutchen et al., 2013; Rios et al., 2018). Fairbanks (2020, p. 98) claims TD has "*been watered down and that developers are* *unaware of the original and insightful ideas."* So, to gain an overview of the existing literature, I performed a systematic literature review following Webster and Watson (2002). Article 1 offers background regarding the research into TD management. Therefore, this chapter presents an introduction to the literature and a summary of the article.



Abbreviations	
TD	Technical debt
Agile	Technical debt in agile setting
ATD	Architectural technical debt
TDM	Technical debt management
Decision	Decision making supporting technical debt management

Figure 3: The areas the literature reviews cover

Because I was new to the research on TD, I began by conducting a literature review. Quickly, several things became apparent. First, the literature on TD is extensive and primarily focused on the IT systems' code. I followed Webster and Watson's (2002) three-step approach going through the literature systematically. From the initial search, I discovered several literature mapping studies (Li et al, 2015; Besker et al, 2018a; Rios et al., 2018). Then I considered conducting a tertiary review, and I mapped the areas covered by the literature mappings studies (see Figure 1). However, I discovered that Rios et al. (2018) have already conducted a relatively comprehensive tertiary study. Therefore, I decided to chart the years of the primary research that was covered by the different secondary and tertiary literature studies (Figure 2). The mapping of the previous literature studies (Figures 3 and 4) enabled me to decide on selection criteria for the literature review I conducted. It became clear that there was a lack of secondary studies focusing on TD management, particularly in recent years. The next section presents the literature review (Article 1) and its main findings



Literature studies	
1	Li et al. (2015)
2	Behutiye et al. (2017)
3	Besker et al. (2018)
4	Ampatzoglou et al. (2015)
5	Ribeiro et al. (2016)
6	Alves et al. (2016)
7	Fernández-Sánchez et al. (2015)
8	Becker et al. (2018)
9	Benldris et al. (2018)
10	Rios et al. (tertiary) (2018)

Figure 4: The time periods the secondary and tertiary literature covers

3.3 Findings of Article 1: Technical Debt Management: A systematic literature review and research agenda for digital government

The article's purpose is to introduce TD to the digital government field and map the TD management research that explores how other researchers have examined TD management. The literature review contributes to the field of TD management and proposes a research agenda for digital government scholars who wish to study TD.

The main findings in this article are:

1) There is a lack of research on the public sector. This is important as the public sector differs from open-source projects and the private sector in terms of specific requirements. For example, the private sector is guided by market signals while the public sector is guide by societal obligations and political decisions (Campbell, McDonald & Sethibe, 2010).
2) There is a tendency to reinvent the wheel or, in this case, a TD tool. The researchers present and evaluate tools and plan to apply them in a different context. They rarely apply other researchers' tools or methods; instead, they develop their own. This is problematic as it does not directly advance the field.
3) There is a lack of diversity in methodological approaches. TD is primarily studied quantitatively; only 30% of the papers apply a qualitative approach. The primary techniques the scholars used were surveys and literature studies. The implication of using surveys is that they keep the researcher at a distance from the actual social situation and processes (Blaikie & Priest, 2019). Meanwhile, literature reviews tend to overlook questions concerning specific IT "practices and techniques," even though they provide great insight into previous knowledge (Kitchenham et al., 2009, p. 12). 11 of the 49 papers used more than one technique, and they often combined a literature study with a technique to shape their research approach. Overall, therefore, TD is primarily studied using only one technique, and no scholars applied observation to the actual use of IT systems. The benefit of using several techniques is triangulation,

which can make conclusions more robust (Miles et al., 2014). Additionally, given the relative lack of observation, it is concluded that people's actual behavior concerning TD and TD management is understudied. As the selected papers focused on programming and technical stakeholders (e.g., IT professionals), it is also argued that their level of analysis was limited to individuals and IT systems and does not address the organizational issues.

4) There is a tendency for the articles to find empirical confirmations of the negative effects of TD. TD has a negative impact on software development work both in terms of morale and wasted time.
5) We see an absence of the use of theory; only a few articles import theory and approaches from other fields, e.g., portfolio finance and games. Theories are needed to explain the relationship between the concepts (Gregor, 2006).

3.4 Knowledge gaps identified

In the literature review, we identify several gaps in the academic literature concerning TD and propose an agenda to digital government scholars for how to research TD management. Furthermore, the literature review provided a direction for the research. Table 1 is an edited version of Table 4 in the article; it includes the ways in which the gaps are addressed in this dissertation.

Table 1: Gaps in the TD literature and how (and where) these gaps are addressed in this dissertation.

Gap	How the gap is addressed in this dissertation	Addressed in article
Lack of theory	Application of stakeholder theory, metaphor theory, and IT portfolio literature	2, 3, 4
Lack of methodological diversity	Observation, interviews, document analysis, and desk research	2, 3, 4
Lack of research in the context of the public sector	This dissertation researched TD management in the context of the public sector using a Danish agency.	3, 4
Limited abstraction in the level of analysis	An embedded case study allows for studying TD on different levels from the individual IT system to IT portfolio management.	3, 4
Lack of venue diversity	Through presentations and articles published in the digital government outlets. I introduced TD to digital government scholars.	1, 2, 3, 4

I addressed the lack of theory by applying three different analytical lenses to examine TD. The literature on metaphor allowed me to understand TD as a metaphor and the strengths and limitations of that metaphor (Article 2). The IT portfolio management literature (Article 3) enabled me to view TD on a higher level of abstraction than the individual IT system. For example, I found TD also exists in IT application portfolios. I used stakeholder theory to map the stakeholders and their activities and consequences and to explain why TD is created and how it is resolved (Article 4).

Previous research has largely been conducted within the discipline of software engineering. This means TD has mostly been explored from a quantitative perspective, often via surveys and code analysis tools. This research informs us about self-admitted TD, the TD detected in the code, and IT professionals' experiences of and thoughts on TD. However, quantitative approaches and techniques have limitations. For example, they cannot bridge the gap between what people say they do and their actual behavior (Blomberg and Burrell, 2012). Thus, a primarily quantitative approach cannot provide the complete picture of TD or explain its underlying mechanisms. In the selected papers, the level of analysis was limited to the individuals and the IT systems involved. I seek to broaden this via a qualitative approach that analyzes not only individual IT systems but also IT management, in addition to observing actual behavior and events. Furthermore, I employ different methods and data sources in order to achieve triangulation (Miles et al., 2014). This enriches the empirical material and strengthens the dissertation's conclusions (Miles et al., 2014).

I addressed the lack of methodological diversity by conducting observations, interviews, document analysis, and desk research. By applying several techniques, I gained access to multiple perspectives on TD, which gave insight into TD and how it is created. This nuance is difficult to obtain thorough surveys and analysis of the individual IT system's source code. I combined the gaps "lack of research in the context of the public sector" and "lack of venue diversity" by studying TD in the public sector. This gap inspired me to position the research within the digital government field.

4. Theoretical frameworks in the thesis

This chapter presents the chosen theoretical frameworks. The purpose of the following sections is to explain the theoretical framework and methodological approach I have used in my Ph.D..

4.1. The development of the research plan

In the research, I have been curious to understand the phenomenon of TD, what it entails, and how far it reaches outside of a public organization's IT department. In the first couple of months, I made two important realizations, which impacted my plans. First, while reading papers for the literature review, I found numerous studies in software engineering that analyzed source code. Further, several scholars had also conducted interviews (Besker et al., 2018c; Ghanbari et al., 2017). Second, I ran a source code analysis using SonarQube and found that the TD items primarily were related to the libraries they imported. Therefore, I decided to adjust the focus of my research to understand how IT professionals' perceptions of TD contribute to TD management.

I chose a theoretical framework that enabled me to explore the various stakeholders and their interaction with TD. Stakeholder theory involved stakeholder mapping and an exploration of how stakeholders affect TD. IT portfolio management forced me to explore TD from a portfolio management perspective and become aware of the different elements of IT portfolios that relate to TD. Lastly, scholarly literature on metaphor facilitated an understanding of TD's essence.

Figure 5 illustrates how the different elements correspond to each other. Engaged scholarship is the form of research I conducted. Within the articles, I have applied different analytical lenses: stakeholder theory (Article 4), IT portfolio management theory (Article 3), and metaphor theory (Article 2). The underlying methodological approach is embedded case study and, for the metaphor article, also document analysis, which is elaborated in section 7.1

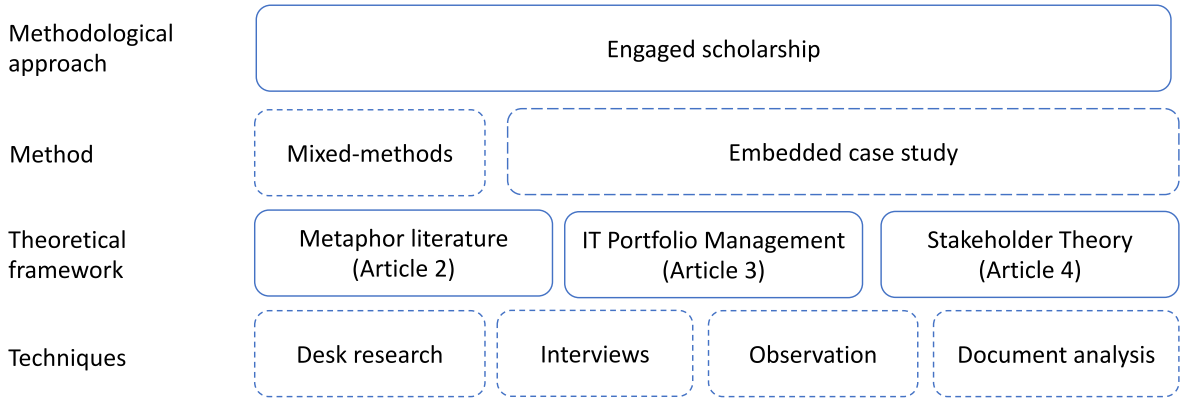


Figure 5: The relation between engaged scholarship, and the theoretical frameworks in this dissertation.

4.1.1 Analytical lenses. I applied three analytical lenses, metaphor theory (Article 2), IT portfolio management theory (Article 3), and stakeholder theory (Article 4). I initially chose to explore 1) the essence of TD by exploring it as a metaphor, 2) the source code, and 3) how other people interacted and perceived it. Then I took a step further and examined the management of TD and the dynamics that influence TD. First, I used metaphor theory (Article 2) to inform me on the phenomenon from a metaphorical perspective. Furthermore, I applied IT portfolio management theory (Article 3) to connect TD with other elements and activities in IT development. Lastly, I applied stakeholder theory to gain the perspectives of various employees interacting with the phenomenon and influencing the phenomenon. Stakeholder theory (Article 4) contributed to identifying different stakeholders in relation to TD and how the stakeholders perceive and interact with TD. This lens enabled an expansion of TD to a higher level of abstraction to include the management. Because of the stakeholders' interactions and frustrations with TD, I realized TD is not limited to coding; the phenomenon or problem takes place on a higher level of abstraction (as well as on the coding level).

4.1.2 Research philosophy of engaged scholarship. Van de Ven (2007) suggested engaged scholarship as a participative form of research for studying complex social problems. Engaged scholarship is a method for obtaining the advice and perspectives of key stakeholders in order to understand a complicated problem or phenomenon (Van de Ven, 2007). Engaged scholarship focuses on involving practitioners' perspectives on a specific problem or phenomenon. Furthermore, engaged scholarship does not rely on different epistemologies (Mathiassen, 2017), but ontologically it adopts critical realism and realistic pragmatism (Van de Ven, 2007). Van de Ven (2007, p. 70) encourages researchers "*to initiate a process of reflexivity in choosing a philosophy of science that suits your scholarly practice*." Engaged scholarship employs multiple perspectives to investigate a complex reality, I will return to engaged scholarship and how I have applied it in section 5.1.

4.2 Metaphor theory as an analytical lens

In Article 2 we apply metaphor theory as an analytical lens to explore the metaphors related to TD and Open Government Data. The purpose of applying metaphor theory is to enhance our understanding of the phenomena we research. Within Open Government Data studies the term "ecosystem" is often used (Dawes, Vidiasova & Parkhimovich, 2016), whereas the concept TD was created based on the "debt" metaphor (Fairbanks, 2020). Therefore, we synthesize the metaphor literature to construct a framework for a metaphor's development; Communities' Applications Of Metaphors (CAOM) framework. The framework brings nuance to metaphors and their application possibilities.

Morgan (1998) defines metaphors as implicit or explicit assertions that something is like something else. Metaphors bring a specific structure to thoughts (Lakoff & Johnson, 2008) even though they are false in the literal sense (Davies, 1982). "*The essence of metaphor is understanding and experiencing one kind of thing in terms of another*" (Lakoff & Johnson, 2008, p. 5). Metaphors can help enrich vocabularies (Black, 1955, Lakoff & Johnson, 2008) and be viewed "*as essential to human understanding and as a mechanism for creating new meaning and new realities in our lives*" (Lakoff & Johnson, 2008, p. 196).

If receivers lack knowledge, they can misunderstand metaphors (Glucksberg, 1989) or fail to grasp them (Ortony, 1975). Using metaphors in ambiguous, knowledge-intensive situations with unclear intentions and purposes can hinder learning and create chaos rather than order (Hekkala et al., 2018). People tend to search for information that aligns with their metaphorical framing, but they seldom recognize that metaphors influence their decisions and behaviors (Thibodeau & Boroditsky, 2011). The receiver's understanding of a metaphor depends on their experiences with the origin phenomenon to which the metaphor refers (Lakoff & Johnson, 2008). At the same time, a metaphor implies a way of thinking as it brings a specific structure to target thoughts based on previous thoughts and can be a way of understanding the world generally (Lakoff & Johnson, 2008; Morgan, 1998).

The receiver highlights similarities between an origin phenomenon and a target phenomenon with a metaphor (Lakoff & Johnson, 2008). Ogden and Richards (1923) empathize that metaphors should only possess features shared between the two phenomena; irrelevant or accidental features need to be removed. The receiver must sort out the differences and focus on the similarities (Glucksberg, 1989). Mars, Bronstein and Lusch (2012) warn that similarities between the target phenomenon and the origin phenomenon might only be superficial. The application of metaphors in a community can be understood from a meme perspective (Dawkins, 2016). A meme is a unit of cultural transmission that can jump from sender to receiver through a process of imitation. A meme can be a metaphor which we assume in article 2. We use the concepts from meme theory to create the CAOM framework. This framework captures the application of metaphors within communities.

The article contributes a new framework that can map the underlying mechanisms of a metaphor (metaphor usage, expression, and metaphor complex) to illustrate the application possibilities and pitfalls. Furthermore, applying metaphor literature we can suggest the best applications of a metaphor while considering the context it is used in. When the metaphor does not cover the phenomenon fully, it can be supplemented by other metaphors. Thus, the limitations of one metaphor can be resolved by applying supporting metaphors.

4.3. IT portfolio management

In Article 3, "IT Portfolio management as a framework for managing Technical Debt," we apply IT portfolio management (ITPM) concepts as a lens for exploring TD management. The purpose of choosing ITPM as a lens is to raise the abstraction level from IT system to IT portfolio management. ITPM is fitting because the purpose of ITPM is to apply a holistic perspective on these sub-portfolios as one interconnected portfolio (Gliedman, 2004; Peppard, 2003; Kumar, Ajjan & Nu, 2008).

Simon, Fischbach, and Schoder (2010, p. 37) define a "portfolio" as "*a collection of items grouped together to facilitate their efficient and effective management*." Typically, a portfolio comprises a group of business investments with something in common, e.g., they contribute to the same overall organizational goal(s). A portfolio is a dynamic entity that must be continuously managed, monitored, and maintained as a set of interrelated assets/activities that efficiently address business needs (Kellerman & Löfgren, 2012).

The main goal of managing an IT portfolio is to ensure that it supports the activities and goals of the business (Lemmetti 2016; McKeen & Smith 2010). Thus, an essential activity across the portfolio is to ensure that it is well-aligned with business needs and goals.

ITPM includes managing applications, infrastructure, projects, and people along with their mutual dependencies (Kumar et al. 2008; Simon, Fischbach & Schoder, 2010). Another key activity in managing an IT portfolio is to decide what parts of the portfolio to operate, maintain, and develop in-house. Additionally, managers must decide which tasks and applications the organization needs to buy from a vendor and which to develop internally (Scholl, 2006b). These decisions are not only a matter of economics but are closely related to considerations of what resources and skills are, can be, and should be available in-house (Roy & Auberts, 2002).

In Article 3, we synthesize the IT portfolio management literature into a framework which we then apply to the embedded case study as further elaborated in section 6.2. By integrating TD into the study of IT portfolios, we can explore TD on a higher level of abstraction. This link is significant as it addresses the abstraction gap identified in the literature study (Article 1). Furthermore, applying IT portfolio management as a lens, we found TD occurred in other areas of the IT portfolio beyond the individual IT systems.

4.4. Stakeholder theory

This section introduces stakeholder theory as it is applied in Article 4, "Stakeholder influence on TD management in the public sector: An embedded case study." Here, I sought to contribute to the understanding of TD by exploring stakeholders' influence on TD in a public-sector setting. I wanted to identify stakeholders and analyze their actions; therefore, I applied stakeholder theory.

Freeman (1984) developed stakeholder theory to increase revenue in a private organization. He argued that organizational profits are higher when stakeholders are satisfied. A stakeholder is defined as *"any group or individual who can affect or is affected by the achievement of the organization's objective*" (Freeman, 1984, p. 25). Stakeholder theory entails identifying 1) the stakeholders affected by or affecting the organizational objective, 2) the stakeholders' actions, and 3) how they affect or are affected by the organization's objective. The theory focuses on the firm's internal stakeholders (e.g., employees and managers) and external stakeholders (e.g., customers and suppliers). Freeman argued that "*you must take your stakeholders into account in a systematic fashion*" (1984, p. 48). This entails identifying them, their actions, and their influence to manage effectively. Stakeholder theory is useful as a lens when researching TD because stakeholder theory addresses the stakeholders' different interests and goals. These different interests and goals can be difficult to navigate when developing and maintaining IT systems. Competing priorities can lead to short-cuts, abandoned development, and similar problems that may result in TD.

Scholl (2001) introduced stakeholder theory to the digital government field and identified how this theory could benefit "*public-sector managerial decision-making*" (Scholl, 2001, p. 745). Flak and Rose (2005) further explored the effects of employing stakeholder theory in digital government studies. They highlighted an issue that results from transferring the theory from a private- to a public-sector context. Increasing profit is an important private sector value but does not apply in the public sector. Rose et al. (2018) contributed to the research on stakeholder theory in digital government by identifying four values relevant to the public sector: professionalism, efficiency, service, and engagement.

Stakeholder theory is useful for identifying stakeholders and their actions, and together with Rose et al. (2018), it identifies values for the public sector that serve as one of my analytical lenses. Although stakeholder theory is widely used within digital government (Balta et al., 2015; Rose et al., 2018; Sæbø et al., 2011), previous research has not applied it to this specific topic. I conducted interviews to better understand the perspectives of the different stakeholders working with IT systems. Furthermore, stakeholder theory explains the objectives of public organizations; this explanation impacted the findings presented in Article 4 and section 7.4.

5. Method

The overall aim of this dissertation is to examine TD management in a public organization. The dissertation is guided by the research question: *How can the concept of technical debt be operationalized to improve the management of IT systems in the public sector?* I explore this question using a case study and an engaged scholarship approach. Previous TD studies has primarily been conducted in a quantitative manner; I address this gap by applying a qualitative approach.

The goal of the thesis is to provide results that are relevant to practitioners by providing useful insights, tools, and strategies for TD management. This chapter is divided into three parts: engaged scholarship, embedded case study, and applied data collection techniques. It presents the methods and techniques applied in the thesis.

5.1 Engaged scholarship

I have adopted the engaged scholarship approach, which is a participatory form of research (Mathiassen, 2017). The core aim of engaged scholarship is to bridge the gap between practice and academia (Van de Ven, 2007). Van de Ven (2007) proposes engaged scholarship as a way for scholarship to be made *with* practice rather than *for* practice. Therefore, he urges scholars to consult relevant experts and practitioners regarding research design, theory building, problem formulation, and problem-solving.

As a former practitioner in the public sector, I have chosen this form of research because I aim to contribute to both practice and academia. During the literature review, I discovered that there is a lack of research on specific TD management tools and strategies addressed to management. I noticed the absence of these tools and strategies as a practitioner. Furthermore, I am a part of the Research Centre for Government IT, which was established because the Danish government found there was a lack of knowledge about and capability for governing IT in the public sector (Danish Ministry of Finance, 2017a; Danish Ministry of Finance, 2017b).

Engaged scholarship influenced my research design. I decided to examine a real-world problem by applying an embedded case-study approach. Furthermore, I learned the different perspectives and understandings of the key stakeholders on TD and TD management. Thus, engaged scholarship influenced the method and techniques I used to gather empirical data. As a former practitioner, I wanted to understand perspectives I was less familiar with. I interviewed and observed employees with different jobs.

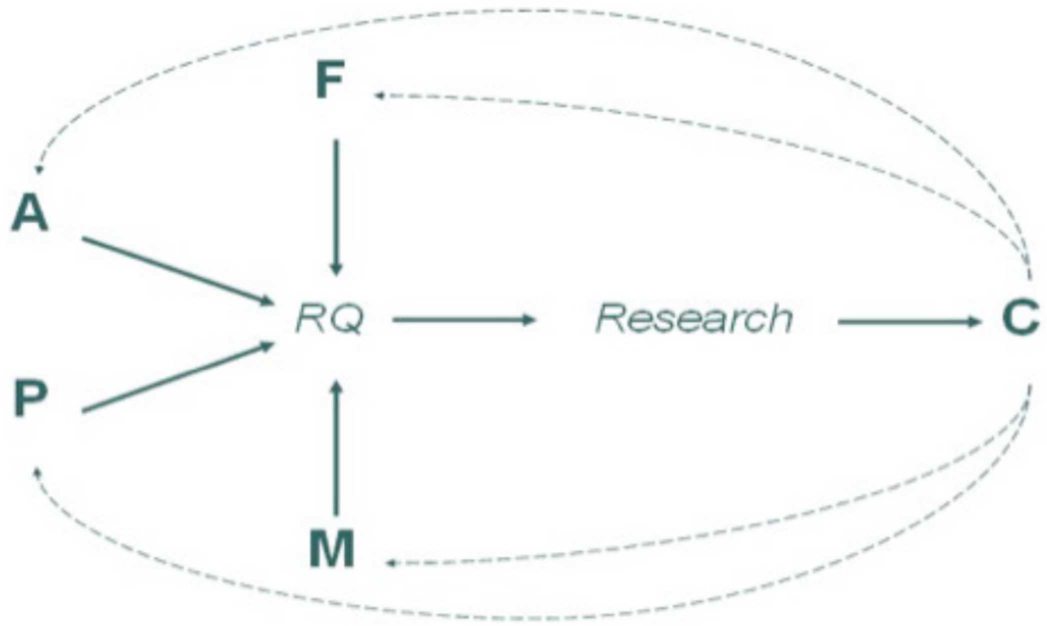


Figure 6: Lars Mathiassen's (2017) generic structure of Engaged Scholarship Study

Mathiassen (2017) lays out a design or approach for practitioner-scholars to use to improve research. Mathiassen argues that there are many variants of engaged scholarship; however, they follow the same basic structure (Figure 6). The research question is informed by the framework (F), the area of concern (A), the problem (P), and the method (M). The research question then guides the research which leads to contributions (C). These contributions can be to the F, A, P, or M. Mathiassen differentiates between the research design and the publication design. However, it is an iterative process oscillating between the two designs, until the research design is stable and consistent. I will return to this topic in Chapter 9 when I present the thesis' contributions.

5.2 Embedded case study

I aimed to understand TD from an ontological standpoint by investigating others' perceptions and interactions with the phenomenon. I employed a qualitative and explorative approach by conducting a case study. The case study research approach is recommended when the researcher wants to explain "how" or "why" a social phenomenon works. Additionally, an embedded case study is especially suitable when the researcher seeks to explain how a social phenomenon works (Yin, 2018). I sought to explore how the concept of technical debt can be operationalized to improve the management of IT systems in the public sector.

5.2.1. Embedded case study intro. Yin (2018, p. 51) says of an embedded case study that it "*may involve units of analysis at more than one level. This occurs when, within a single-case (the first level), attention is also given to a subunit or subunits (second level).*" The embedded case study entails exploring the phenomenon through the sub-units (Yin, 2018). Therefore, I chose an embedded case study, because it provides context as it enables exploration at different layers. It helped me to explore both the individual IT systems and the IT portfolio.

Yin recommends a single-case design when the case is longitudinal and represents an unusual case. I consider the case longitudinal as data was collected over five months. Furthermore, I have selected an unusual case, which is defined as a case "*deviating from theoretical norms or even everyday occurrences*" (Yin, 2018, p. 50). The case is unusual because the organization is considered one of the most mature public sector IT organizations in Denmark. Thereby, researching this case allowed me to observe good IT management and how TD management is performed in a public organization. I conducted an embedded case study using a single case with four subunits (Table 2) as they allow for "*extensive analysis*" and enhance "*the insights into the single case*" (Yin, 2018, p. 54). I chose an agency that develops its own IT systems and has developed IT systems for other authorities. The agency has a standardized setup and processes for maintaining its IT systems. They work in partnership with software development companies to maintain and develop IT. The software developers are placed on-site to ensure close collaboration.

Throughout the dissertation, I differentiate between four subunits and four IT systems. The subunits are the conceptual terms used in the method, and the IT systems are operational terms used when I am analyzing and referring to the actual IT systems. I conducted an embedded case study with four subunits to explore TD on both an individual IT system level and a holistic IT development level (Yin, 2018). Thereby, I researched TD management in the case organization by investigating four IT systems (Article 4).

Table 2: Overview of the examined IT systems

Name of IT systems	Type of IT systems	Short description of the IT systems
REPORT	Front-end	Political IT system which is developed on behalf of a different authority
SERVICE	Front-end	A service provided to the citizens in exchange for a fee
TEXT	Back-end	A content-management system that provides the other IT systems with text
CASE	Back-end	A case management system with a front-end to caseworkers and a database for other IT systems to draw upon

5.2.2 Data collection techniques. To ensure that the research was of high quality, I conducted the case study using multiple sources of evidence (Yin, 2018). I chose the three techniques to gain insights into IT practices: (1) in-situ and semi-structured interviews, (2) participant observation, and (3) document analysis and source code analysis (Table 3). The code and documents are physical artifacts the practitioners use, whereas observation and interviews gave insight into their routines, practices, and thoughts. These techniques enabled me to understand the phenomenon through the experience of people interacting with the phenomenon.

The data collection took place during 2019. In May 2019, I began the data collection as an observer in the organization's quarterly IT planning workshop. This event entailed planning the IT development for the next several months (known as PI planning in agile settings). Most of the observations took place from August 2019 to December 2019. I was on-site three days a week conducting interviews and making observations.

Table 3. Overview of the applied techniques

Technique	Count	Amount of data
Interview w/ application manager	4	210 minutes
Interview w/ architect	4	258 minutes
Interview w/ developer	4	223 minutes
Interview w/ law graduate	1	53 minutes
Interview w/ user representative	2	86 minutes
Interview (in situ) w/ user representative	1	35 minutes
Interview (in situ) w/ user	1	73 minutes
Total number of interviews	17	937 minutes
Participant observations		
On-site 3 days/week	7 months	13,555 words in field notes
2-day workshops	3 times	3,193 words in field notes
Document analysis		
Source code	4 IT systems	
Documents	30 documents	

Next, I conducted 15 semi-structured interviews with Agency X employees and software developers working on the IT systems (Kvale, 2008). The interviews consisted of open-ended questions framed around the employee's daily work routines and their collaboration with other employees maintaining and managing the IT system. I applied purposeful sampling and interviewed only the employees that were part of the daily management of the IT systems (Kvale, 2008). From the interviews, I gained insights into the employees' perception of and reflection on the maintenance processes and their interactions with their peers.

Additionally, I conducted two in situ interviews with a user and a user representative of the IT systems (Bodker, Kensing & Simonsen, 2004; Blomberg & Burrell, 2012). During the in situ interviews the interviewees performed their tasks while I asked about their actions and reflections. This method gave me insight into the challenges that IT systems pose to their users and the users' reactions to these challenges.

The participant observation was carried out by being present at the case organization (Brannan & Oultram, 2012). Conducting observations allows the researcher to address the say/do problem, where people may say one thing during interviews and do something else in practice (Bloomberg & Burrell, 2012). Additionally, I attended three of the quarterly PI workshops, in which the following months of IT development were presented by the projects and maintenance teams. Furthermore, IT tasks were prioritized with assistance from the business unit. The dependencies on other IT systems and projects were coordinated on an analog board with thread and pins. Furthermore, concerns were raised by the employees and, if possible, attended to immediately by the managers. I was introduced to the employees at the beginning of the workshop and my role was made clear. I found people willingly included me in their discussions and started conversations with me during the workshop and their daily activities. Attending the workshops provided rich insight into how IT portfolio management took place (Blomberg & Burrel, 2012).

I had been acquainted with Agency X before the data collection, therefore I was able to blend in and take on the role of a participant as an observer (Brannan & Oultram, 2012). This relation also meant employees whom I did not interview initiated casual conversations on the research topic. They shared their experiences, thoughts, and concerns. This led to valuable insights into how other authorities and the parliament influenced IT development.

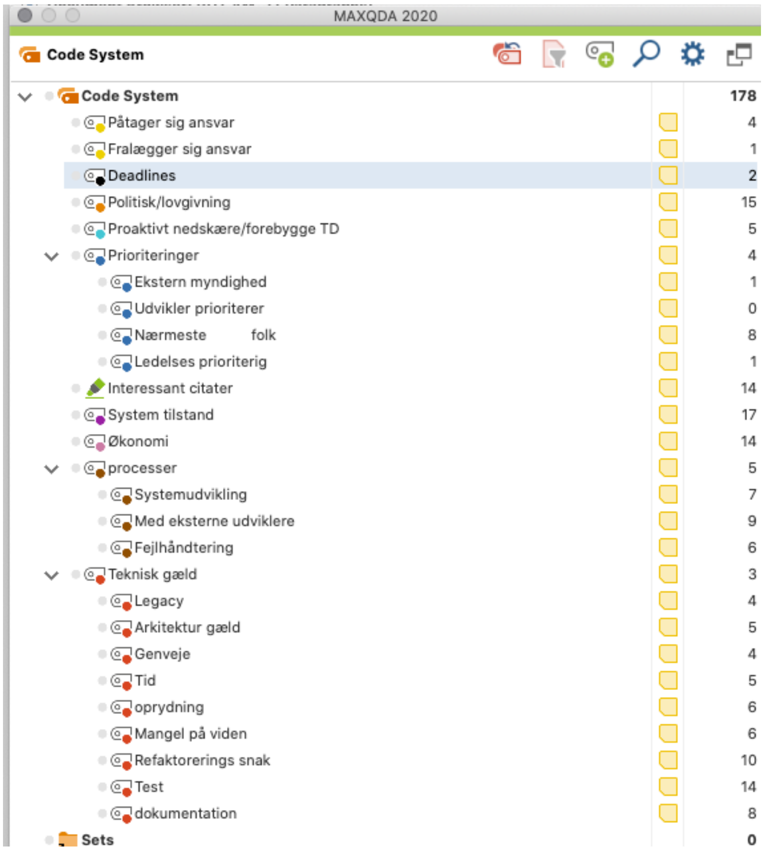
I collected documentation regarding the IT systems including the processes and the task backlogs. I also attained access to the source code and Agency X's code evaluation tool. These sources have not been explicitly used in my data collection simply because when I began the research I did not know where the most interesting findings would be. My original idea was to map out the development of the system and see if I could identify a pattern in the IT tasks. Furthermore, I believed there was a story in how TD was perceived in the organization and how the IT systems were evaluated by the organization's tool and by SonarQube.

Although the original idea was interesting, existing TD literature had begun pointing out how measurement tools are used in practice (Vassallo, Palomba, Bacchelli & Gall, 2018) and how the tools might not be as useful in decreasing defects (Pfeiffer, 2021). Furthermore, software developers do not agree on what constitutes TD (except for architecture) saying, "*tools do not capture the key areas of accumulating problems in technical debt*" (Ernst et al., 2015, p. 57). Additionally, Sjøberg et al. (2012) demonstrate that code smells (an indicator of problems within the design of software) have a limited impact on increased maintenance effort. The knowledge gap here appears to be addressed, so I abandoned this direction to pursue findings in the other empirical materials.

5.2.4. Data analysis. The observations were captured as field notes (Blomberg & Burrell, 2012) through continuous notetaking throughout the workshops and the daily IT operations. The interviews were transcribed and, together with the field notes and documents, imported to MAXQDA 2020 (VERBI Software, 2019), where I coded them.

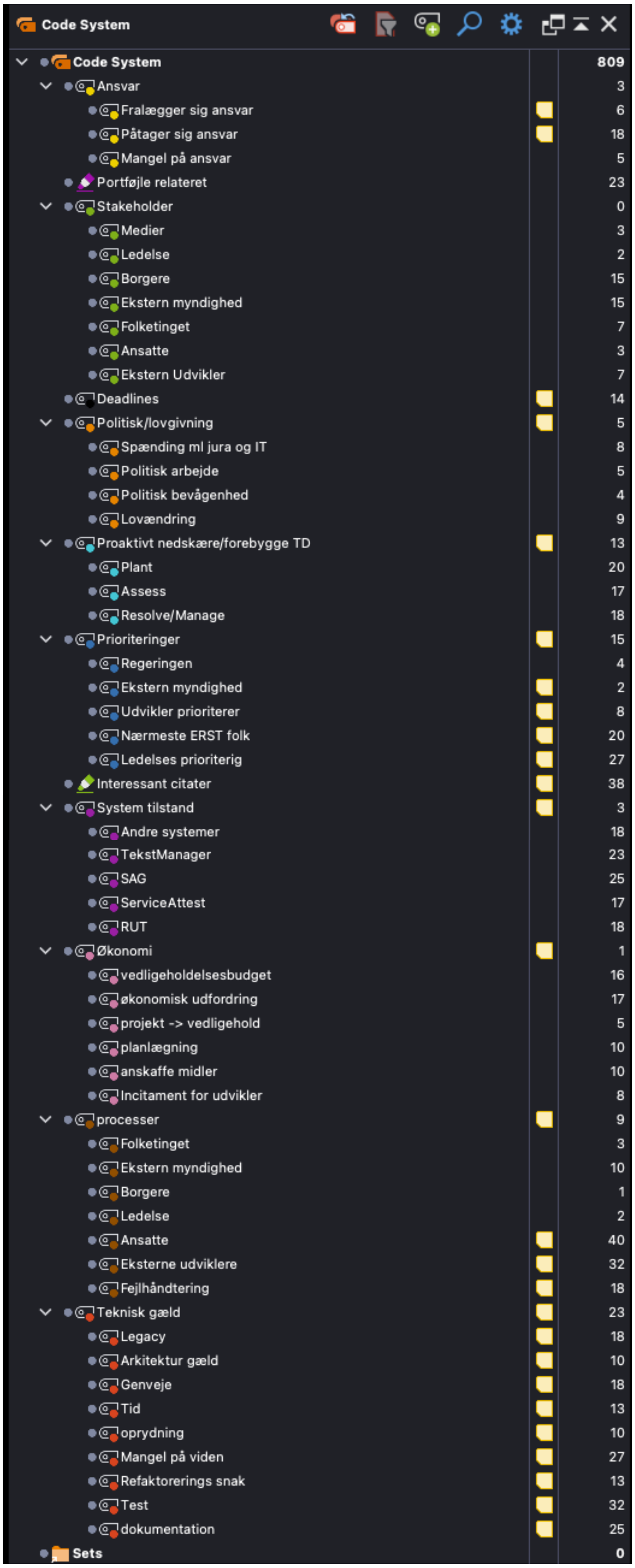
I combined the open coding with a more theoretically founded analysis. I applied IT portfolio management and stakeholder theory to code the empirical material. However, the metaphor literature was applied solely to the interviews and the desk research presented in Article 2. We went over the words that entailed a metaphorical meaning and grouped these words into leading metaphors.

I began by conducting open coding, going through all the material directly linked to one IT system. The initial coding is displayed in Figure 7; at this point, I had created the codes I thought I needed and had gone through them and created sub-codes, for example to "processer" (processes). Some sub-codes have been deleted or removed, for example, "genveje" (short-cuts) had its own sub-codes. The colors were aligned so the sub-codes match the parent codes.



```
MAXQDA 2020
Code System
Code System                              178
    Påtager sig ansvar                     4
    Fralægger sig ansvar                   1
    Deadlines                              2
    Politisk/lovgivning                   15
    Proaktivt nedskære/forebygge TD        5
    Prioriteringer                         4
        Ekstern myndighed                  1
        Udvikler prioriterer               0
        Nærmeste      folk                 8
        Ledelses prioriterig               1
    Interessant citater                   14
    System tilstand                       17
    Økonomi                               14
    processer                              5
        Systemudvikling                    7
        Med eksterne udviklere             9
        Fejlhåndtering                     6
    Teknisk gæld                           3
        Legacy                             4
        Arkitektur gæld                    5
        Genveje                            4
        Tid                                5
        oprydning                          6
        Mangel på viden                    6
        Refaktorerings snak               10
        Test                              14
        dokumentation                      8
Sets                                       0
```

Figure 7: Overview of codes August 2020



```
Code System                                   809
  Ansvar                                        3
    Fralægger sig ansvar                        6
    Påtager sig ansvar                         18
    Mangel på ansvar                            5
  Portføjle relateret                          23
  Stakeholder                                   0
    Medier                                      3
    Ledelse                                     2
    Borgere                                    15
    Ekstern myndighed                          15
    Folketinget                                 7
    Ansatte                                     3
    Ekstern Udvikler                            7
  Deadlines                                    14
  Politisk/lovgivning                           5
    Spænding ml jura og IT                      8
    Politisk arbejde                            5
    Politisk bevågenhed                         4
    Lovændring                                  9
  Proaktivt nedskære/forebygge TD              13
    Plant                                      20
    Assess                                     17
    Resolve/Manage                             18
  Prioriteringer                               15
    Regeringen                                  4
    Ekstern myndighed                           2
    Udvikler prioriterer                        8
    Nærmeste ERST folk                         20
    Ledelses prioriterig                       27
  Interessant citater                          38
  System tilstand                               3
    Andre systemer                             18
    TekstManager                               23
    SAG                                        25
    ServiceAttest                              17
    RUT                                        18
  Økonomi                                       1
    vedligeholdelsesbudget                     16
    økonomisk udfordring                       17
    projekt -> vedligehold                      5
    planlægning                                10
    anskaffe midler                            10
    Incitament for udvikler                     8
  processer                                     9
    Folketinget                                 3
    Ekstern myndighed                          10
    Borgere                                     1
    Ledelse                                     2
    Ansatte                                    40
    Eksterne udviklere                         32
    Fejlhåndtering                             18
  Teknisk gæld                                 23
    Legacy                                     18
    Arkitektur gæld                            10
    Genveje                                    18
    Tid                                        13
    oprydning                                  10
    Mangel på viden                            27
    Refaktorerings snak                        13
    Test                                       32
    dokumentation                              25
Sets                                            0
```

Figure. 8. Final overview of codes September 2021

Using this coding scheme as a template, I began coding the rest of the material, while I added codes as they appeared in the empirical data, for example, "økonomi" (economy) gained sub-codes (Figure 8). While coding the material I realized the impact of external stakeholders, and so I decided to apply stakeholder theory as an analytical lens. Therefore, I added stakeholders as a code and recoded the material to ensure I captured the instances when the different stakeholders were mentioned. Furthermore, I explored the code "ansvar" (responsibility), and when I examined these snippets more closely a pattern emerged. I realized that TD could be caused by someone other than the developer of the IT system. IT portfolio management, therefore, presented itself as an interesting analytical framework to capture this behavior.

6. A brief note on the articles' development

This chapter presents the development of Articles 2, 3, and 4. (The development of Article 1 is presented in Chapter 3).

6.1 Article 2. Exploration of metaphors as a way to understand socio-technical phenomena: An emergent analytical framework

TD is a concept derived from the debt metaphor. The software engineer community has mapped the different debt terms (e.g., interest, principal) onto TD (Curtis, Sappidi & Szynkarsk, 2012). I chose to dive into the concept of metaphors and what it entails when we apply it—the benefits and the challenges. I wanted to improve my understanding of TD and the differences between the phenomenon and the metaphor as well as to uncover the strength and limitations of the metaphor. I found fellow researchers at Linköping University who were likewise interested in exploring metaphors and their relation to socio-technical phenomena. We established a collaboration and aimed to explore how researchers and practitioners apply metaphors to socio-technical phenomena within their respective communities.

The work on this article enlightened me on why TD is often perceived negatively and as something which needs to be paid off as quickly as possible. This perception is due to people's experience with the original object "debt"; this perception and association are transferred to the TD phenomenon. Of course, an investor could have a positive association with debt and view it as a possibility to invest. Moreover, there is tension in the metaphor because not all of a financial debt's attributes can be transferred to TD. For example, the opposite of debt is savings; this does not transfer. Furthermore, the idea of debt being completely avoided or paid down is not realistic within IT development. Therefore, I focus on unmanaged TD because it affects IT development and maintenance as well as the stakeholders.

6.2 Article 3. IT Portfolio management as a framework for managing Technical Debt: Theoretical framework applied on a case study.

Articles 3 and 4 rely on the empirical material gathered at the Danish agency (as described in the method chapter). The idea for Article 3 occurred when I worked on Article 4 with my co-supervisor Christian Østergaard Madsen. He pointed out I was using several concepts (i.e., IT portfolio, IT systems, IT projects, and TD), but it was difficult to follow all the concepts and their relations. I searched for an illustration that captured the different concepts to address this challenge, but I came up empty-handed. Therefore, I decided to develop an illustration (Figure 9), which captured the relation between these concepts. The illustration was grounded in research. I presented Figure 9 in an article at the SWEG conference in 2021. However, it was pointed out the illustration needed an article of its own. Therefore, I separated the figure from Article 4 and partnered with Søren Skaarup to write an article on TD and IT portfolio management. That article become Article 3 of this dissertation.

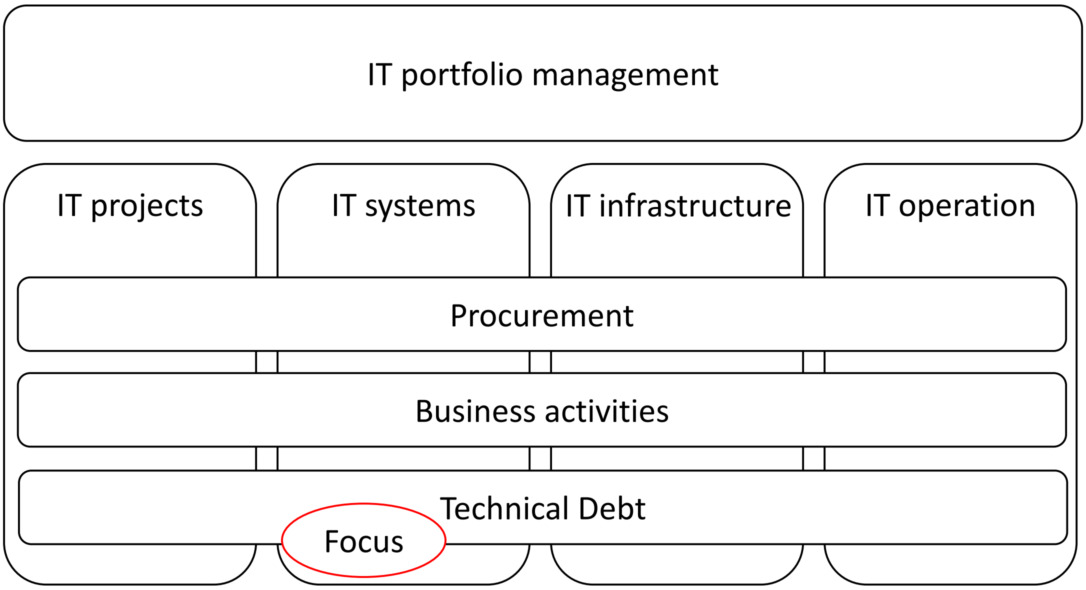


Figure 9: Early version of the framework

6.3 Article 4. Stakeholder influence on technical debt management in the public sector: An embedded case study

After submitting Article 3, I resumed work on Article 4, which focuses on IT systems and the stakeholders affecting the TD in IT systems. This article seeks to understand how stakeholders influence TD management. Here the focus is on the individual IT system. I had coded the empirical material for stakeholders. During the analysis, it became apparent that I had to limit the number of examined IT systems if I wanted to be as rigorous as possible. I decided to make a theoretical sampling for this article; however, the analysis of the remaining two IT systems will be presented in section 7.4.

7. Findings

This chapter summarizes the articles included in this dissertation and specifies the findings of Articles 2, 3, and 4. The results from the literature review (Article 1) have already been presented in Chapter 3. As Article 4 presents an examination of only two of the four IT systems in Agency X, I here present an analysis of the remaining two IT systems at the end of this chapter to supplement these results. At this point I expect the reader to have read the articles in full.

7.1 Article 2. "Development of metaphors used to understand socio-technical phenomena: An emergent framework"

This article sets out to explore metaphor development by practitioners and researchers using two metaphors as examples: ecosystem and debt. Ecosystem is explored within the field of open government data (OGD), and TD is explored in the fields of IT development and maintenance. We find metaphors are widely used to comprehend and analyze socio-technical phenomena within information systems (Hekkala, Stein, & Rossi, 2018; Oats & Fitzgerald, 2007). We identify knowledge gaps in the use of metaphors within information systems and in the understanding of metaphors as components of knowledge building in communities of researchers and practitioners. Researchers and practitioners have used metaphors to understand intangible socio-technical phenomena and prescribe a structure to them. Their use of metaphors might lead to different experiences with the metaphor and cause the metaphor to develop differently. For example, the researchers and practitioners might use TD differently because they are dealing with slightly different challenges.

We synthesize the academic literature on metaphor and create a framework, which we call the development of metaphor framework (CAOM framework), for analyzing metaphor. The framework serves as an analytical lens for both our quantitative and qualitative research. We then discuss the framework and to what extent we find it useful.

We apply a mixed-method approach to explore metaphor development within two fields. We examine both academic literature and practical material generated in the period 2015-2020. For the researchers' community, we explore six journals and conference proceedings within software engineering and digital government. While for the practitioners' community, we use online material such as blogs, reports, and news articles. OGD and TD differ in the level of analysis; TD focuses on a local phenomenon related to a system and is limited to the IT department in the organization. While OGD focuses on actors and interactions, and the EU provides OGD guidelines. Due to the differences between TD and OGD, however, we chose to include interviews about TD.

For the quantitative analysis, we conduct a word frequency analysis to gain an overview of the collected material, and we then apply the CAOM framework. For the qualitative analysis, we examine material manually to capture the nuances; the CAOM framework provides us with the elements to analyze (i.e., the metaphors' usage, expressions, and metaphor complexes). We focus the analysis on the respective metaphors: debt and ecosystems. We also include their relation to other metaphors.

We find the CAOM framework to be beneficial in understanding a community's application of a metaphor. The framework combines the perspective from meme theory with metaphor literature. When we apply the framework, we find the development of metaphors is not as linear as expected. Furthermore, the metaphors do not appear to exist alone; instead, other metaphors supplement the understanding of and interaction with a socio-technical phenomenon.

We find that metaphors are part of ever-evolving processes following similar patterns even between communities and socio-technical phenomena. Additionally, the application of metaphors for conceptual development in communities is influenced by community type (e.g., researcher or practitioner; open or semi-private) and metaphors' role (e.g., core or peripheral). The communities' socio-technical phenomena influence their selection of metaphors but have a negligible influence on their application of metaphors. The community structure influences how senders communicate to receivers but also acts as an entry barrier for newcomers to the community. It impacts the quantity and quality of empirical material.

The article provides several contributions. First, it offers a framework that can be applied as an analytical lens. It captures the underlying mechanisms of metaphor usage, expression, and complex. Second, it provides empirical insights into how researchers and practitioners employ metaphors when thinking about and discussing TD and OGD.

7.2 Article 3. "IT portfolio management as a framework for managing technical debt: Theoretical framework applied on a case study".

Although previous research has discussed the need to view TD in a larger context than the individual IT system, none to our knowledge has described how to view TD in a larger context or how TD relates to IT portfolio. The goal of this article is to make it easier for management to incorporate TD in their IT planning. In the case study conducted at Agency X, we found TD to be affected by management decisions and other applications than the application where the TD resides.

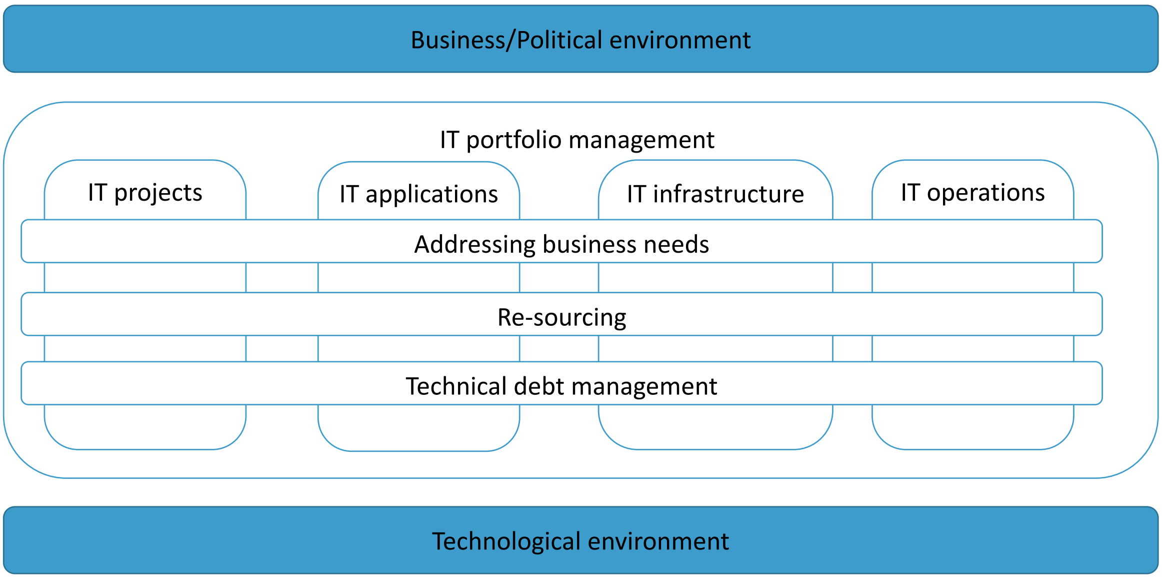

Business/Political environment
IT portfolio management
IT projects
IT applications
IT infrastructure
IT operations
Addressing business needs
Re-sourcing
Technical debt management
Technological environment

Figure 10: A framework that encompasses technical debt management in IT portfolio management

The article explores how TD can be created, sustained, managed, and resolved from a portfolio perspective. We develop a framework for IT portfolio management that integrates TD management (Figure 10). We apply the framework as an analytical lens on the empirical material gathered.

The article has four main findings: 1) TD can reside in an application portfolio. For example, Agency X had several content management systems with similar functions. Thus, the agency had to spend resources on maintaining the different IT systems. 2) TD may not always be created and reside in the application where it is handled. For example, TD can be caused in System A by System B because System B transferred the wrong data or data in a wrong format to System A. Thus, to repay the debt, System A must clean the data, and System B must prevent this from happening again. 3) TD is created and revealed by re-sourcing decisions. For example, a change of vendor revealed existing but unknown technical debt in the infrastructure. A second example is when re-sourcing decisions are made so a system is stripped from its developers to support a different project, this decision can cause TD in the system. And 4) TD is created by changes in politics and business needs. For example, TD can be created by changes in political priorities due to parliamentary elections. Or, TD may be created by business needs and thus make the system more complicated to maintain with unnecessary code.

This article makes several contributions. First, it offers deep empirical insight into how an agency managed its TD in its IT portfolio over time. Second, it contributes to problem-solving for practitioners by providing a framework that integrates TD management into IT portfolio management and makes concrete TD management recommendations. Lastly, the framework is also a contribution to TD studies because it relates TD management to IT portfolio management.

7.3 Article 4. "Stakeholder influence on technical debt management in the public sector: An embedded case study"

This article presents a stakeholder analysis of two IT systems. The analysis of the remaining two IT systems will be presented after the summary of this article (section 7.4).

We employed stakeholder theory as a theoretical lens and used it to 1) identify the stakeholders influencing the agency's TD management, 2) map their actions, and 3) identify their influence on TD. We found that most of the stakeholders were non-technical. Likewise for the front-end system, most of the TD stakeholders were external. We argue it is necessary to consider all the stakeholders when creating strategies to manage TD. Identifying how stakeholders influence TD assists in TD prevention and monitoring and thus lowers risks and costs. Furthermore, the stakeholders influenced TD in more than one way and could cause both TD creation and resolution.

The main findings of the study are, first, that the stakeholder landscape was larger than reported in previous studies and larger than what we expected. For example, we identify 12 stakeholders who influence TD activities performed on one of the systems. Second, we found that the stakeholders differ across IT systems. For example, the front-end system had more external stakeholders than the back-end system. Third, most of the identified stakeholders were non-technical. We found that several groups of non-technical stakeholders acted in ways that had consequences for TD. Fourth, stakeholder theory can be a tool to explore how different stakeholders influence TD. For example, the stakeholder group "users" (of IT systems) are inclined to identify TD, while another stakeholder group, "task and funding providers," tend to both resolve and create TD. Fifth, many stakeholders in the public sector are external, and hence difficult to manage. For example, the Agency for Digitization set quality requirements that impacted other authorities. Lastly, we found that Rose et al. (2018) identified values professionalism, efficiency, service and engagement were central to Agency X, a fact reflected in the influence given to the external stakeholders. As a result, the agency served the external authorities quickly and risked creating TD, but also prioritized the IT system and TDM management.

This article contributes to the knowledge on IT maintenance in digital government by providing new empirical insights. Second, it contributes to the knowledge on TD management by identifying the stakeholders who affect an IT systems' TD. Third, it assists practitioners to identify the stakeholders affecting TD management and to map how their actions affect TD.

7.4 Additional findings of two of the systems

This section expands on the findings of Article 4. In that article, I presented an analysis of the two systems called TEXT and REPORT. In this section, I present the analysis of the remaining two systems: CASE and SERVICE. As in the article, I will apply stakeholder theory as an analytical lens, and the findings will follow the structure presented in the article.

7.4.1 System description of the CASE system. The CASE system serves as a case management tool; it integrates the agency with other agencies' IT systems and caseworkers. The system contains several elements: a database for cases, casework, search across business areas, and an index for other IT systems to use. Some business units have developed a custom front-end to the CASE system and primarily use CASE as a database and search engine. At the same time, other caseworkers conduct their casework in this system and use the database and search functions. Should this system have a breakdown, the citizens would not be able to save an entry in the front-end systems, nor would the caseworker be able to work on their cases. Luckily, the system appears to run with only minor difficulties, and the critical issues are prioritized by management.

I identified seven internal stakeholders and two external stakeholders that influenced CASE's TD (Figure 11). The internal stakeholders are 1) other IT systems; 2) IT projects that integrate into the CASE system to store cases; 3) the caseworker representative who works with caseworkers to ensure any difficulties are reported to the application manager; 4) application managers who maintain and develop the organization's IT systems; 5) the IT architects who plan the overall architecture of the systems and ensure that future solutions fit the architectural frame; 6) the IT operations staff who maintain the servers on which the IT systems run; and 7) the board of directors that prioritizes projects and the use of resources. The two external stakeholders are 1) the software companies that develop IT systems with the agency, and 2) the citizens who use the front-end systems to make entries that are saved as cases.

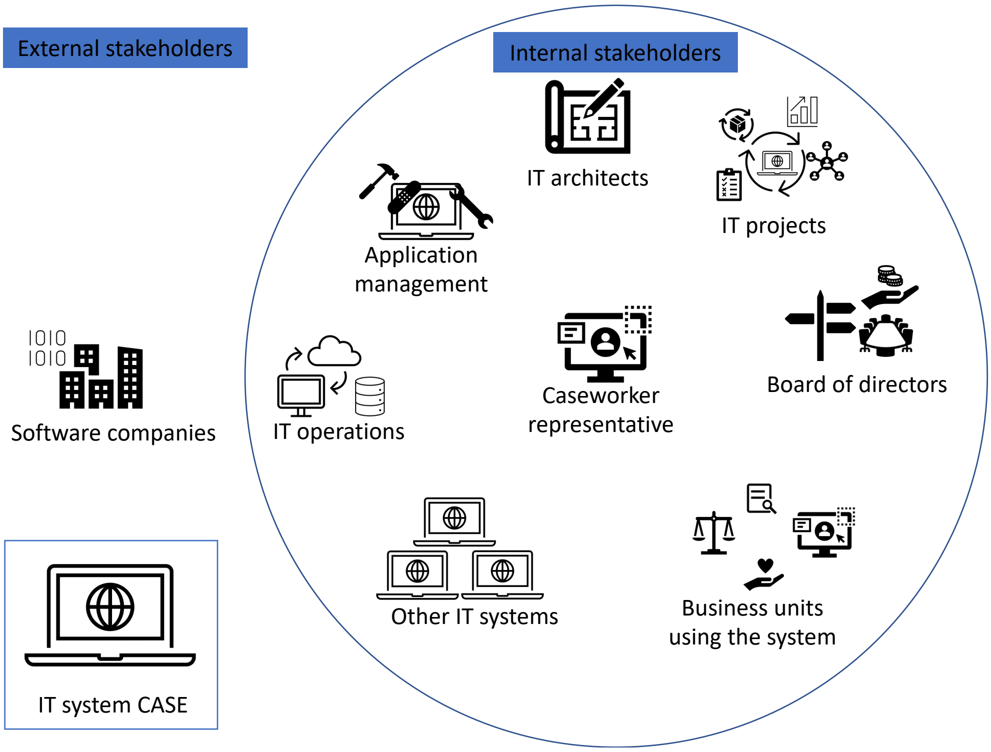


Figure 11: The Internal and External Stakeholders of the IT System CASE

7.4.2. Stakeholders' actions and their TD consequences in the CASE system. I mapped the identified stakeholders, the stakeholders' actions impacting TD, and the TD consequences (Table 4). For example, the application managers (Column 1) perform actions (Column 2) that have TD consequences (Column 3). I categorized the TD consequences according to Rios et al.'s (2018) TD management macro activities classification. However, I have expanded the category "Monitor TD" so it to include workarounds and postponing dealing with TD.

Table 4. A map of stakeholders' actions and TD consequences in the CASE System

Stakeholders	Actions	Consequences
Application managers	Test the system, prioritize tasks, communicate technical needs, explore possible solutions, and plan maintenance on the system	Identify TD, Pay TD, Monitor TD, Prevent TD
Board of directors	Decides priorities and moves resources to projects/tasks that are given higher priority	Pay TD, Create TD
Caseworker representatives	Collect and provide feedback, communicate business needs, and prioritize tasks	Identify TD, Pay TD, Monitor TD
IT architects	Design the system architecture, approve solution descriptions, and provide counsel on technical tasks	Identify TD, Pay TD, Monitor TD
IT operations staff	Maintain the servers that the systems run on and escalate server capacity and power	Identify TD, Monitor TD
Other IT systems/IT projects	Integrates and use CASE, and needs support integrate or use CASE	Create TD
Citizens	Use the system indirectly and give feedback	Identify TD, Pay TD
Software companies	Code systems, provide descriptions of possible solutions, and give feedback	Create TD, Pay TD, identify TD

7.4.3. Stakeholder effects on TD in CASE: An example. The CASE system serves several business applications. The teams developing the applications (IT systems and IT projects) did not always comply with the data formats required by the CASE system due to a lack of knowledge or lack of resources. This fact created performance problems for the CASE system. TD was created and resided in the other IT systems, but the issues were seen as problems in the CASE system. Therefore, it was left to the developers maintaining CASE to clean up the data before it affected the citizens who relied on it by swiftly changing their priorities. TD created by business applications created problems for operations because developers had to change focus and prevent this TD from affecting citizens.

7.4.4. Overview of the stakeholders and their influence. In Table 5, I present the internal and external stakeholders of CASE, their actions, and the TD consequences of these actions. The primary consequence of their actions was to pay TD and identify TD; six of the eight stakeholders had this effect on TD. The least frequent consequence was Prevent TD, which was caused by one stakeholder: application managers.

Table 5. CASE Stakeholders' Influence on TD

Stakeholders	Prevent TD	Create TD	Identify TD	Monitor TD	Pay TD
Application managers	X		X	X	X
Board of directors		X			X
Caseworker representative			X	X	X
IT architects			X	X	X
IT operation team			X	X	
IT systems/ IT projects		X			
Citizens			X		X
Software companies		X	X		X

7.4.5. Remarks on the CASE system. When the CASE system went from IT development to operation, a considerable backlog followed. The maintenance team did not have the resources to keep up, and it took a while before a dedicated caseworker representative was assigned. The application manager was left with a series of questions regarding the backlog and current business needs. A caseworker representative was needed because the numerous business units rely on the CASE system. Furthermore, board of directors deprioritized the system, so it was often stripped of its developer. Thus, a new developer had to be trained. However, management is now focusing on TD and maintenance, and developers have been assigned to maintain the system.

7.4.6. System description of the SERVICE system. The SERVICE system offers a platform where citizens can request a certificate, which is a solemn declaration. The agency collects the relevant information from the different authorities. It requires a fee from the citizens to cover maintenance costs for the agency and the other authorities. When the maintenance expenses change, the fee required of the citizen is affected. In theory, the citizen can ask the different authorities for the information, but the collaborating authorities discourage this as they prefer the order to come from the IT system.

The service began as a manual service and was increasingly digitalized. The system has been refactored on several occasions. Citizens can order the service from the agency's website and find the certificate in their digital mailboxes (which contains correspondence between citizens and the public sector in Denmark). Thus, the system saves time for both the citizens and the authorities by unifying the request process.

7.4.7. Stakeholder mapping of the SERVICE system. I identified six internal stakeholders and three external stakeholders that influenced SERVICE's TD (Figure 12). The internal stakeholders are 1) law graduates; 2) the expert users with caseworkers who ensure any difficulties are reported to the application manager; 3) application managers who manage the maintenance and development of the organization's IT systems; 4) the IT architects who plan the overall architecture of the systems and ensure that future solutions fit the systems' framework; 5) the IT operations staff who maintain the servers on which the IT systems run, and 6) the board of directors that prioritizes projects and the use of resources. The three external stakeholders are 1) the software companies that develop IT systems with the agency; 2) the citizens who use the front-end systems to make entries saved as cases; and 3) the different authorities which supply relevant information.

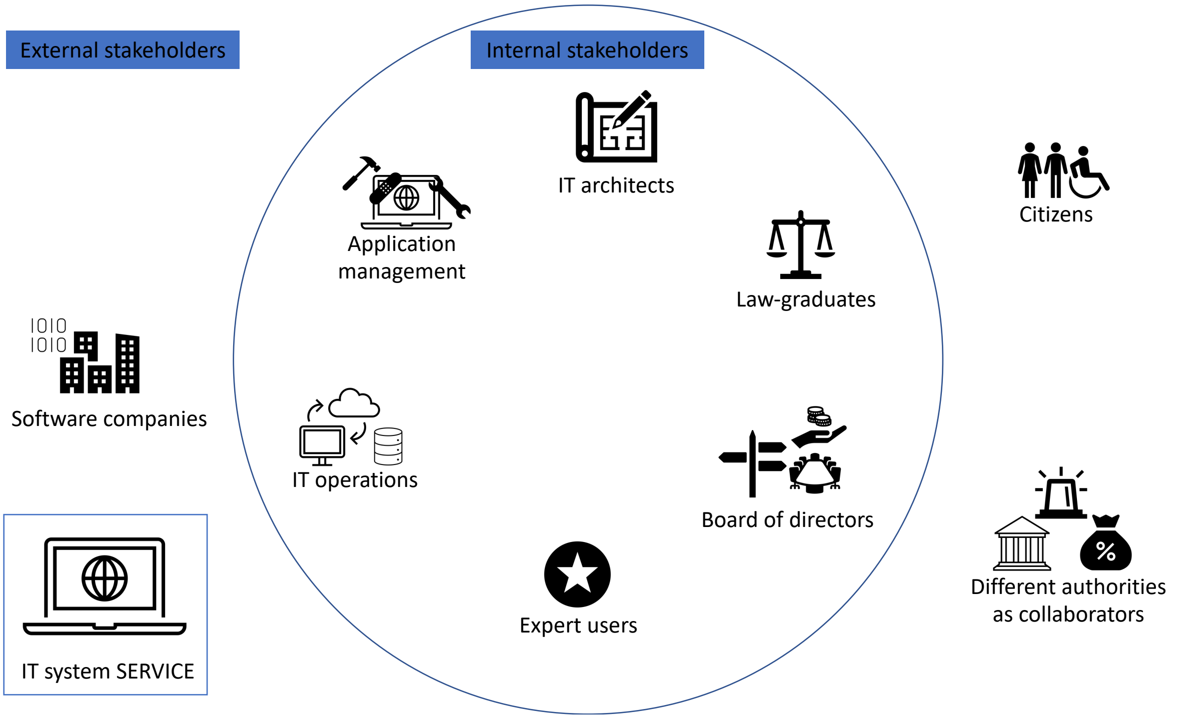


Figure 12: The Internal and External Stakeholders of the IT System SERVICE

Table 6. A Mapping of Stakeholders' Actions and TD Consequences in the SERVICE System

Stakeholders	Actions	Consequences
Application managers	Prioritize tasks, communicate business needs, explore possible solutions, and plan future maintenance on the system	Identify TD, Pay TD, Monitor TD, Prevent TD
Board of directors	Decides priorities	Pay TD, Create TD
Expert user	Provide feedback, prioritize tasks, and test the system	Prevent TD, Identify TD, Pay TD
IT architects	Provide counsel on technical tasks	Pay TD, Monitor TD, Prevent TD
IT operations staff	Maintain the servers the systems run on	Identify TD, Monitor TD
Law-graduates	Communicate other authorities' needs and ensure compliance with legislation	Prevent TD, Pay TD
Citizens	Provide feedback	Identify TD, Pay TD
Different authorities	Integrate with the system and provide feedback	Pay TD
Software companies	Code systems, provide descriptions of possible solutions, and provide feedback	Create TD, Pay TD, Identify TD

7.4.8. Stakeholders' actions and their TD consequences in the SERVICE system. Table 6 presents my map of the SERVICE systems' stakeholders, their actions, and the consequences of these actions on TD.

For example, the expert user (Column 1) uses the system, provides feedback, prioritizes tasks, and tests the system (Column 2). These actions can prevent TD, identify TD, and pay TD (Column 3). As explained in Article 4, I categorized the TD consequences according to Rios et al.'s (2018) classification of TD management macro activities. However, I made a few adjustments, adding the category "Create TD" and expanding "Monitor TD" to include workarounds and postponing dealing with TD.

7.4.9. Stakeholder effects on TD in SERVICE: An example. SERVICE is a front-end system, and a modification was needed to improve the user experience and increase efficiency for the caseworkers. However, SERVICE was deprioritized by the board of directors and stripped of its developers on several occasions. Furthermore, SERVICE was affected by a move of servers initiated by IT operations, which blocked development. Depriving the system of resources and freezing the development made the change more costly and time-consuming than initially planned.

7.4.10. Overview of the stakeholders and their influence. In Table 7, I present the internal and external stakeholders of SERVICE, their actions, and the TD consequences of those actions. The primary consequence of their actions is to pay TD; seven of the eight stakeholders have this effect on TD. The least frequent consequence is Create TD. It was caused by two stakeholders: the board of directors and software companies.

Table 7. SERVICE Stakeholders' Influence on TD

Stakeholders	Prevent TD	Create TD	Identify TD	Monitor TD	Pay TD
Application managers	X		X	X	X
Board of directors		X			X
Law graduates	X				X
Expert users	X		X		X
IT architects	X			X	X
IT operation team			X	X	
Citizens			X		X
Different authorities					X
Software companies		X	X		X

7.4.11 Remarks on the SERVICE system. Even though SERVICE is self-funded, it experienced challenges by being deprioritized by the board of directors. The system is relatively simple and does not have a database, unlike many other IT systems within Agency X. This makes it easier to maintain but also more vulnerable to errors in the back-end systems like CASE.

SERVICE is forced to wait for CASE's maintenance team to solve errors. It is perceived as an “easy” system with few errors, and the application manager is often changed. As a result, it gets a low priority.

7.5 Comparing the analysis of CASE and SERVICE with findings from Article 4

In the article, I compared the REPORT and TEXT systems. But now I find it more interesting to compare SERVICE and CASE with the analysis of REPORT and TEXT.

SERVICE, like REPORT, is externally funded. SERVICE is funded by the fee the citizens pay to use the service. Both SERVICE and REPORT are front-end systems and have more external stakeholders than the back-end systems. However, unlike REPORT, SERVICE was deprioritized. This fact indicates it is not the funding source that caused the low priority of SERVICE. Instead, the reason for the priority appears to depend on the political focus that there is on REPORT, in addition to Agency X’s responsibility to develop the system for a different authority and used by another authority. Furthermore, this finding debunks one of my assumptions, that front-end applications do not experience being deprioritized. Not all of Agency X’s front-end systems are self or externally funded, and I have not explored the division between funded and non-funded front-end systems.

Other authorities have chosen Agency X to develop several IT systems on their behalf (such as the REPORT system) due to Agency X’s expertise in the area. With development comes funding. However, it also increases the agency’s portfolio of systems. Additionally, when resources are limited, the agency must prioritize between IT systems developed for other authorities and IT systems developed for its own use.

The TEXT and CASE systems are back-end systems, and they both encounter TD problems, which were created by a different IT system. This tendency could be something interesting for future studies to explore.

Lastly, the analysis of SERVICE and CASE does not appear to change the overview of stakeholder influence on TD in Agency X (Table 7 in Article 4). Instead, it supports the findings in the article and generates a more nuanced understanding of how stakeholders influence TD in the given IT systems.

8. High-level findings

In this chapter, I present and explain the dissertation's theoretical contributions. To understand what is meant by theory and what constitutes a theoretical contribution, I draw on Gregor (2006) and Whetten (1989) and summarize their overall arguments. First, I present four empirical examples of TD, one from each IT system. Then, I use Gregor and Whetten to identify the central theoretical elements in the examples, before highlighting the causes and impact of TD.

8.1 What constitutes a theoretical contribution?

In this sub-chapter, I examine the meaning of theory and theoretical contribution, as set out by Gregor (2006) and Whetten (1989).

Gregor (2006) argues that while data can be the foundation of theoretical development, it does not constitute a theory on its own. Gregor (2006) further argues that theory should include a level of generalization and a form of explanation that is linked to the idea of causation. The body of knowledge about causal connections is the foundation of an explanation for (i.e., cause of) an event or a prediction, and a theory can be derived from either. Gregor distinguishes between four types of causation:

(1) Regularity: things that happen regularly (a type of causation often used in the natural sciences).
(2) Counterfactual analysis: for instance, if A does not happen, then neither will B.
(3) Probabilistic causal analysis: for example, social processes and mechanisms are likely to make A happen, which increases the probability of B.
(4) Manipulation or teleological causal analysis: for instance, if you turn on the light switch, the light is turned on.

The goals of explanation and prediction are central to different understandings of a theory. An explanation can be provided to induce a subjective state of understanding in an individual. Explanations include notions of causality that refer to causal mechanisms, including teleological-type causes.

Gregor (2006) outlines different perspectives on theory, which together present it as an abstract creation that aims to describe and explain phenomena, and to enhance understanding of the world. In some cases, theory also makes predictions about what will happen in the future, providing a basis for intervention and action. Gregor (2006, p. 620) presents a taxonomy of five types of theory in information systems research (Table 8).

Table 8: Gregor's (2006, p. 620) taxonomy of theory types in information systems research

Theory Type	Distinguishing attributes
1. Analysis	Say what is
2. Explanation	Says what is, how, why, when, and where
3. Prediction	Says what is and what will be
4. Explanation and prediction	Says what is, how, why, when, where, and what will be
5. Design and action	Says how to do something

Gregor (2006) presents the following theory components: means of representation, constructs, statements of relationship, and scope (the degree of generalizability of the statements of relationship). These components are contingent on theory purpose, causal explanation, testable propositions (hypotheses), and prescriptive statements. What Gregor (2006) outlines is similar to Whetten's (1989) guidance for developing a theory. This focuses on four central elements that a theory must contain:

(1) What: the factors that should be considered as part of the explanation of the social phenomenon of interest (descriptive).
(2) How: the relationships between the factors. These introduce the causality (descriptive).
(3) Why: the underlying dynamics that justify the selection of factors and the causal relationships (explanatory).
(4) Who, when, where: the contextual and temporal factors. These act as conditions and place limitations on the generalizability (scope).

Whetten (1989) claims that "why" is the most fruitful, but also the most difficult aspect of theory development. Here it is beneficial to borrow perspectives from other fields, as these can challenge the underlying rationale. Whetten also acknowledges the inherent causal nature of a theory, even though the researcher might not have been able to adequately test all the causal relationships ("how"). Whetten further argues that a theory (or theoretical model) is only useful to researchers when all the relationships it describes have not been tested. If they have all been tested, then the theory has little value in the laboratory and is ready for the classroom.

Gregor's and Whetten's concepts are used to theorize findings and propose models that seek to explain the causal relationships between different elements and events. In Section 8.2, I explain "what is" TD by giving examples of where, how, why, when, and where it occurs. In section 8.3, I examine "why" TD happens and say when it is likely to do so, thus predicting "what will be." In this way, I create an "explanation and prediction" theory (Type 4 in Gregor's taxonomy).

8.2 Examples of TD processes

In Chapter 7, I presented several examples of TD (including Article 3). To further explore how TD occurs in Agency X, I present four empirical examples and highlight the stakeholders, TD activities, and TD type. In doing so, I apply stakeholder theory and Rios et al.'s (2018) categorizations (as I did in Article 3). I also illustrate the examples in separate models, in order to map the different activities. Following Gregor, these empirical examples together create a theory, as they describe and, to some degree, explain TD, as well as when, where, and how it occurs. In Section 8.3, I explore the causal relationships between the different stakeholders, activities and IT systems.

Example 1: SERVICE

The IT system SERVICE offers a non-mandatory, fee-based service to citizens. The IT system communicates with *other authorities* [stakeholders] to obtain the information needed to provide this service. The *citizen* [stakeholder] must upload an ID card to gain the certificate that the IT system provides. The IT system validates the ID card, expedites the case, and sends it to relevant authorities (Figure 13). This process usually proceeds as expected without any problems.

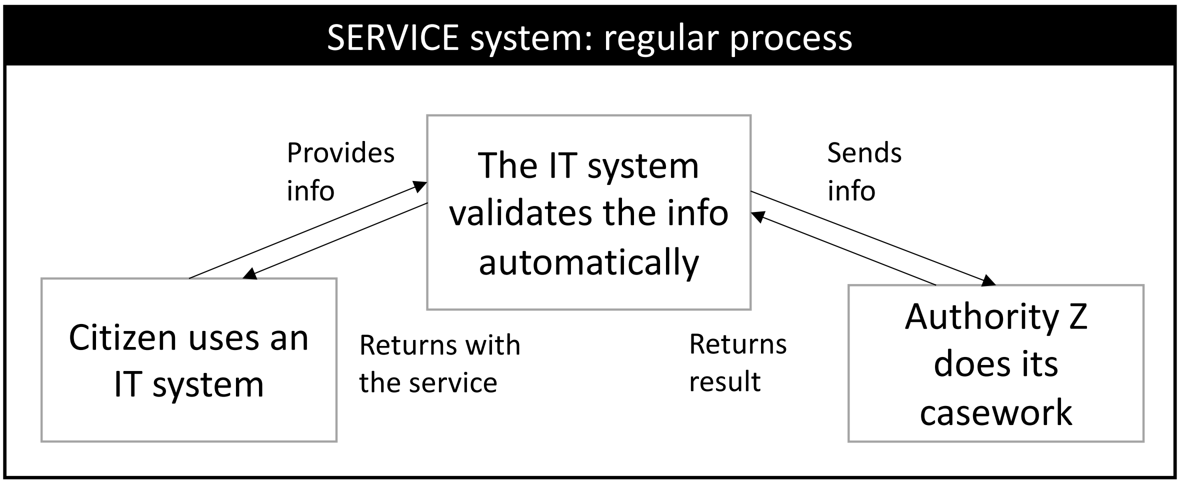

SERVICE system: regular process
The IT system validates the info automatically
Provides info
Sends info
Citizen uses an IT system
Returns with the service
Returns result
Authority Z does its casework

Figure 13: The regular process of a citizen interacting with the SERVICE system

However, *one authority* [stakeholder] notices that a document containing a citizen's foreign ID card is invalid (Step 1). This identification deviates from the regular process (Figure 13) and is illustrated in Figure 14. The authority informs Agency X about this matter (Step 2). Agency X looks into it and identifies a general issue with the validation of foreign ID cards [TD activity: identify debt] (Step 3). The IT system should be able to identify both valid and invalid ID cards, regardless of whether they are national or foreign. That the system cannot do so is an example of requirement debt [TD type: requirement debt]. Agency X realizes that the technology cannot detect invalid foreign ID cards, so it decides to *create a workaround and let foreign ID card validation be a part of manual casework*, instead of being covered by its digital processes [TD activity: monitor TD] (Step 4). To repay the TD, the Agency must replace the validation technology with an improved or alternative version that can validate foreign ID cards successfully (Figure 14).

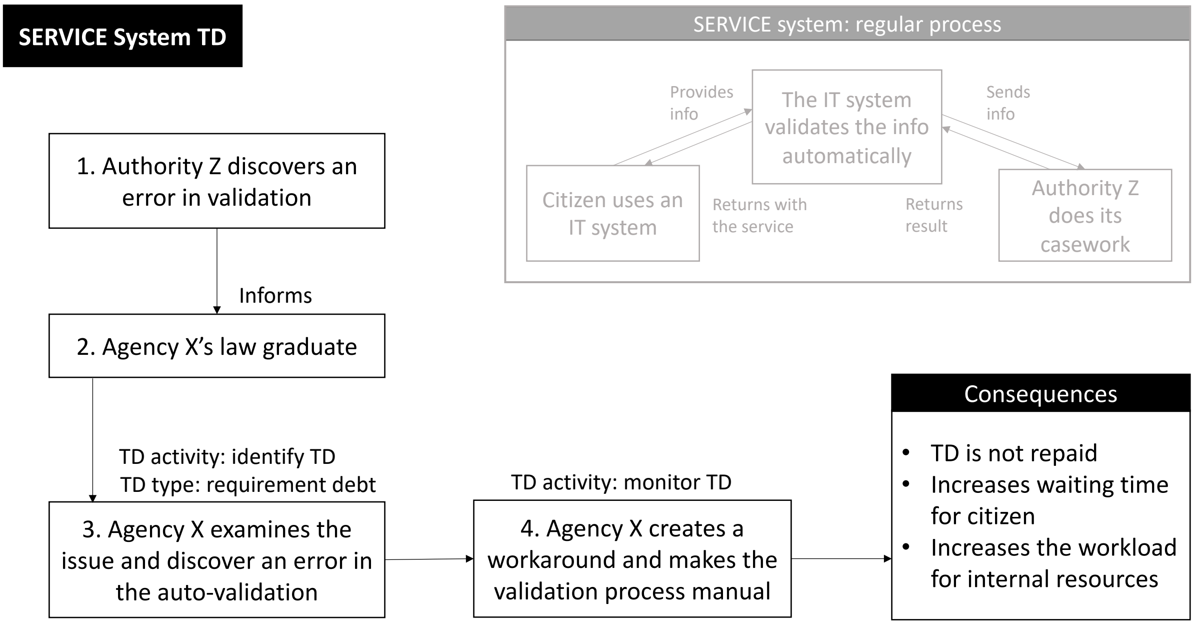

SERVICE System TD
1. Authority Z discovers an error in validation
Informs
2. Agency X's law graduate
TD activity: identify TD
TD type: requirement debt
3. Agency X examines the issue and discover an error in the auto-validation
TD activity: monitor TD
4. Agency X creates a workaround and makes the validation process manual
Consequences
TD is not repaid
Increases waiting time for citizen
Increases the workload for internal resources
SERVICE system: regular process
Provides info
The IT system validates the info automatically
Sends info
Citizen uses an IT system
Returns with the service
Returns result
Authority Z does its casework

Figure 14: An example of TD identification and management within the SERVICE system

Usually, citizens request documentation from the SERVICE system, which the system automatically provides them. However, the requirement debt enables foreign citizens to upload invalid ID and obtain “valid” documentation, even though they should not be able to do so. The organization’s workaround does not repay the actual TD. Instead, it increases the workload for internal personnel, such as the caseworker. Additionally, the TD may increase the waiting time for the citizen. This is problematic because, for authorization purposes, other organizations rely on citizens being able to provide the documentation. Fortunately, the other authority identified the TD, which enabled Agency X to respond. However, the example shows how unmanaged TD lowers Agency X’s credibility and makes it appear unprofessional in the eyes of another authority.

Example 2: TEXT
TEXT is a content management system developed by Agency X. It is a core system that allows the developer to draw on its functionality and not hardcode text snippets into the different IT systems. Instead, developers can use the TEXT system to create a reference that the communication and business units [stakeholders] can edit as they see fit. The citizen receives a letter or uses a system that draws the text snippets from the TEXT system. The regular process for this is illustrated in Figure 15.

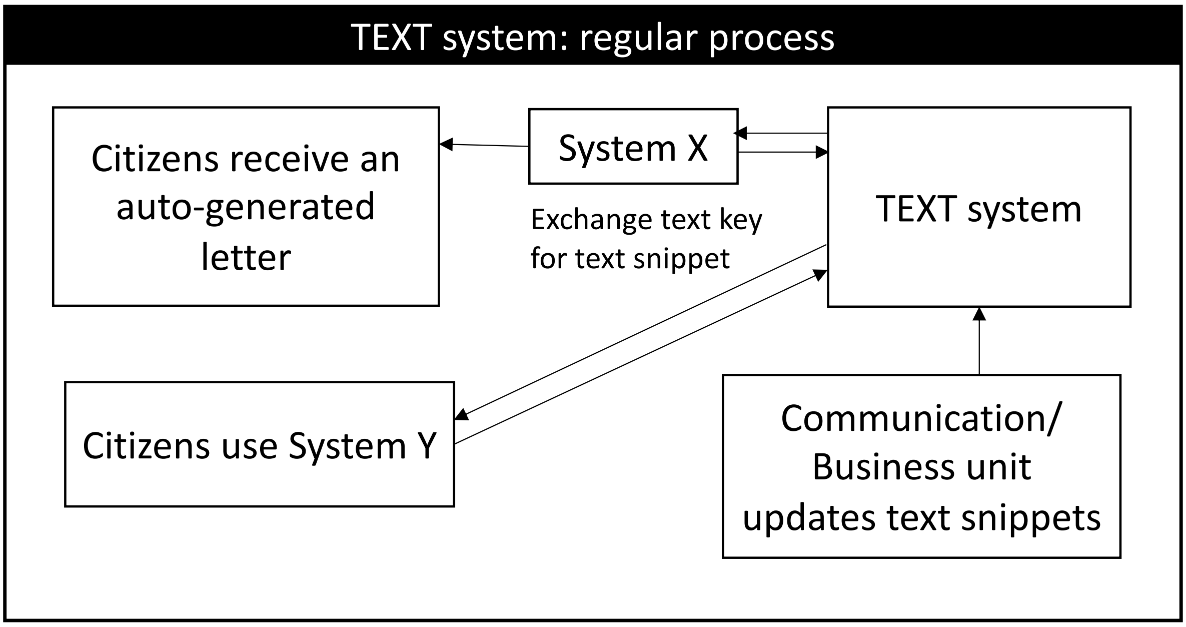

TEXT system: regular process
Citizens receive an auto-generated letter
System X
Exchange text key for text snippet
TEXT system
Citizens use System Y
Communication/ Business unit updates text snippets

Figure 15: The regular process of the TEXT system providing text to front-end systems or letters to citizens

As displayed in Figure 16, Agency X uses the TEXT system to create automated letters for citizens [stakeholders]. A letter typically consists of 12 components. When the different letters are constructed, *the Agency reuses existing pieces created for other letters* [TD activity: create TD] (Step 1). However, *the reuse of the pieces has not been documented* [TD type: documentation debt] (Step 2.1). The *communication or business unit* [stakeholder] would like to change the wording of these letters to make the message clearer for citizens and lower the volume of calls to the support center. However, if *they make a change in one of the text-snippets, they are likely to change several letters* [TD type: process debt] (Step 2.2). This debt (along with other types of debt) *has caused several employees to change work tasks or* *positions to avoid working with this system* [TD activity: create TD] [TD type: people debt] (Step 3).

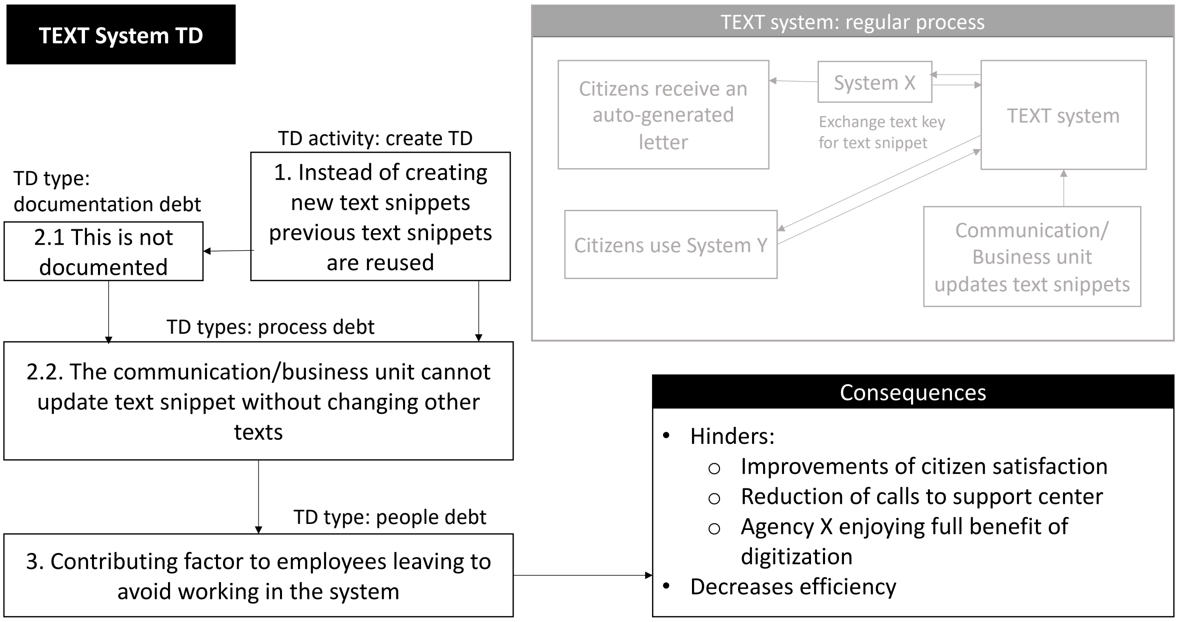

TEXT System TD
TEXT system: regular process
Citizens receive an auto-generated letter
System X
Exchange text key for text snippet
TEXT system
Citizens use System Y
Communication/ Business unit updates text snippets
TD activity: create TD
1. Instead of creating new text snippets previous text snippets are reused
TD type: documentation debt
2.1 This is not documented
TD types: process debt
2.2. The communication/business unit cannot update text snippet without changing other texts
TD type: people debt
3. Contributing factor to employees leaving to avoid working in the system
Consequences
Hinders:
Improvements of citizen satisfaction
Reduction of calls to support center
Agency X enjoying full benefit of digitization
Decreases efficiency

Figure 16: An example of identifying TD within the TEXT system

The TD in this example may hinder Agency X in achieving its aim of improving the level of customer satisfaction among citizens. Moreover, if the Agency cannot edit official letters to make them more comprehensible to citizens, it cannot limit the volume of calls to the support center. The TD also increases the difficulty of navigating and working in the TEXT system, *requiring more resources than necessary* [TD type: process debt]. This process debt decreases the efficiency of the Agency, preventing it from reaping the full benefits of digitizing the letters. Additionally, the TD in the IT system discourages employees from working with the system. This leads to employee turnover, and thereby causes knowledge loss in the Agency. In general, the IT system requires more resources due to the debt. The debt should therefore be repaid, either by replacing the system or fixing the issues.

Example 3: CASE

The third example involves the CASE system, an internal case management system developed by Agency X. The system contains information that management and others can use for statistical analysis and reporting (Figure 17). However, the system was initially not expected to handle a high number of cases.

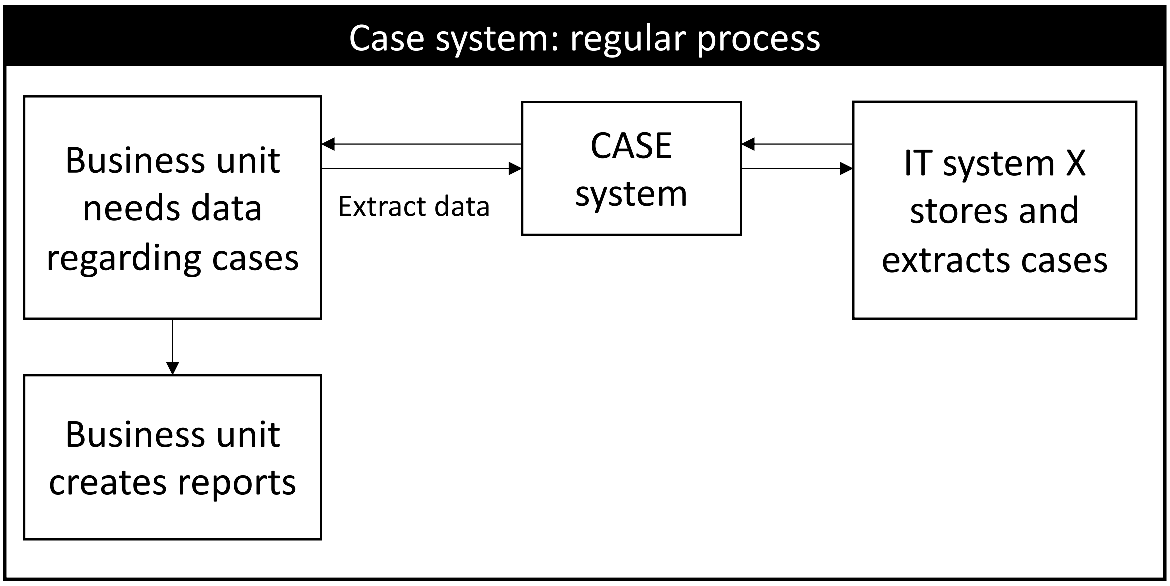

Case system: regular process
Business unit needs data regarding cases
Extract data
CASE system
IT system X stores and extracts cases
Business unit creates reports

Figure 17: The regular process of data extraction using the CASE system

In Figure 18, I illustrate the following example of TD identification and the consequences thereof. In a project, a *caseworker* [stakeholder] needs to extract large datasets from the CASE system; yet the numbers do not add up (Step 1). The project starts to mistrust the data, particularly the data quality (Step 2). The caseworker discovers that *the CASE system cannot handle vast data extractions, and thus returns them incomplete* [TD type: requirement debt] (Step 3.1). However, this limitation is *not documented nor widely known* [TD type: documentation debt] (Step 4.1). To retrieve the data extractions, the caseworker must make smaller batches, *which is more time-consuming* [TD type: process debt] (Step 4.2). Furthermore, the caseworker [stakeholder] suspects poor data quality (step 3.2), and others share this concern. The deficiency of data quality is likely a consequence of the CASE system's development process and its various integrations into *other IT systems* [stakeholder]. The CASE system supports *several business units* [stakeholders] in their casework. This means that there are competing demands for further development of the CASE system, as the units do not all use the system in the same manner. These competing demands have led to *numerous labels, some overlapping, whose purpose and usage are not documented centrally* [TD activity: create TD] [TD type: documentation debt]. The labels are used when creating statistics and reports; yet the lack of consistency and documentation makes it difficult to create these accurately.

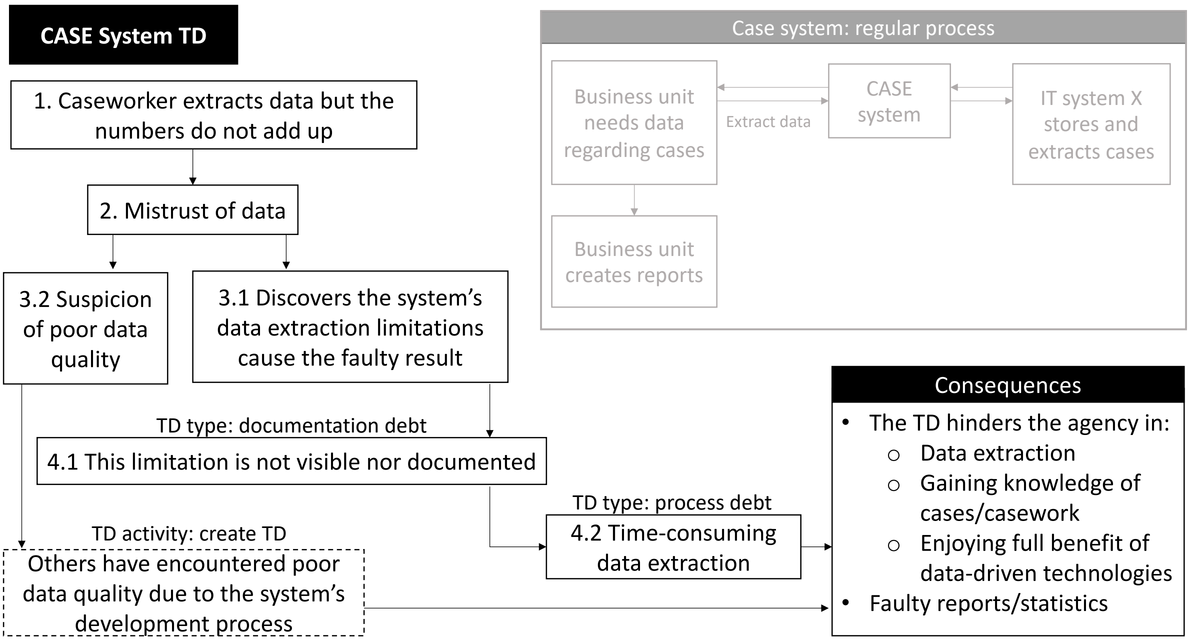

CASE System TD
1. Caseworker extracts data but the numbers do not add up
2. Mistrust of data
3.2 Suspicion of poor data quality
3.1 Discovers the system's data extraction limitations cause the faulty result
TD type: documentation debt
4.1 This limitation is not visible nor documented
TD type: process debt
4.2 Time-consuming data extraction
TD activity: create TD
Others have encountered poor data quality due to the system's development process
Case system: regular process
Business unit needs data regarding cases
Extract data
CASE system
IT system X stores and extracts cases
Business unit creates reports
Consequences
The TD hinders the agency in:
Data extraction
Gaining knowledge of cases/casework
Enjoying full benefit of data-driven technologies
Faulty reports/statistics

Figure 18: An example of identifying TD within the CASE system

As stated in Article 3 (Nielsen & Skaarup, 2021), it is difficult to predict a business' future needs, as these needs might change. In this instance, the unknown element is how many cases to expect and the scale of data extraction. The TD in this example could hinder Agency X in terms of acquiring knowledge about its casework and gaining an insight into which initiatives are beneficial. The TD could also prevent the Agency from enjoying the full benefits of new data-driven technologies. Additionally, if the reports produced by the CASE system are needed for external usage, the TD may damage perceptions of Agency X's credibility and professionalism. Not all TD can necessarily be avoided. Nonetheless, Agency X has taken steps to prevent some of the TD from occurring, for example, by appointing a product owner representative who decides which business needs should be solved in the CASE system and which belong in the respective IT systems for individual business units.

Example 4: REPORT

The IT system REPORT offers a platform where citizens [stakeholders] can report specific activities. This information is then used by *authorities* [stakeholders] to conduct casework. The IT system is developed on behalf of *a different authority* [stakeholder], who is responsible for providing the service to *citizens* [stakeholder], and the *authorities who need it to perform their casework* [stakeholders] (Figure 19).

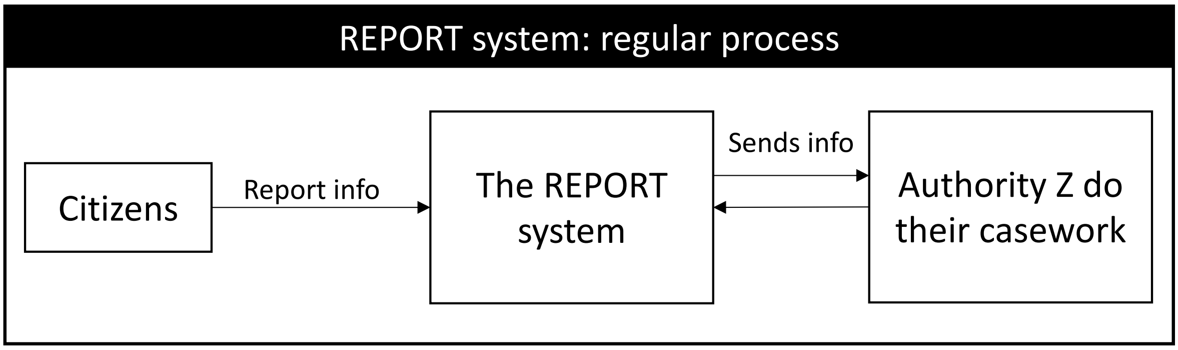


Figure 19: The regular process of citizen interaction with the REPORT system

As displayed in Figure 20, the REPORT system is often affected by political decisions made by both *Danish politicians and the EU* [stakeholders]. For example, the EU has passed legislation that requires a new group of citizens to register their activities in the REPORT system, so that a new kind of control can be carried out (Step 1). This information is different compared to what other citizens register, and thus requires the REPORT system to be further developed (Step 2). Because of the political attention that the REPORT system receives, there are frequent changes to this development as well as a strict deadline (Step 3). The IT system is prioritized because of the political interest. If developers are working on other tasks, *they have to hand over these tasks to others or return to them later, as the REPORT system takes precedence* [TD activity: create TD] (Step 3.1). The development of the REPORT system is varied, and the system has *undergone several refactoring projects* [TD activity: TD repay]. Political decisions steer the development of the system (Step 4), meaning there is no concrete vision or long-term plan. As a result, *the development of the system becomes incoherent, branching off in different directions* [TD activity: TD creation]. These *changes in business needs make it difficult to make suitable architectural decisions* in time [TD type: architecture debt]. Therefore, the *law graduate* [stakeholder], the product owner of the system within Agency X, argues that it is time to *rethink the system and consider refactoring it* [TD activity: repay TD and prevent TD].

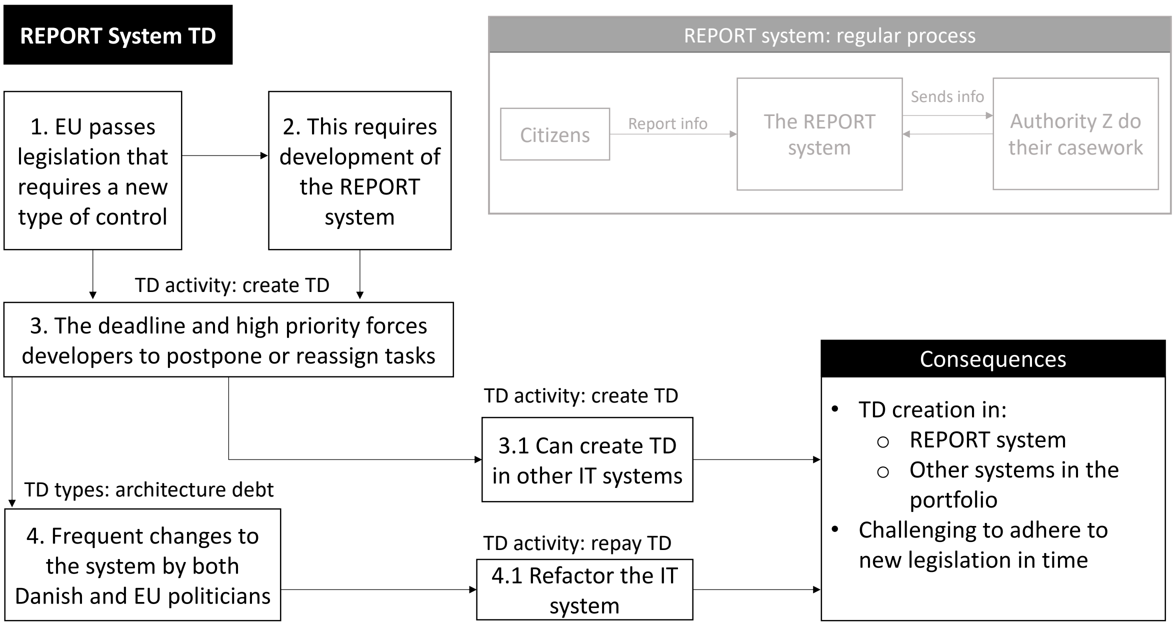

REPORT System TD
1. EU passes legislation that requires a new type of control
2. This requires development of the REPORT system
TD activity: create TD
3. The deadline and high priority forces developers to postpone or reassign tasks
TD activity: create TD
3.1 Can create TD in other IT systems
TD types: architecture debt
4. Frequent changes to the system by both Danish and EU politicians
TD activity: repay TD
4.1 Refactor the IT system
Consequences
TD creation in:
REPORT system
Other systems in the portfolio
Challenging to adhere to new legislation in time
REPORT system: regular process
Citizens
Report info
The REPORT system
Sends info
Authority Z do their casework

Figure 20: Identifying and repaying TD in the REPORT system

In this example, Agency X strives to adhere to legislation and meet the deadlines set by politicians. It prioritizes the development of the REPORT system so that it can appear professional and efficient and offer the required services to citizens. However, this may cause TD in other IT systems, as deadlines and poor planning are known TD causes (Ramač et al., 2022). Additionally, the political focus on the IT system makes it challenging to develop an architecture that is in line with politicians' wishes.

8.3 Overarching theory

Following Gregor (2006) and Whetten (1989), this subchapter highlights the causal relations and theoretical elements in the four examples presented above. I briefly re-illustrate the examples in four models, each showing the causal relations of the four systems presented in Section 8.2. The reader will, therefore, find some overlap between sections 8.2 and 8.3. Based on the empirical material collected at Agency X, I present an explanation and prediction theory, which explains why TD is created and when different TD activities are likely to take place.

SERVICE example

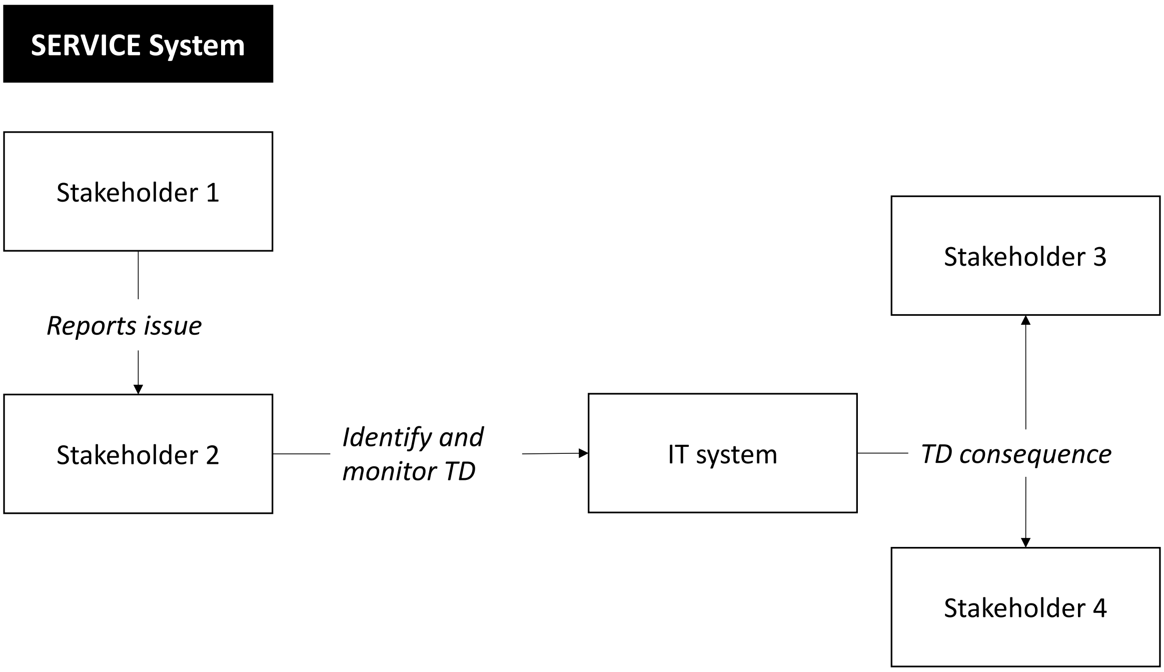


Figure 21: The relationships in a TD scenario within the SERVICE system.

In the SERVICE system example, we saw an external stakeholder identifying an issue and informing an internal stakeholder about this issue. The internal stakeholder finds the root cause and identifies TD. The agency then decides to manage the TD by creating a workaround, making the casework manual instead of digital. The TD affects both caseworkers, who must now carry out processes manually, and citizens, who may face longer waiting times (Figure 21). This example contains probabilistic *causal analysis* (Gregor, 2006: p. 620), in that an issue is identified by a stakeholder receiving information from Agency X. The erroneous automatic validation of foreign ID cards indicates underlying TD (likely, requirement debt). As the documentation that citizens obtain is used for authorization purposes, it is problematic that they can obtain it without a valid ID. The agency thus seeks to address the TD immediately. Yet managing the debt affects other stakeholders. Furthermore, the workaround requires extra resources; specifically, caseworkers spend more time on their work.

TEXT example

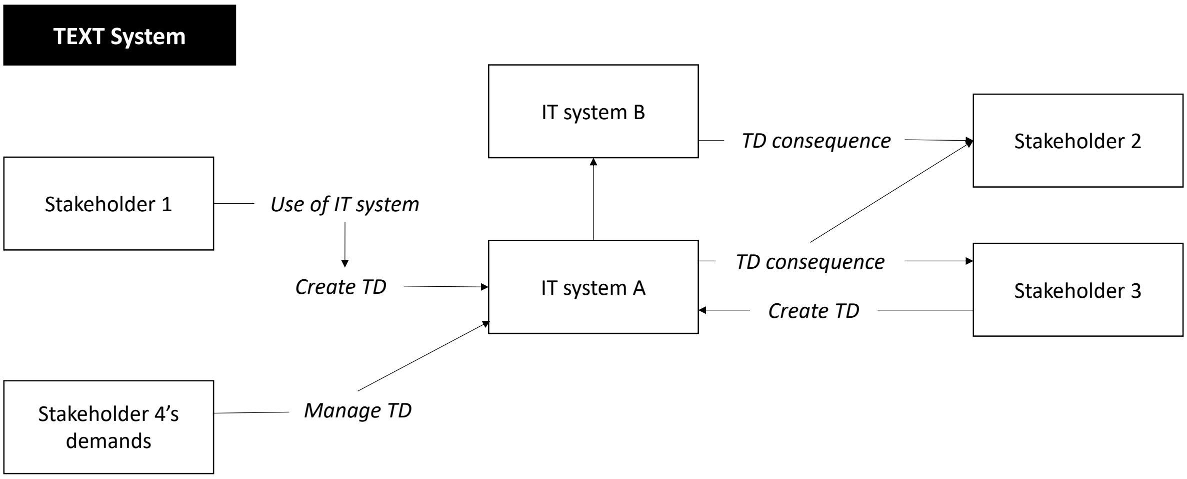


Figure 22: The relationships in a TD scenario within the TEXT system.

The TEXT system example shows how stakeholders' use of this system creates TD. Both citizens presented with the text (indirectly through a different IT system) and people working in the TEXT system are affected by the TD. This is because other IT systems and the letters sent out to citizens depend on text from the TEXT system. The TD has made it difficult to change the text, which hinders communication with citizens. Furthermore, the TD has led to frequent changes in the people working in the TEXT system, creating additional debt (people debt). In this example, the causal relationship corresponds to *counterfactual analysis* (Gregor, 2006, p. 620). Had the employees used the system in the way it was designed, the TD would not have occurred. Additionally, as the TD (in the form of process debt) contributed to frequent changes in employees, it also helped to create further TD (in the form of people debt). Thus, there is a probabilistic causal analysis: if TD occurs, it is likely to generate more TD (i.e., other types of TD). Earlier in the dissertation, I discussed how the Digitization Agency's requirements lead Agency X to make an action plan for their IT systems, and thereby encourage TD management. For the TEXT system, this causal relation leads to a plan for TD repayment (Figure 22).

CASE example

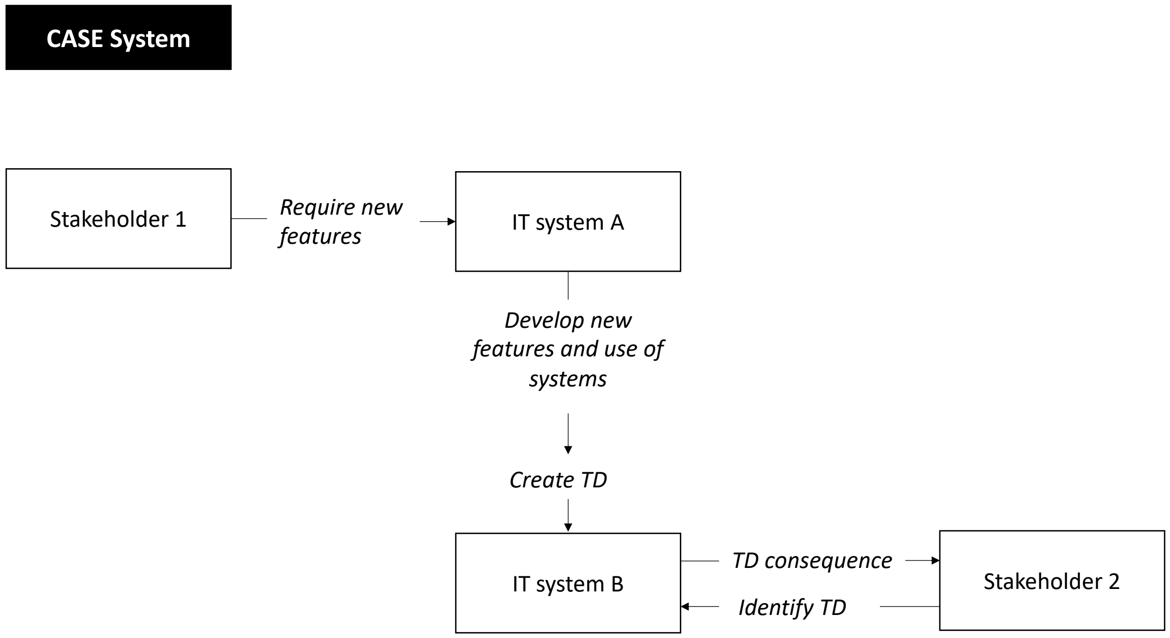


Figure 23: The relationships in a TD scenario within the CASE system.

The systems integrated into the CASE system generate TD within it. As illustrated in the example in Section 8.2, this can start with a stakeholder requiring a functionality offered by an IT system. This IT system then integrates into the CASE system, meaning the stakeholder's requirement contributes to the development of the system (Figure 23). However, such development is sometimes a shortcut. For example, it can lead to the creation of new labels when existing labels could have been used. Additionally, it can lead to a case being labelled to control who has access to it, when the proper solution would be to go through the user access management system. However, the pressure of deadlines leads to the creation of an architecture debt. These examples illustrate a *probabilistic causal analysis* (Gregor, 2006, p. 620). The core IT systems, which several IT systems integrate into, are prone to TD. Other IT systems use the core systems for shortcut solutions because of deadlines, limited resources, and the core systems' lack of solid ownership. Lastly, the stakeholders affected by the TD are typically the ones who identify it. In the example in Section 8.2, poor data quality and unknown limitations of the IT system affect a caseworker's ability to generate reports. The caseworker identifies the root cause of the problem, and then works around this to obtain the dataset.

REPORT example

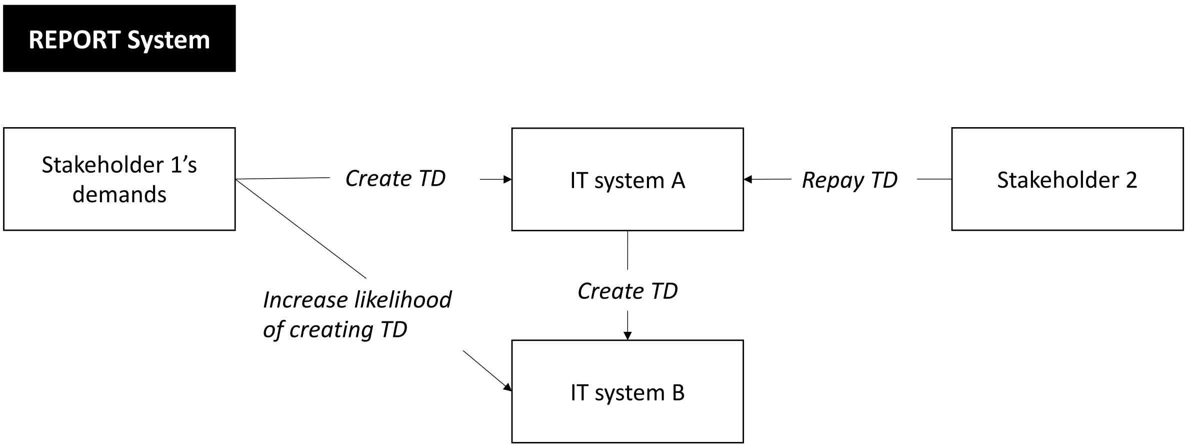


Figure 24: Relationships in a TD scenario within the REPORT system.

In the REPORT example, I illustrated how Danish and EU politicians' frequent demands for legislation and control, which the REPORT system supports, create TD in this system. Development to address the demands leads to architecture debt, as the existing architecture is unable to support new demands. In order to avoid architecture debt, new demands must align with or be incorporated into the architecture. The law graduate in the example decides to repay the architecture debt to better support the politicians' requirements of the REPORT system. In addition, the political focus on the system distorts its priorities, with employees having to be ready to drop or hand over tasks in order to work on it (Figure 24). Thus, attention from politicians increases the likelihood of TD creation in other IT systems. The relationship corresponds to a *probabilistic causal analysis* (Gregor, 2006, p. 620), as the politicians' deadlines and frequent demands increase the likelihood of TD creation.

To sum up, this subchapter explains why TD happens and predicts when it will occur. The causal relationships displayed in the examples correspond to *probabilistic causal analysis* and *counterfactual analysis* (Gregor, 2006, p. 620). Below, in Table 9, I summarize five factors that increase the likelihood of TD creation and explain the causal relationships. The five factors are drawn from both from the current chapter and from other parts of the dissertation (articles 3 and 4).

Table 9. Five overarching factors that increase the likelihood of TD creation

TD creation factors	Description
(1) Core IT systems	Massive integration into an IT system (typically known as a core IT system) increases the number of stakeholders and demands on this system. There is a tendency for other IT systems to make shortcuts in the core IT system. Thus, moving the responsibility of repaying the debt to the core system speeds up the other systems' development. The benefit of having a core IT system is to reuse the functionality so that every IT system does not have to contain the same functionality, for example, sending messages or text editing. Instead, other systems can draw on the core IT system's functionality. However, it can be challenging to prioritize the core system over other IT systems. This means that **core systems are more prone to TD**.
(2) Lack of ownership	The benefit of having a stakeholder with solid ownership of an IT system is that someone feels responsible for "protecting" it. They seek to prevent potential TD and ensure TD repayment. Thus, a **lack of active ownership of an IT system makes it more prone to TD**.
(3) Low priority	**IT systems with a low priority are more prone to TD**. This TD usually only affects internal employees or the system's future development. As a result, the organization does not prioritize TD prevention and repayment. Additionally, high-priority systems take precedence because public organizations adhere to public sector values: professionalism, efficiency, service, and engagement.
(4) Frequent demands	**Frequent demands on tasks or new development can cause TD**, either in the IT system that carries out the tasks or in the surrounding IT systems. Frequent demands for new development can hinder strategic development, increase the risk of the development becoming incoherent, and make it challenging to create a suitable architecture. Demands often come with a deadline, and so employees are pulled from other tasks, increasing the risk of TD elsewhere.
(5) Unanticipated use	**Unanticipated use of an IT system** can increase the complexity of the system and cause TD. It can decrease data quality, complicate maintenance, and compromise the future purpose and usage of the IT system. Furthermore, unanticipated use is rarely documented and may be difficult to fix. It often appears due to shortcuts, deadlines, and a lack of insight into how the IT system was meant to be used. Thus, poor implementation of the system and a lack of documentation are key contributing factors.

9. Contributions in relation to previous studies

Drawing on Mathiassen's (2017) categorization (Figure 4), this chapter outlines the contributions of this dissertation and relates them to previous studies. I present (1) empirical contributions within the specific research area (IT maintenance in the public sector); (2) theoretical contributions; and (3) practical contributions that aim to improve TD management.

Based on my literature review (Article 1), I map recent TD research and identify several gaps in the literature. Most importantly, I argue that TD research could benefit from a digital government approach. In my exploration of digital government research, I found TD to be of significance to this field. Therefore, I propose a research agenda for future digital government scholars. I recommend that researchers study TD in the public sector and apply a qualitative approach, focusing on either the organizational or individual level.

9.1 Empirical contributions

In the literature review, I found that data sources and techniques were limited to IT professionals and coding, and were primarily of a quantitative nature. Previous research has also largely concentrated on the private sector and open-source projects. Against this background, this dissertation provides new empirical insights into TD management in the public sector. Using various methodological techniques, it captures the way researchers and practitioners employ metaphors about TD and OGD. Moreover, through an embedded case study of an agency's IT development, the dissertation provides insights at both the system and IT portfolio levels, and then scales the findings to the wider public sector. The research is supplemented by a quantitative and qualitative exploration of metaphor use surrounding TD in both the research and practice communities. According to Gregor (2006), this kind of contribution can be considered theoretical contribution (Type 1) as it explains "what is" TD.

Furthermore, I provide empirical insight into stakeholders affecting IT systems. Previous research—for example, Besker et al. (2017), Ghanbari et al. (2017), and Besker et al. (2018b)—has focused on IT professionals and their experiences with TD. However, like Klinger et al. (2011), I find that several non-technical stakeholders are involved in TDM activities. Additionally, I identify some stakeholders external to the selected agency, although these differ from IT system to IT system. In general, I find the stakeholder landscape to be larger than expected, extending far beyond the IT department. I recognize that not all stakeholders can be managed, and that some TD is inevitable. The importance lies in recognizing the TD and mitigating it—if possible, by identifying the stakeholders and being aware of their impact on TD.

9.2 Theoretical contributions

In the literature review of TD research (Article 1), I found theory on, and clear theoretical contributions to TD was very limited. In order to address this gap, this dissertation proposes two theoretical frameworks—as presented in articles 2 and 3—as well as four models of TD processes and relationships (Chapter 8).

Previous literature has attempted to explain metaphors (Lakoff & Johnson, 2008; Black, 1955; Morgan, 1998) and some researchers (Oates & Fitzgerald, 2007; Kendall & Kendall, 1993) have identified metaphor use within the field of IT. However, to my knowledge, previous research has not proposed an analytical framework for metaphors. The first framework (CAOM) I present uses metaphor theory, along with the possibilities and limitations of metaphors, to explore metaphor development in communities. The CAOM framework provides an understanding of the TD metaphor and how practitioners and researchers use it. I found that the debt metaphor is often combined with other metaphors, thus extending its explanatory possibilities and enabling it to be used for different proposes, for instance, as a framework. This realization can help us improve how we communicate technical phenomena and understand how they develop. A theoretical contribution of this kind is what Gregor (2006) considers a Type 2 theory—one concerned with what is, how, when, where, and why TD processes take place.

In Article 3, we present a framework that incorporates TD management into IT portfolio management, and illustrates how the two relate. We apply the framework and find TD can occur in different areas of the IT portfolio. Thus, TD can reside both in the IT portfolio and in individual IT systems. TD can be caused by different kinds of decisions, from poor resource planning to assumed business needs. I recommend identifying the TD in the IT portfolio and becoming aware of what caused it. It then becomes possible to prevent TD, or at least to identify and monitor it. Portfolio theory as the mathematical term has been used as an analytical lens to improve decision-making about how to best manage TD (Guo & Seaman, 2011; Seaman et al., 2012; Albarak & Bahsoon, 2018). In Article 3, IT portfolio management helps to set TDM in a broader context and visualize where and how TD appears. In line with types 4 and 5 in Gregor's (2006) taxonomy of theory, Article 3 makes several theoretical contributions, exploring what is, how, when, where, and why, as well as predicting future TD events and making practical recommendations.

Lastly, in Chapter 8, I describe and illustrate four empirical examples of TD processes and explain the causal relations between stakeholders, activities, IT systems, and TD. I then identify five factors that can be used to predict TD creation. This theoretical contribution can be considered a Type 4 theory (Gregor, 2006), as it explores what is, how, when, where, and why, while also predicting future TD events. Previous TD research has mapped TD causes and effects, along with preventive and payment actions (Freire et al., 2020; Pérez et al., 2021; Ramač et al., 2022; Rios et al., 2020). However, it has not empirically described TD processes nor explored the stakeholders that impact these causes. In the field of digital government, Balta et al. (2015) demonstrate how stakeholders and resource dependencies influence the progress of IT projects, thus displaying the importance of stakeholders and their influence. Although previous TD research offers valuable insights into TD and IT development, this dissertation provides rich empirical descriptions that uniquely illustrate the complexity of TD and how and why it occurs.

9.3 Digital government perspective and TD

Within software engineering, TD has been known to challenge the continuous development of existing IT systems; it can also cause errors that are hard to fix. TD can cause the IT system to be temporarily or permanently inaccessible. Its consequences can be severe, therefore, and are not limited to private organizations. Throughout this dissertation, I have illustrated how TD can be viewed in a digital government context. In this section, I apply the main concepts of digital government research to identify how TD impacts public organizations and the government. Digital government research involves several key concepts and takes place on multiple levels. To explain the magnitude of TD, I relate it to Scholl's (2007) model of the key concepts within digital government research (Figure 25).

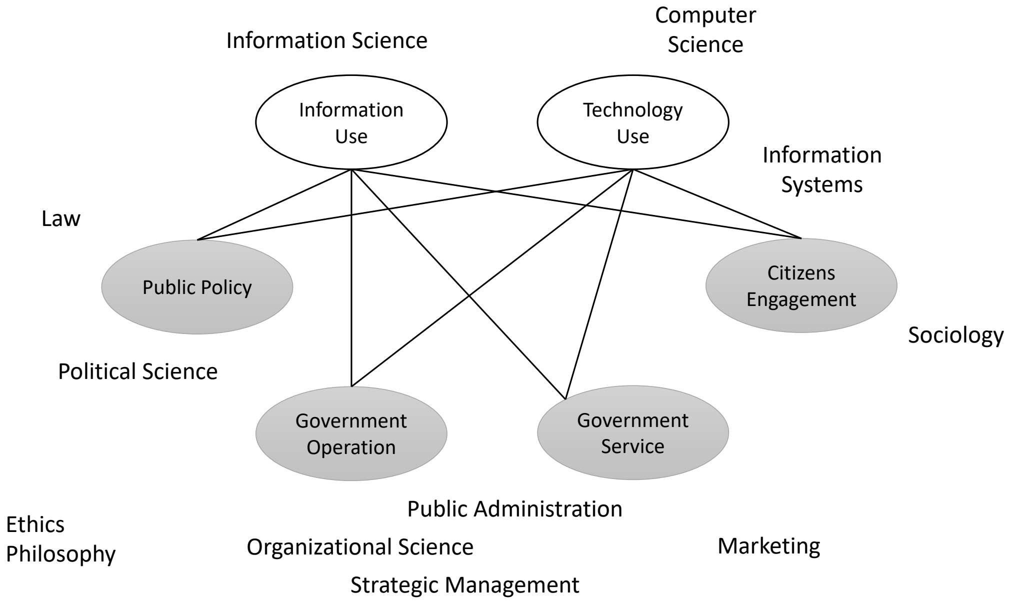


Figure 25: Scholl's (2007) model of the key concepts within digital government research

The CASE example illustrates how TD can impact the information usage, such as data quality and data availability. This type of TD can hinder technology usage and impact digitization in an agency. Several new technologies are data-driven (e.g., machine learning), so TD can hamper innovation and the adoption of new technologies. Additionally, TD decreases efficiency and complicates automated casework, to the extent that caseworkers may be forced to do casework manually (Example 1, SERVICE; Example 3, CASE), which affects government operations (and their efficiency). We have seen how Agency X is required to spend resources in order to manage these problems, prevent them from happening again, and clean up the data.

TD can impede an agency in supporting a minister and in meeting other authorities' request for data assistance, as it might not be possible to extract the necessary data (Example 3, CASE). Thus, TD impacts public policy and makes it challenging for an agency to know which initiatives are beneficial. Even more importantly, public policymakers impact the creation of TD by applying pressure and constantly requesting new development, without providing a long-term vision (Example 4, REPORT). Additionally, TD can lead to an agency no longer adhering to legislation or failing to provide the services that enable citizens to adhere to legislation.

TD may also influence the engagement of citizens, as it can hinder the improvement of government communication, and thereby affect the service they receive from public organizations (Example 2, TEXT). Additionally, we have seen in the case of Agency X how citizens can wrongfully obtain documentation as a result of problems with automatic validation (Example 1, SERVICE).

To sum up, the implications of TD vary across societal and organizational levels. It can hinder or complicate casework and allow citizens to obtain documentation they were not supposed to. It may also impede the delivery of exact data to other authorities and ministries. Additionally, TD has several internal consequences. It can hinder the adoption of new technologies, complicate the further development of IT systems, and make casework more costly in terms of time and resources. Lastly, it can complicate supporting legislation and may even lead to the infringement of legislation.

9.4 Practical contributions

The CAOM framework can help practitioners understand the possibilities of metaphors and avoid pitfalls in metaphor application. Furthermore, the framework provides practitioners with a repertoire of actions and interpretations for metaphor development. I encourage reflection on what a metaphor entails; for example, a metaphor affects how a phenomenon is perceived. Practitioners can use the framework to analyze metaphors to help them avoid pitfalls in their communication. Furthermore, the CAOM can be used to gain a deeper insight into a metaphor and the phenomenon.

I explore TD management strategies and recommend that organizations employ a holistic portfolio-wide approach to resource planning and consider the full consequences of ad hoc changes in resource allocations. The article gives concrete recommendations; TD tasks are bundled into TD projects, and the tasks are included in TD resolution during the development of existing applications. Additionally, I recommend carefully documenting TD that may be created when withdrawing resources from a project and addressing TD during IT development projects rather than "exporting" TD issues to maintenance and operations.

I recommend that organizations recognize the stakeholders who influence TD in different directions and keep in mind that stakeholders vary depending on the IT system. Several of the stakeholders are external or non-technical. It is therefore important to identify and consider them in TD management.

Agency X prioritized the IT systems and projects according to Rose et al.'s (2018) values. For example, REPORT was prioritized because Agency X has a responsibility to assist other authorities and to follow legal regulations. Thus, other IT systems were deprioritized, even though monetary resources were available for their development. Therefore, agencies should be aware that IT systems developed on behalf of other authorities take high priority. One solution is for the agencies to sign service agreements with several software companies, so they don't "steal" their own resources when prioritizing the IT systems created for other authorities.

Rios et al. (2018) map the different TDM activities and the tools and strategies belonging to the different activities. They ask future studies to create more strategies to prevent, identify, and repay TD. This dissertation has offered strategies to support these TDM activities. I have inserted the strategies developed through this PhD into Rios et al.'s (2018) TD management landscape (Figure 26).

I encourage stakeholder mapping (ST mapping) to be applied during TD prevention. This will make the organization aware of the stakeholder, and enable it to assess whether they pose a threat or an opportunity for collaboration. In turn, this will allow the organization to determine the stakeholder's possible impact on TD. The organization can then take action to

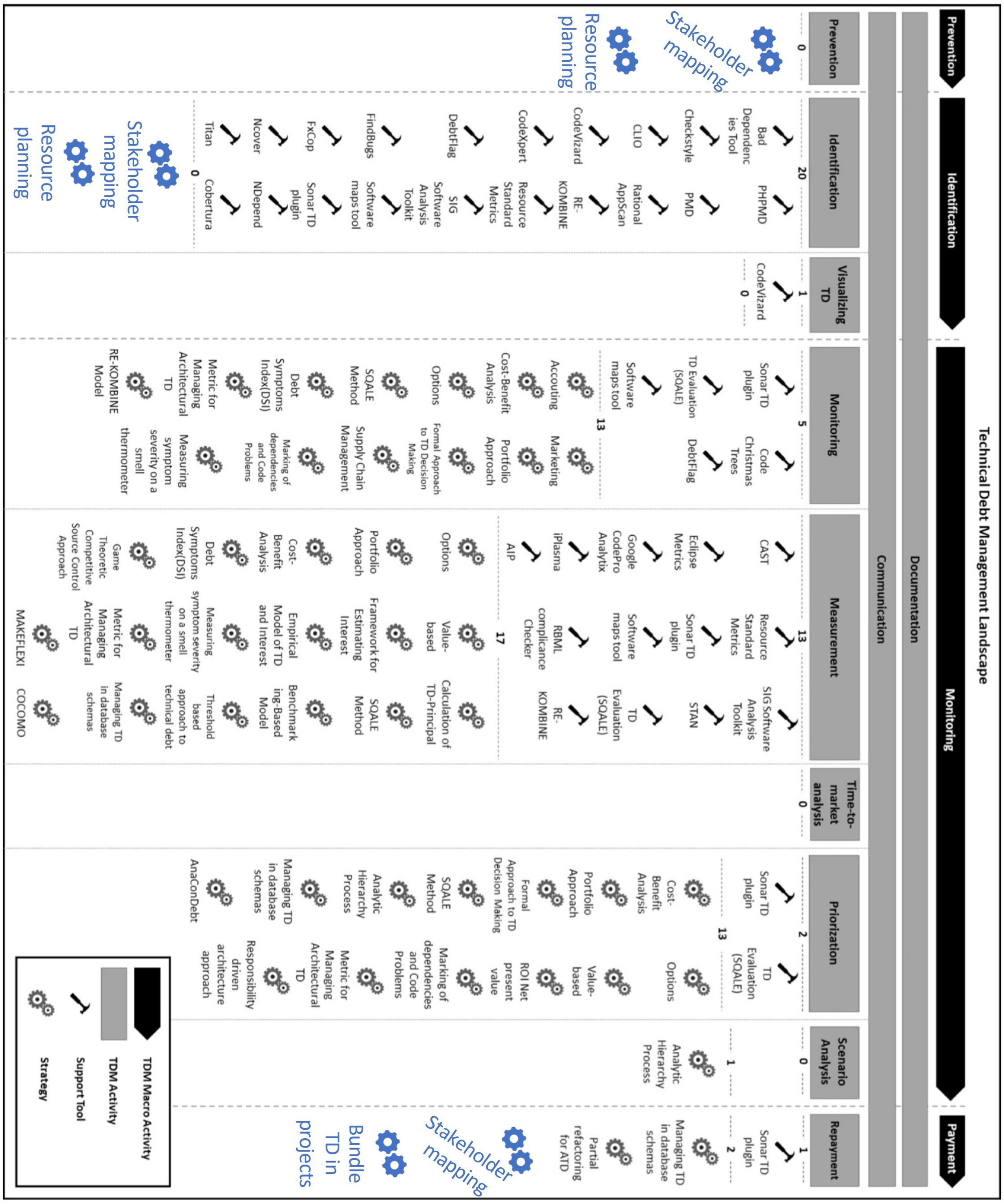


Fig. 5. TD management landscape: Activities, strategies, and tools.

mitigate the risk the stakeholder poses or otherwise manage their potential impact. Furthermore, ST mapping can be used to identify TD, as it gives insight into when TD is likely to occur. Lastly, it can increase an organization's awareness of which stakeholders impact TD repayment. This can help an organization achieve a higher repayment of TD and identify IT systems that do not have stakeholders with this kind of impact.

Figure 26: This dissertation's strategies incorporated into Rios et al.'s (2018) TD management landscape

I found that resource planning incorporated into IT planning can prevent TD from being created. Without this incorporation, the planning of task-dependencies will fail to take into account the resources shared in IT development and maintenance. Usually, the same person can oversee or maintain several IT systems simultaneously because the task level is relatively low. However, suppose the same person must deliver on several IT systems within a short time frame. This could lead to the creation of TD. Therefore, resource planning should be used to identify when and where TD can occur, and to prevent it from occurring. Lastly, incorporating TD repayment tasks into projects, either incorporating it as large tasks on existing systems or bundling relating TD repayment tasks as a project, ensures TD is prioritized on the same level as development projects.

10. Conclusion, limitations, and future work

This chapter presents the conclusion and answers briefly the overarching research questions. Then, the limitations of this research are presented together with suggestions for future research.

10.1 Summarizing how I addressed the research questions

The research questions posed in the introduction have guided the research presented in this dissertation. The following section addresses the research questions more concisely and highlights the implications for both research and practice.

Table 10. An overview of the thesis' research questions and the articles who address them

Research question	Article
RQ1 How can the concept of technical debt be operationalized to improve the management of IT systems in the public sector?	1, 2, 3, 4
RQ2: How is technical debt defined and studied in the existent academic literature?	1, 2
RQ3: How and why is technical debt created in a public organization?	3, 4
RQ4: How does technical debt become visible in a public organization?	3, 4
RQ5 How can the recognition and management of technical debt become part of IT portfolio management?	3

The overarching research question of this dissertation was how the concept of "technical debt" could be operationalized to improve the management of IT systems in the public sector. This question is addressed throughout the four articles (Table 10). I argue that, by understanding the metaphorical limitations of the concept, TD can be operationalized by implementing TD management within IT portfolio management. Additionally, organizations can benefit from recognizing the stakeholders who influence TD in different directions.

10.1.1 How is technical debt defined and studied in the existent academic literature? First, I conducted a literature review and explored how academic research defines TD (Article 1). The literature review was supplemented by exploring TD as a metaphor (Article 2). Here I developed a framework that can be used to understand the applications and limitations of metaphors within a specific context.

When I initially began the literature review, I wanted to know how TD was defined by academia. The original introduction of TD by Cunningham (1992) define TD as: "*Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite … The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt*" (p. 30). Several TD definitions has since then been proposed (Avgeriou et al., 2016), and researchers still discuss the TD definition (Fairbanks, 2020). I find many of the definitions inadequate as they are limited to source code. Most researchers refer to the original definition by Cunningham. As I argue in Article 3, I find it necessary to expand TD to an enterprise-level, following Klinger (2011) and

Magnusson et al. (2018). Klinger (2011, p. 35) defines TD as "*the increased cost of changing or maintaining a system due to expedient shortcuts taken during its development*." Magnusson et al. (2018, p. 78) create a new umbrella term, 'technological debt,' which they define as the "*accumulated obligation owned by current CIO (the debtor) to future CIO (the creditor), where previous decisions limit prospective decisions."* I rely on both of their definitions in my analysis of TD.

Moreover, I explored the TD as a metaphor. I find the metaphor has some limitations. These limitations are called "tensions" in Article 2. Despite the limitations of the TD metaphor, they appear to be addressed by other metaphors such as "repair." Furthermore, a metaphor requires the receiver to be familiar with the original object and able to see a connection between the original object and the phenomenon (Lakoff & Johnson, 2008). If the connection is not clear it can complicate applications of the metaphor.

10.1.2 How and why is technical debt created in a public organization? The original definition of TD by Cunningham encompasses an explanation of why and how TD is created (presented above). Developing an IT system entails creating TD (Cunningham, 1992). Furthermore, TD is created because it speeds up development. The other circumstances contributing to TD are explored within software engineering (Rios et al., 2018). However, the literature review (Article 1) revealed that TD research has not been conducted in the public sector (Nielsen et al., 2020).

Through an embedded case study, I explored how and why TD was created and identified. Moreover, I found TD is created on an IT portfolio level by management decisions about resourcing, assumed business needs, and political priorities (Nielsen & Skaarup, 2021). Furthermore, different stakeholders influence a system's TD, and these stakeholders are external and internal but also technical and non-technical. By exploring TD from a stakeholder theory perspective, I answer "why" TD is created. I draw on the organizational values in the public sector identified by Rose et al. (2018). The values "professional" and "effective" shape management's priorities for IT systems (Nielsen & Madsen, 2022). These priorities must be established because there are limited resources. For example, REPORT was highly prioritized due to responsibilities to other authorities. In addition, TEXT was further analyzed, and the issues were identified and addressed based on the Danish Agency for Digitization's reports and recommendations. CASE was continuously stripped of development resources until it experienced errors that affected users.

Lastly, I propose four models of how TD processes occur and explain the
identify five factors which increase the likelihood of TD creation (table 9): 1) Core IT systems, 2) Lack of ownership, 3) Low priority, 4) Frequent demands, and 5) Unanticipated use.

10.1.3 How does technical debt become visible in the organization? TD becomes visible in different ways, as illustrated in Articles 3 and 4. An example I mention is a move of servers that made TD documentation visible. Additionally, the Agency for Digitization's analysis of the government's IT systems made agencies map the condition of their IT systems. The analysis was followed by making IT action plans for the IT systems, which were found to be inadequate. Furthermore, the systems' users (caseworkers and citizens) experienced errors that originated from TD. Moreover, the developers and testers experienced the TD during the development and maintenance of the systems. Some of the developers kept track of these items on a separate log, while others had their own log of TD items or did not record them.

10.1.4 How can the recognition and management of technical debt become part of IT portfolio management? In Article 3, I developed a framework for integrating TD management into IT portfolio management (Figure 10). Moreover, I applied the framework and showcased how TD management occurs across different activities and sub-elements in IT portfolio management. Lastly, I presented how to integrate TD management into IT portfolio management and how to become aware of the consequences of reprioritizing resourcing.

10.2 Overall conclusion

With this dissertation, I wanted to encourage management to address TD and realize their involvement in TD activities. Therefore, I set out to explore how the concept of "technical debt" can be operationalized to improve the management of IT systems in the public sector. To address this research question, I conducted an embedded case study at a Danish agency. The articles included in the dissertation collectively address the question.

I argue that TD should be managed to avoid IT system breakdowns and errors that affect citizens. I deliver empirical insights into TD management in a Danish agency by mapping the stakeholders affecting TD and highlighting TD occurrences in the IT portfolio. Furthermore, I present insights into how the TD metaphor is applied in research and by practitioners. Then I propose two frameworks. The first is the CAOM framework, which captures metaphor development and the possibilities of metaphor application. Second, I propose a framework that integrates TD management into IT portfolio management. Lastly, I highlight the contributions and recommendations about how to manage TD.

I conclude that TD can be operationalized by using TD on a higher level of abstraction and implementing TD management in the IT portfolio management. For example, I found that TD occurs in different areas of the IT portfolio and should be discussed at general IT planning meetings. It is essential to recognize the metaphorical limitations of the concept and the possibilities of combining TD with other metaphors. I propose models which explains TD processes and predict when TD is likely to happen. Additionally, I recommend that organizations recognize the stakeholders who influence TD in different directions and keep in mind the stakeholders vary depending on the IT system.

10.3 Limitations

This section will highlight two main limitations of this dissertation. Throughout the research, I made several choices regarding direction, research philosophy, and method. The summary (kappa) entails reflections of the choice of direction; I will now elaborate on my methodological and philosophical choices and their consequences.

I chose to conduct a case study with an embedded single-case design. This choice limited the study to one agency with subunits; thus, the findings cannot be statistically generalized (Yin, 2018). However, this dissertation aims not to generalize but to explore and understand TD in the public sector and generate recommendations. The case study allowed for developing new approaches to TD management (Eisenhardt & Graebner, 2007). I developed and presented two frameworks that I applied in the research, and I found both frameworks beneficial to understanding TD. However, these frameworks have not been tested empirically by other scholars, nor have they been implemented in an organization. If other researchers applied and further developed the framework, it would be validated and its analytical power would be increased (Eisenhardt & Graebner, 2007).

10.4 Future research

This dissertation is a step toward understanding TD processes and how to manage TD in the public sector. It offers empirical, theoretical, and practical contributions to TD management. Based on the research conducted throughout this PhD process, I make the following four proposals for future research.

Firstly, I encourage scholars to **replicate this study in other public or private organizations**. By replicating the study in public organizations, it will be possible to further generalize the findings and uncover any differences in, for instance, countries with a different digitization history. Meanwhile, replicating this study in private organizations would reveal whether the findings are applicable in the private sector and how they might differ. In general, replication would validate and increase the analytical power of the findings set out in this dissertation (Eisenhardt & Graebner, 2007).

Second, I considered comparing source code quality, as assessed by tools and experienced by users, architects, and software developers. I chose not to pursue this direction, but I propose that future scholars follow it as my preliminary findings were promising. I found that the assessment of source code by the software quality assessment tool was vastly different from how the system’s quality was experienced by software developers, architects, and users. Thus, I suggest future studies **compare the results from software quality assessment tools on source code** (e.g., Software Improvement Group's SIGRID, SonarQube, or similar) **to how relevant stakeholders perceive the system’s quality** through surveys, interviews, observations, or a combination thereof.

Third, I suggest scholars **conduct a longitudinal study examining TD processes and exploring TD management strategies**. A longitudinal study of TD and its management could provide more indicators on when TD is created and when specific strategies are beneficial. This would help to build on previous research (Ramač et al., 2022; Rios et al.,2020) on TD creation factors, as I have outlined in Table 10. Additionally, future studies could further expand on Freire et al.’s (2020) and Pérez et al.’s (2021) research on TD prevention.

Fourth, I found that policymakers significantly impact TD management in public organizations. Policymakers’ requirements in terms of how public organizations evaluate their IT systems contribute to TD management, particularly TD repayment. During my PhD, policymakers have shown interest in understanding TD and have requested tools and strategies to assist public organizations in their TD management. Therefore, I recommend that scholars **explore how TD management strategies can be incorporated into policy documents.** This research would assist policymakers (e.g., ministers) in facilitating TD management, repayment of TD, and preventing TD creation in public organizations.

References

Aagaard, P., & Pedersen, J. S. (2022). Digitalising Denmark: Efficiency versus privacy. In Public Governance in Denmark. Emerald Publishing Limited.

Agency for Digitization. (n.d.). 20 års fælles digitaliseringsstrategier. Digitaliseringsstyrelsen. Retrieved April 6, 2022, from https://digst.dk/strategier/digitaliseringsstrategien/20-aars-faelles-digitaliseringsstrategier/

Agency for Public Finance and Management. (n.d.). Finanslovsudarbejdelse. Økonomistyrelsen. Retrieved April 6, 2022, from https://oes.dk/oekonomi/finanslov-og-udgiftsopfoelgning/finanslovsudarbejdelse/

Albarak, M., & Bahsoon, R. (2018, May). Prioritizing technical debt in database normalization using portfolio theory and data quality metrics. In Proceedings of the 2018 International Conference on Technical Debt (pp. 31-40).

Al-Ibraheem, N., & Ruël, H. (2011). In-House vs. Off-the-shelf e-HRM applications. In Enterprise Information Systems: Concepts, Methodologies, Tools and Applications (pp. 752-775). IGI Global.

Alves, N. S., 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.

Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., & Avgeriou, P. (2015). The financial aspect of managing technical debt: A systematic literature review. Information and Software Technology, 64, 52-73.

Andersen, K. N., Henriksen, H. Z., Medaglia, R., Danziger, J. N., Sannarnes, M. K., & Enemærke, M. (2010). Fads and facts of e-government: A review of impacts of e-government (2003–2009). International Journal of Public Administration, 33(11), 564-579.

Anderson, K. V., & Henriksen, H. Z. (2008). The first leg of e-government research: Domains and application areas 1998-2003. Electronic Government: Concepts, Methodologies, Tools, and Applications, 8-22.

Avgeriou, P., Kruchten, P., Ozkaya, I., & Seaman, C. (2016). Managing technical debt in software engineering (dagstuhl seminar 16162). In Dagstuhl Reports (Vol. 6, No. 4). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Balta, D., Greger, V., Wolf, P., & Krcmar, H. (2015, January). E-government stakeholder analysis and management based on stakeholder interactions and resource dependencies. In 2015 48th Hawaii International Conference on System Sciences (pp. 2456-2465). IEEE.

Bannister, F., & Connolly, R. (2015). The great theory hunt: Does e-government really have a problem?. Government Information Quarterly, 32(1), 1-11.

Becker, C., Chitchyan, R., Betz, S., & McCord, C. (2018, May). Trade-off decisions across time in technical debt management: a systematic literature review. In Proceedings of the 2018 International Conference on Technical Debt (pp. 85-94).

Behutiye, W. N., Rodríguez, P., Oivo, M., & Tosun, A. (2017). Analyzing the concept of technical debt in the context of agile software development: A systematic literature review. Information and Software Technology, 82, 139-158.

BenIdris, M. (2020). Investigate, identify and estimate the technical debt: a systematic mapping study. Available at SSRN 3606172.

Berg-Sørensen, A., Grøn, C. H., & Hansen, H. F. (2011). Organisering af den offentlige sektor.

Besker, T., Martini, A., & Bosch, J. (2017, September). The pricey bill of technical debt: When and by whom will it be paid?. In 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 13-23). IEEE.

Besker, T., Martini, A., & Bosch, J. (2018a). Managing architectural technical debt: A unified model and systematic literature review. Journal of Systems and Software, 135, 1-16.

Besker, T., Martini, A., & Bosch, J. (2018b, May). Technical debt cripples software developer productivity: a longitudinal study on developers' daily software development work. In Proceedings of the 2018 International Conference on Technical Debt (pp. 105-114).

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

Black, M. (1955). XII.—Metaphor. Proceedings of the Aristotelian Society, 55(1), 273–294. https://doi.org/10.1093/aristotelian/55.1.273

Blaikie, N. (2007). Approaches to social enquiry: Advancing knowledge. Polity.

Blaikie, N., & Priest, J. (2019). Designing social research: The logic of anticipation. John Wiley & Sons.

Blomberg, J. & Burrell, M. (2012). An Ethnographic Approach to Design. In Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications (3rd ed.), J. A. Jacko (ed.). Boca Raton, FL.

Blom-Hansen, J. (2012). Local government in Denmark and the 2007 municipal reform. Rethinking local government: Essays on municipal reform, 43.

Bodker, K., Kensing, F., & Simonsen, J. (2004). Participatory IT design: designing for business and workplace realities. MIT Press.

Bozeman, B., & Bretschneider, S. (1994). The “publicness puzzle” in organization theory: A test of alternative explanations of differences between public and private organizations. Journal of public administration research and theory, 4(2), 197-224.

Brannan, M. & Oultram, T. (2012). Participant Observation. In Qualitative Organizational Research: Core Methods and Current Challenges, Gillian Symon and Catherine Cassell (eds.). Sage Publications, 296–305.

Campbell, J., McDonald, C., & Sethibe, T. (2010). Public and private sector IT governance: Identifying contextual differences. Australasian Journal of Information Systems, 16(2)

Cunningham, W. (1992). The WyCash portfolio management system. ACM SIGPLAN OOPS Messenger, 4(2), 29-30.

Curtis, B., Sappidi, J., & Szynkarski, A. (2012, June). Estimating the size, cost, and types of technical debt. In 2012 Third International Workshop on Managing Technical Debt (MTD) (pp. 49-53). IEEE.

Damgaard, E. (2004). Developments in Danish parliamentary democracy: Accountability, parties and external constraints. Scandinavian Political Studies, 27(2), 115-131.

Danish Ministry of Finance. (2017a). Regeringens kasse eftersyn på it-området. Denmark, Copenhagen

Danish Ministry of Finance. (2017b). Et solidt it-fundament - strategi for it-styring i staten. Denmark, Copenhagen

Danish Ministry of Finance. (2018, June). Regeringens kasseeftersyn på it-området. https://digst.dk/media/20237/regeringens-kasseeftersyn-paa-it_omraadet-2018-_-publikation.pdf

Danish Ministry of Finance. (2021, October 1). Regeringens Digitaliseringspartnerskab offentliggør sine anbefalinger for Danmarks digitale fremtid. Finansministeriet. Retrieved April 6, 2022, from https://fm.dk/nyheder/nyhedsarkiv/2021/oktober/regeringens-digitaliseringspartnerskab-offentliggoer-sine-anbefalinger-for-danmarks-digitale-fremtid/

Danish Ministry of Finance, KL - Local Government Denmark, & Danish Regions. (2016, May). Et stærkt og mere trygt digitalt samfund. Digitaliseringsstyrelsen. https://digst.dk/media/12810/faellesoffentlige-digitaliseringsstrategi-2016-2020-dobbeltopsl.pdf

Davies, M. (1982, January). Idiom and metaphor. In Proceedings of the Aristotelian Society (Vol. 83, pp. 67-85). Aristotelian Society, Wiley.

Dawes, S. S., Vidiasova, L., & Parkhimovich, O. (2016). Planning and designing open government data programs: An ecosystem approach. Government Information Quarterly, 33(1), 15-27.

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. Academy of management journal, 50(1), 25-32.

Ejersbo, N., & Greve, C. (2014). Moderniseringen af den offentlige sektor. Akademisk Forlag.

Ernst, N. A., Bellomo, S., Ozkaya, I., Nord, R. L., & Gorton, I. (2015, August). Measure it? manage it? ignore it? software practitioners and technical debt. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (pp. 50-60).

Fairbanks, G. (2020). Ur-technical debt. IEEE Software, 37(4), 95-98.

Fernández-Sánchez, C., Garbajosa, J., & Yagüe, A. (2015, October). A framework to aid in decision making for technical debt management. In 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD) (pp. 69-76). IEEE.

Flak, L. S., & Rose, J. (2005). Stakeholder governance: Adapting stakeholder theory to e-government. Communications of the Association for Information Systems, 16(1), 31.

Freeman, R. (1984). E.(1984), Strategic management: A stakeholder approach. Boston: Pitman, 46.

Freire, S., Rios, N., Mendonça, M., Falessi, D., Seaman, C., Izurieta, C., & Spínola, R. O. (2020, March). Actions and impediments for technical debt prevention: results from a global family of industrial surveys. In Proceedings of the 35th Annual ACM Symposium on Applied Computing (pp. 1548-1555).

Ghanbari, H., Besker, T., Martini, A., & Bosch, J. (2017, November). Looking for peace of mind? manage your (technical) debt: An exploratory field study. In 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 384-393). IEEE.

Gliedman, C. (2004). Defining IT Portfolio Management. Forrester. Best Practices.

Glucksberg, S. (1989). Metaphors in conversation: How are they understood? Why are they used?. Metaphor and Symbol, 4(3), 125-143.

Gregor, S. (2006). The nature of theory in information systems. MIS quarterly, 611-642.

Grönlund, Å. (2004, August). State of the art in e-Gov research–a survey. In International Conference on Electronic Government (pp. 178-185). Springer, Berlin, Heidelberg.

Guo, Y., & Seaman, C. (2011, May). A portfolio approach to technical debt management. In Proceedings of the 2nd Workshop on Managing Technical Debt (pp. 31-34).

Heeks, R., & Bailur, S. (2007). Analyzing e-government research: Perspectives, philosophies, theories, methods, and practice. Government information quarterly, 24(2), 243-265.

Hekkala, R., Stein, M. K., & Rossi, M. (2018). Metaphors in managerial and employee sensemaking in an information systems project. Information Systems Journal, 28(1), 142-174.

Henriksen, H. Z. (2015, August). Scrutinizing Open Government Data to Understand Patterns in eGovernment Uptake. In International Conference on Electronic Government (pp. 144-155). Springer, Cham.

Høybye-Mortensen, M. (2011). Teknologi i offentlige organisationer. Organiseringen af den offentlige sektor-Grundbog i offentlig forvaltning, 377-397.

IT-Branchen. (2021, August 12). Hvad siger loven om It-udbud? Retrieved April 6, 2022, from https://itb.dk/raadgivning/it-udbud/hvad-siger-loven-om-it-udbud/

Jacobsen, B., Tanggaard, L., & Brinkmann, S. (2020). Fænomenologi. In Kvalitative metoder: En grundbog (pp. 281-308). Hans Reitzels Forlag.

Jensen, H. S. J. (2021, September 24). Digitaliseringen af Danmark, efter 1968. danmarkshistorien.dk. Retrieved February 25, 2022, from https://danmarkshistorien.dk/vis/materiale/digitaliseringen-af-danmark-efter-1968/

Jæger, B. (2020). Digitalisering af den offentlige sektor i et historisk perspektiv. In Digitalisering i offentlig forvaltning (pp. 51-74). Samfundslitteratur.

Jæger, B., & Löfgren, K. (2010). The history of the future: changes in Danish e-government strategies 1994–2010. Information Polity, 15(4), 253-269.

Kellerman, I., & Löfgren, P. (2008). Application Portfolio Management A Framework for Application Destiny Determination. IT University of Göteborg, Göteborg, Sweden.

Kendall, J. E., & Kendall, K. E. (1993). Metaphors and methodologies: Living beyond the systems machine. MIS quarterly, 149-171.

Klinger, T., Tarr, P., Wagstrom, P., & Williams, C. (2011, May). An enterprise perspective on technical debt. In Proceedings of the 2nd Workshop on managing technical debt (pp. 35-38).

Kruchten, P., Nord, R. L., Ozkaya, I., & Falessi, D. (2013). Technical debt: towards a crisper definition report on the 4th international workshop on managing technical debt. ACM SIGSOFT Software Engineering Notes, 38(5), 51-54.

Kumar, R., Ajjan, H., & Niu, Y. (2008). Information technology portfolio management: literature review, framework, and research issues. Information Resources Management Journal (IRMJ), 21(3), 64-87.

Kvale, S. 2008. Doing Interviews. Sage Publications.

Lakoff, G., & Johnson, M. (2008). Metaphors we live by. University of Chicago press.

Lemmetti, J. (2016, September). Construction of Enterprise Architecture in Discourses Within the Public Sector. In International Conference on Electronic Government (pp. 287-298). Springer, Cham.

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.

Lindgren, I., Melin, U., & Sæbø, Ø. (2021). What is E-Government? Introducing a Work System Framework for Understanding E-Government. Communications of the Association for Information Systems, 48(1), 43.

Magnusson, J., Juiz, C., Gómez, B., & Bermejo, B. (2018, May). Governing technology debt: beyond technical debt. In Proceedings of the 2018 International Conference on technical debt (pp. 76-84).

Mars, M. M., Bronstein, J. L., & Lusch, R. F. (2012). The value of a metaphor: Organizations and ecosystems. Organizational Dynamics, 41(4), 271-280.

Mathiassen, L. (2017). Designing engaged scholarship: From real-world problems to research publications. Engaged Management Review, 1(1), 2.

Matthiesen, S., & Bjørn, P. (2015, February). Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (pp. 876-890).

McKeen, J. D., & Smith, H. A. (2010). Developments in Practice XXXIV: Application portfolio management. Communications of the Association for Information Systems, 26(1), 9.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). Qualitative data analysis: A methods sourcebook. Sage publications

Morgan, G. (1998). Images of organization: The executive edition. Thousand Oaks, CA.

Nielsen, M. E., Madsen, C. Ø., & Lungu, M. F. (2020, August). Technical Debt Management: A systematic literature review and research agenda for digital government. In International Conference on Electronic Government (pp. 121-137). Springer, Cham.

Nielsen, M. E., & Skaarup, S. (2021, October). IT Portfolio management as a framework for managing Technical Debt: Theoretical framework applied on a case study. In 14th International Conference on Theory and Practice of Electronic Governance (pp. 89-96).

Oates, B. J., & Fitzgerald, B. (2007). Multi-metaphor method: organizational metaphors in information systems development. Information Systems Journal, 17(4), 421-449.

Ogden, C. K., & Richards, I. A. (1923). The meaning of meaning: A study of the influence of thought and of the science of symbolism.

Ortony, A. (1975). Why metaphors are necessary and not just nice. Educational theory, 25(1).

Peppard, J. (2003). Managing IT as a Portfolio of Services. European Management Journal, 21(4), 467-483.

Pérez, B., Castellanos, C., Correal, D., Rios, N., Freire, S., Spínola, R., ... & Izurieta, C. (2021). Technical debt payment and prevention through the lenses of software architects. Information and Software Technology, 140, 106692.

Pfeiffer, H. 2021. IDENTIFYING CRITICAL PROJECTS VIA PAGERANK AND TRUCK FACTOR. IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). IEEE, (pp. 41-45)

Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., ... & Spinola, R. (2022). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. Journal of Systems and Software, 184, 111114.

Ribeiro, L. F., de Freitas Farias, M. A., Mendonça, M. G., & Spínola, R. O. (2016, April). Decision Criteria for the Payment of Technical Debt in Software Projects: A Systematic Mapping Study. In ICEIS (1) (pp. 572-579).

Rios, N., de Mendonça Neto, M. G., & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. Information and Software Technology, 102, 117-145.

Rios, N., Spínola, R. O., Mendonça, M., & Seaman, C. (2020). The practitioners' point of view on the concept of technical debt and its causes and consequences: a design for a global family of industrial surveys and its first results from Brazil. Empirical Software Engineering, 25(5), 3216-3287.

Rose, J., Flak, L. S., & Sæbø, Ø. (2018). Stakeholder theory for the E-government context: Framing a value-oriented normative core. Government Information Quarterly, 35(3), 362-374.

Rose, J., Persson, J. S., & Heeager, L. T. (2015). How e-Government managers prioritise rival value positions: The efficiency imperative. Information polity, 20(1), 35-59.

Roy, V., & Aubert, B. A. (2002). A resource-based analysis of IT sourcing. ACM SIGMIS Database: the DATABASE for Advances in Information Systems, 33(2), 29-40.

Scale Agile Inc. 2021. PI Planning. Retrieved April 19, 2021 from https://www.scaledagileframework.com/

Scholl, H. J. (2001). Applying stakeholder theory to e-government. In Towards the E-society (pp. 735-747). Springer, Boston, MA.

Scholl, H. J. (2006a, September). Is e-government research a flash in the pan or here for the long shot?. In International Conference on Electronic Government (pp. 13-24). Springer, Berlin, Heidelberg.

Scholl, H. J. (2006b). Electronic government: Information management capacity, organizational capabilities, and the sourcing mix. Government Information Quarterly, 23(1), 73-96.

Scholl, H. J. (2007). Central research questions in e-government, or which trajectory should the study domain take?. Transforming Government: people, Process and policy.

Scholl, H. J. (2014, September). The EGOV research community: An update on where we stand. In International Conference on Electronic Government (pp. 1-16). Springer, Berlin, Heidelberg.

Scholl, H. J. (2015a). Electronic government: Introduction to the domain. In e-Government: Information, technology, and transformation (pp. 19-26). Routledge.

Scholl, H. J. (2015b). Electronic government: A study past its infancy. In e-Government: Information, technology, and transformation (pp. 19-26). Routledge.

Scholl, H. J. (2020). Digital government: Looking back and ahead on a fascinating domain of research and practice. Digital Government: Research and Practice, 1(1), 1-12.

Scholl, H. J. (2022). Digital Government Research: A Diverse Domain. In Scientific Foundations of Digital Governance and Transformation (pp. 51-71). Springer, Cham.

Schou, J., & Hjelholt, M. (2019). Digitaliseringen af den danske offentlige sektor: hvor er vi på vej hen?. Økonomi & Politik, 92(2).

Schou, J., & Pors, A. S. (2019). Digital by default? A qualitative study of exclusion in digitalised welfare. Social policy & administration, 53(3), 464-477.

Seaman, C., Guo, Y., Zazworka, N., Shull, F., Izurieta, C., Cai, Y., & Vetrò, A. (2012, June). Using technical debt data in decision making: Potential decision approaches. In 2012 Third International Workshop on Managing Technical Debt (MTD) (pp. 45-48). IEEE.

Skaarup, S. (2020, August). The role of domain-skills in bureaucratic service encounters. In International Conference on Electronic Government (pp. 179-196). Springer, Cham.

Simon, D., Fischbach, K., & Schoder, D. (2010). Application portfolio management--an integrated framework and a software tool evaluation approach. Communications of the Association for Information Systems, 26(1), 3.

Sjøberg, D. I., Yamashita, A., Anda, B. C., Mockus, A., & Dybå, T. (2012). Quantifying the effect of code smells on maintenance effort. IEEE Transactions on Software Engineering, 39(8), 1144-1156.

Statistics Denmark. (2020, November 27). It-anvendelse i befolkningen 2020. Danmarks Statistik. Retrieved April 12, 2022, from https://www.dst.dk/da/Site/Dst/Layouts/Main.aspx?cid=29450

Swedish National Audit Offie. (2019, October 15). Föråldrade it-system – hinder för en effektiv digitalisering (RiR 2019:28). https://www.riksrevisionen.se/. Retrieved October 31, 2021, from https://www.riksrevisionen.se/rapporter/granskningsrapporter/2019/foraldrade-it-system---hinder-for-en-effektiv-digitalisering.html

Sæbø, Ø., Flak, L. S., & Sein, M. K. (2011). Understanding the dynamics in e-Participation initiatives: Looking through the genre and stakeholder lenses. Government Information Quarterly, 28(3), 416-425.

The Public Accounts Committee & The Auditor General's Office. (2020, September). Gevinstrealisering i statslige it-projekter. The Auditor General's Office. https://rigsrevisionen.dk/Media/6/4/sr2119.pdf

Thibodeau, P. H., & Boroditsky, L. (2011). Metaphors we think with: The role of metaphor in reasoning. PloS one, 6(2), e16782.

Tinholt, D., van der Linden, N., Enzerink, S., Geilleit, R., Groeneveld, A., Cattaneo, G., Aguzzi, S., Pallaro, F., Nocci, G., Benedetti, M., Marchio, G. & Salvadori, A. (2018). eGovernment Benchmark 2018: Securing eGovernment for all.

Twizeyimana, J. D., & Andersson, A. (2019). The public value of E-Government–A literature review. Government information quarterly, 36(2), 167-178.

UNDESA. (2017, June 21). World Population Prospects: The 2017 Revision | Multimedia Library - United Nations Department of Economic and Social Affairs. Https://Www.Un.Org/En/Desa/Products/Publications. Retrieved October 31, 2021, from https://www.un.org/development/desa/publications/world-population-prospects-the-2017-revision.html

Van de Ven, A. H. (2007). Engaged scholarship: A guide for organizational and social research. Oxford University Press on Demand.

Vassallo, C., Palomba, F., Bacchelli, A., & Gall, H. C. (2018, September). Continuous code quality: are we (really) doing that?. In Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (pp. 790-795).

VERBI Software. (2019). MAXQDA 2020 [computer software]. Berlin, Germany: VERBI Software. Available from maxqda.com.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, xiii-xxiii.

Whetten, D. A. (1989). What constitutes a theoretical contribution? Academy of management review, 14(4), 490-495.

Yin, R. K. (2018). Case Study Research and Applications: Design and Methods (6th ed.). SAGE Publications, Inc.

Yildiz, M. (2007). E-government research: Reviewing the literature, limitations, and ways forward. Government information quarterly, 24(3), 646-665.

Appendix

This is an article-based Ph.D. dissertation that consists of four articles and a summary (kappa) to the research conducted throughout the Ph.D. The appendix includes articles which are a part of this dissertation.

Article 1: Nielsen, M. E., Madsen, C. Ø., and Lungu, M. F. (2020, August). Technical debt management: A systematic literature review and research agenda for digital government. **Status: Published** in International Conference on Electronic Government (pp. 121-137). Springer, Cham.

Article 2: Nielsen, M. E., Crusoe, J., Melin, U. Exploration of metaphors as a way to understand socio-technical phenomena: An emergent analytical framework. **Status: 2nd round of revisions** at the Journal of Information Polity June 2022

Article 3: Nielsen, M. E., Skaarup, S. (2021, October). IT Portfolio management as a framework for managing technical debt: Theoretical framework applied on a case study. **Status: Published** in Proceedings of the 14th International Conference on Theory and Practice of Electronic Governance (ICEGOV 2021). ACM Press; New York.

Article 4: Nielsen, M. E., Madsen, C., Ø (2022). Stakeholder influence on technical debt management in the public sector: An embedded case study. **Status: Published** at Government Information Quarterly.

Technical Debt Management: A systematic literature review and research agenda for digital government

Mille Edith Nielsen[1, 2], Christian Østergaard Madsen[1, 2], Mircea Filip Lungu[1]

[1] The IT University of Copenhagen, Denmark
[2] Research Centre for Government IT
{miln, chrm, mlun}@itu.dk

Abstract. Technical debt is created when software engineers knowingly or unknowingly introduce shortcuts or unsuitable choices in the development or maintenance of the software system, that will have a negative impact on the future evolution of the system until corrected. Therefore, it is crucial to manage established debt particular in the public sector. The aim of this study is to introduce Technical debt to the field of Digital Government. We create an overview of the state of the art of the knowledge on technical debt management, the methods applied to gain this knowledge, and propose a research agenda to Digital Government scholars. We conduct a systematic literature review, which focuses on the concept of technical debt management. Forty-nine papers published within 2017-2020 are selected and analyzed. We identify several gaps in the existing literature: 1) an absence of theory explaining the relation of events, 2) a shortage of studies conducted in the public sector, 3) and an absence of specific techniques such as observation to study actual technical debt management behavior.

Keywords: Technical debt management, managing legacy systems, systematic literature review, technical debt

1 Introduction

Public and private organizations use information technology (IT) to improve their efficiency and service offerings [1]. During the development of IT-projects and the subsequent operation/maintenance, temporary suboptimal solutions are sometimes introduced to profit from the solutions faster. To capture this reality, Cunningham [2] coined the term technical debt (TD) to explain the process and pitfalls of programming to the management in the banking sector in 1992. Rios et al [3, p. 117] describe TD as a conceptualization of *"problems faced during software evolution considering the tasks that are not carried out adequately during software development."* They conducted a tertiary literature review and found a variation in the application of the term. TD is often associated with any impediment related to the software product and the development process [3]. Griffith et al describe technical debt management (TDM) as comprising "*the actions of identification, assessment, and remediation of technical debt throughout a software system*." [4, p. 1016].

TDM is important because it enables an organization a more optimal use of its resources [5]. For instance: An organization's IT-system breaks down, which leads to reduced production. For management, the possible solutions are to 1) accept the reduced production, 2) conduct a root cause analysis and fix the actual problem, 3) create a work-around that increases momentarily production speed, but does not solve the actual problem and will make the maintenance of the solution harder in the future. Solution 3 may be a viable course of action under certain circumstances, but it will result in the creation of technical debt. Moreover, since the consequences are no longer visible to management, this debt may be forgotten. If proper TDM methods were applied, management could identify the debt and repay it in a timely fashion.

We will claim, TD is important for the field of Digital Government, because TD can hinder the public sector in fully reaping the benefits of digitalization. Scholl encourages the field of Digital Government to engage with other disciplines "*which overlap with Digital Government as a practice area, but which might lack the forward-looking capabilities that Digital Government Research at least can provide in part.*" [7, p. 11]. Moreover, Digital Government scholars can contribute to TD research, because they have both domain specific knowledge of the public sector's use of IT and methodological experience in studying IT and operation in this context [8].

In 2017, The Danish Ministry of Finance published an analysis of Danish public IT-systems. The report concluded that 157 of 428 society or business-critical IT systems' technical components (applications and IT-infrastructure) were not fully maintained [9]. Outdated software or hardware components can increase the risk of breakdowns, security breaches, and trouble the maintenance and the future development of the IT-systems maintained [9]. The Swedish National Audit Office conducted a similar analysis and found 70% of their IT-systems were outdated [10].

TD and the concept of legacy systems "*discuss a state of software that is sub-optimal, time constrained, and explain how this state can decrease an organization's development efficiency*" [8, p. 80]. A considerable source of TD originates from software legacy [12], e.g. during continuous development of a system in an outdated environment. In this review, we focus solely on TDM.

TD studies are primarily published in Software Engineering, especially after Cunningham's [2] introduction of the metaphor [3, 5, 13, 14]. However, Information Systems researchers have also published studies on TD [15-16].

We found nine literature reviews and one tertiary review on TD and TDM (Appendix A). While these reviews offer important contributions, they only cover the literature up until 2017 and do not focus on examining methodology, use of theory or unit of analysis. In this study we aim to address these gaps.

We identify 49 TDM papers published within 2017-2020. We find the focus of the TD research fragmented: TD decreases morale, TD is difficult to measure, and numerous tools. The MTD workshops and a tertiary study encourage more research on strategies and management [P13, 3]. The papers primarily present data from open source projects and the private sector. This leaves a gap for research in the public sector. Finally, we offer a research agenda for Digital Government scholars on TDM.

1.1 Aim of the study and research questions

This study reviews the latest published TDM papers (2017-2020). To the best of our knowledge, these papers have not been studied by other secondary studies. The TD field is rapidly evolving with nine secondary studies being published in the past six years. Our study indicates that the publishing rate has not decreased. In a four-year period, TD papers were published in 30 different outlets. This study introduces TD research and a research agenda to the Digital Government scholars. Therefore, we follow established guidelines for conducting systematic literature reviews within the Digital Government field [17, 18]. Our research questions are:

- RQ1. How is TDM studied and in which fields? RQ1a: Which authors and fields have contributed to technical studies debt management since 2017? RQ1b: What are the methods, context, level of analysis and data level used? RQ1c: Which theories and theoretical concepts are applied?
- RQ2. What does the TDM literature focus on? RQ2a Which topics are studied? RQ2b What do the findings show?
- RQ3. What research agenda should the Digital Government scholars investigate in the context of TDM? RQ3a What suggestions does the literature have for future studies? RQ3b What are the identified knowledge gaps in the literature?

The following sections are organized as follows: 2) search process, 3) brief mapping of previous TD literature reviews, 4) analysis, 5) results and suggestions for future study, 6) discussion, limitations, and finally, 7) conclusion. The previous literature reviews, and the pool of papers are listed in Appendix A and B.

2 Search process

The initial search for papers occurred from January to April 2019, with an updated search conducted in March 2020. We applied Webster and Watson's [18] method for conducting systematic literature reviews. Webster and Watson present a three-step process to search for papers [18].

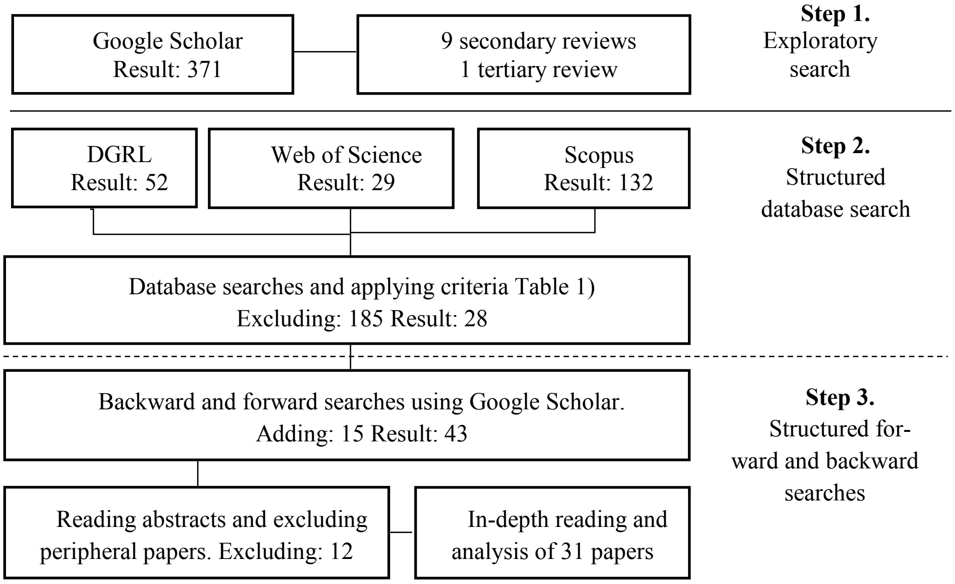

Google Scholar
Result: 371
9 secondary reviews
1 tertiary review
Step 1.
Exploratory search
DGRL
Result: 52
Web of Science
Result: 29
Scopus
Result: 132
Step 2.
Structured database search
Database searches and applying criteria Table 1)
Excluding: 185 Result: 28
Backward and forward searches using Google Scholar.
Adding: 15 Result: 43
Step 3.
Structured forward and backward searches
Reading abstracts and excluding peripheral papers. Excluding: 12
In-depth reading and analysis of 31 papers

Figure 1 illustrates the initial process.

Webster and Watson recommend, that scholars begin a review by searching for papers in known key outlets. The second step is a database search, as it enables the researcher to discover other fields. The third step is a backward-forward search, where papers citing or cited by the pool of papers are identified. The following sections explain each search step in detail.

1. Exploratory search: Using the software tool 'Publish or Perish' (Google Scholar)
2. Database search: Web of Science, DGRL and Scopus
3. Backwards and forward search (Google Scholar citations)

2.1 Explorative search (Step 1)

Webster and Watson [18] recommend starting a review by searching key outlets. They assume that the researcher is familiar with the literature and key outlets. However, we were not familiar with these at the time. Therefore, we chose to deviate from their method in the first step. In January 2019, we conducted an explorative search through Google Scholar to become familiar with the topic and academic literature [19]. We used the software "Publish or Perish" and Google Scholar as an underlying search engine [20]. This search created a foundation for the second step: the structured literature database search. The search informed: the selected search term, the selection criteria, and the temporal limitation. In this search process, we discovered nine secondary literature reviews and a tertiary review (Section 3 & Appendix A). The secondary and tertiary reviews informed our inclusion and exclusions criteria (table 1).

Table 1. Selection criteria (+ included / - excluded)

+/-	Assessment criterion	Argument
+	Topic should be TDM	Only papers focusing on TDM were included
+	Published within the period 2017-2020	The papers posted up until 2017 have been covered by other literature studies [21, 22]
+	Published conference or journal papers only	To ensure high quality, only papers that had undergone a peer-reviewed process were included
+	Only papers written in English	Due to the authors' language skill, we included papers in English only.
-	Books, chapters, blog posts, citations, thesis, presentations, reports, editorials & summaries	The scope of this literature review is to analyze the peer-reviewed papers. To avoid research duplicates books and books chapters were excluded [17].
-	Inaccessible papers	Papers, which appeared in the search but could not be accessed were excluded
-	Duplicates	Only the newest version of the paper was included (typically journal paper). Journal papers are generally more in-depth than conference papers [17].

2.2 Structured database search (step 2)

In February 2019, we searched in the three databases: Digital Government Reference Library (DGRL), Scopus and Web of Science. DGRL version 15.5 contains references to approximately 12.500 peer-reviewed papers within the digital government field.

Webster and Watson [18] suggest using keywords when searching in databases, However, not all papers are presented with keywords in these databases. Therefore, we searched the databases for papers including the expression 'Technical Debt Management' anywhere in the text. We identified 213 primary research papers on TD.

Due to a large number of identified papers, and the existence of secondary and tertiary studies, we decided to introduce two additional selection criteria in our study (table 1). This was done to position our review in relation to the previous secondary and tertiary studies. The secondary studies cover all papers on TD published until early 2017 (see Figure 2). Thus, we focused the review on 1) papers studying TDM specifically, this brought the number of papers to 130, 2) papers published from 2017 and onwards, which reduced the number to 28 papers.

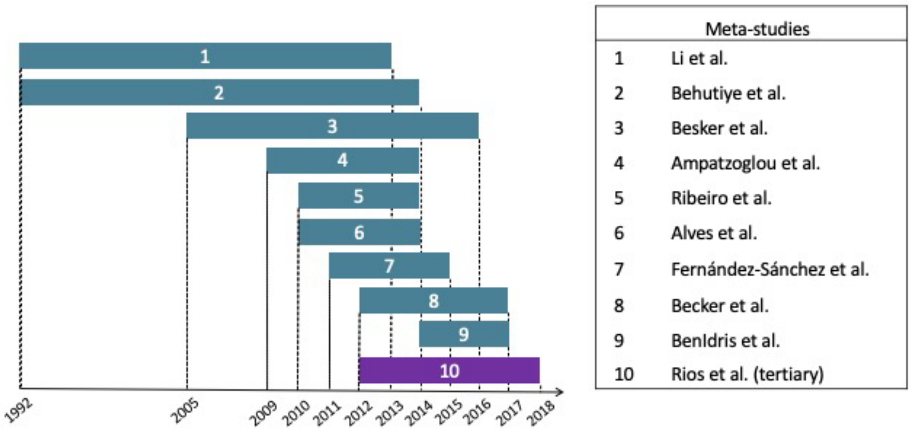
2.3 Forward/backward search (Step 3)

In April 2019, we conducted backward and forward searches [18]. First, we reviewed the references in the previously identified papers (backward search) adding two papers. Second, we used Google Scholar to conduct a forward search for papers citing the identified papers, adding 13 papers. This brought the pool of papers to 31.

In March 2020 we updated the search and repeated step 2 and 3, we identified 18 new papers, bringing the total pool of papers in this review to 49 (Appendix B).

3 Previous secondary studies on TD

In the explorative search, we found nine secondary and one tertiary study. The nine studies cover literature published from 1992-2017 (Figure 2). Benldris et al. [22] and Becker et al. [21] cover the year 2016 completely with a broad view of TD. We position



Meta-studies	
1	Li et al.
2	Behutiye et al.
3	Besker et al.
4	Ampatzoglou et al.
5	Ribeiro et al.
6	Alves et al.
7	Fernández-Sánchez et al.
8	Becker et al.
9	Benldris et al.
10	Rios et al. (tertiary)

Figure 2 illustrates previous meta-studies research on TD

our literature review to these existing secondary studies in the discussion section. Rios et al. published a tertiary study in 2018, which explores the state of TD research [3]. They evaluate 13 secondary studies from 2012 to March 2018. They identify three research directions and concepts studied in the secondary studies: TD identification, TD concepts, and TDM. They develop a taxonomy of 15 TD types and generate a TD landscape mapping out the TDM activities, strategies and tools. The period from 2017 does not appear to be covered and none of the literature reviews examine the context of the data.

4 Method for analysis

The total pool of papers in our review includes 49 papers. Next, we explain the overall coding process, then we go more into detail in the following section.

The coding process was conducted iteratively. We developed a coding sheet (a template) containing: origin of data, theory and method, and unit of analysis [23]. We applied the template on 10 papers and discussed and adjusted the categories. The first author conducted the coding, while coding issues were discussed among all authors.

4.1 Detailed description of coding elements

This section describes additional categories besides author and journal. We explain the reason for our coding, and the coding process in detail.

The origin of the data. We coded for country and sector (private or public) to uncover where the data originates from. We expanded the categorization to include open source projects. Open source projects allow for anyone, anywhere to contribute to the code without stating who they are or where they are from. In the papers, the authors describe where they extracted the data, e.g. by state the type of organization, from which we could interpret the sector.

Theory and technique. We made an open coding for the technique; thus, we included a description of how the data was extracted. We coded for theory, focusing on explanations of the relation between concepts, observed phenomenon and why these relationships exist [24–26].

Findings and future studies. We coded the papers' main findings, typically from the results, discussion, and conclusion sections to identify the latest findings within TDM studies. Additionally, we coded for the authors' suggestions for future studies.

Unit of analysis and concepts. Webster and Watson [18] suggest creating a concept matrix and adding another dimension: the level of analysis. This dimension analyzes the abstraction level of the paper. This allows for more accurate identification of the existing literature. They suggest three levels of analysis: individual, group and organization. We discovered several papers that analyzed at the IT-system level and added this to the existing three levels of analysis. We coded for the overall concept within the paper, besides TDM. We began with a careful read-through of the first 10 papers. Here, a pattern emerged, and we identified four concepts. They were confirmed after going through the remainder of the pool of papers.

5 Results

Next, we present the findings of our study according to our research questions.

5.1 RQ1 How is TDM studied and in which fields?

RQ1a: Which authors and disciplines have contributed to TDM research since 2017? Papers concerning TDM have been published in 30 different journals and conferences since 2017. The eight outlets containing more than one paper are listed in table 2, the remaining 23 papers are published in 26 different outlets.

The International Conference on TD contains the most papers regarding TDM followed by the Journal of Systems and software. The mapping shows that TDM appears in a variety of outlets within the field of Software Engineering. The 65% of the papers do not specify the countries where the study was conducted. The papers which do specify this, primarily come from the Nordic countries - particularly Sweden [P2, P4, P11, P18, P21, P45]. The papers not specifying the country mainly use open source projects.

Table 2. The most frequent publishing outlets for TDM research

Outlet	No. of papers
International Conference on Technical Debt	7
Journal of Systems and Software	4
ACM SIGSOFT Software Engineering Notes	2
ACM SIGSOFT Software Engineering	2
IEEE Transactions on Software Engineering	2
International Conference on Software Engineering	2
Euromicro Conference on Software Engineering and Advanced Applications	2
International Conference on Software Maintenance and Evolution	2

124 different authors have contributed to the literature of TDM, 15 of them contributed to two papers. Figure 3 presents the top contributors, note that many papers have several authors.

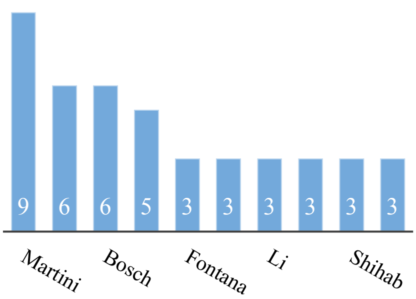


Fig. 3. The most active authors

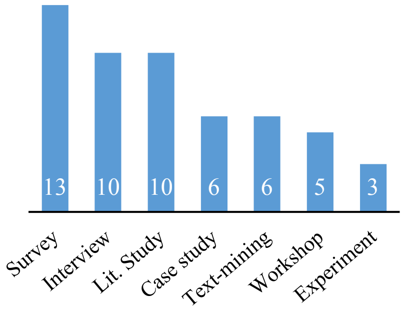


Fig. 4. The most frequently used techniques

RQ1b: What are the methods, context, level of analysis and data level used? Only half of the papers specify the organizational context, which the data originates from. One paper use data from the public sector, 12 papers present open source data, and 12 papers present data from the private sector. The papers use different techniques during their research. The most frequently used techniques are survey, interviews and literature studies (figure 4). Note that some studies use more than one technique.

RQ1c: Which theories and theoretical concepts are applied? The authors introduced different concepts and frameworks to explain TD and the underlying interrelations. This was done from different perspectives; assessment, working environment or awareness. The papers seldomly use theory and the word 'theory' rarely occurs in the text of the papers. Two papers are presented as foundation for future theory [P39, P45]. The papers offer indicators and methods on how to reduce TD.

5.2 RQ2: What does the TDM literature focus on?

RQ2a: Which topics are studied? Inspired by Webster and Watson [18], we have identified four main concepts and mapped the papers' level of analysis into a concept matrix table 3). This gives an overview of how the concepts have been studied and which levels or concepts have not received attention in recent studies.

The concept *TD Assessment* entails research which assesses the effect of TD or as it is called, *interest rate*. This concept is the most analyzed and it has been analyzed at four different levels. Ten of the papers use the concept *Self-admitted technical debt*, which covers TD that is consciously admitted and is visible in code comments. This has been analyzed at the IT-System and Project level. The third concept of *TD awareness* covers the research creating awareness of TD. This can be achieved by visualizing the assessed TD; it is analyzed at the IT-system and the project level. The last concept *Working environment* focuses on morale and organizational culture. This is analyzed at both an individual and an organizational level.

Table 3. Concept matrix illustrating the studied topics

Level of analysis	Individual employee	IT-System	Project	Organization
Self-admitted TD		P28, P19, P31, P17, P30, P27	P12, P23, P8, P41	
TD assessment	P4, P10, P33, P44	P20, P26, P5, P14, P31, P27, P37	P16, P22, P6, P14, P34, P38	P5, P15, P9, P21, P35, P39
TD awareness	P45, P46, P49	P24, P26, P19, P28, P17, P42, P43, P48, P37	P16, P22, P6, P47	
Working environment	P1, P2, P3, P11			P1, P18, P46

A third of the papers analyze TD at the IT-System level, the most frequent analyzed level. The primary focus is TD assessment, which is explored in 21 papers. The organizational level is the least used level of analysis; one explanation is that organizational data is more challenging to access than surveys and on open source projects.

RQ2b: What do the findings show? Almost a third of the papers propose a tool, method, model or technique to aid TDM [P8, P16-20, P29-31, P34, P35, P41, P42, P47, P48]. Their findings focus on the results of evaluating their presented tools. The tools generally aid in TDM e.g. through TD identification or visualization. Increased TD visibility can benefit communication between stakeholders [P37].

The papers vary in approach and focus; thus, the findings of the papers are fragmented. We list seven general findings from the pool of papers: **1)** TD harms software development work [P2-4, P11]. **2)** All roles related to the system are affected [P4], and community-related factors contribute to TD's intensity [P46], however, **3)** the developer morale can be increased by proper management [P11]. **4)** Organizational factors can influence TD [P1], the number of collaborators and the size of the project correlate significant with the amount of TD [P14], and the breadth of the developers' experience lower the amount of TD [P1, P33]. **5)** TD is time consuming: practitioners estimate 36% of development time is wasted due to TD [P4], and TD increases the need to perform additional time-consuming activities [P3]. However, TD is not visible in the backlog [P45] and lack of development processes increase TD [P1] **6)** Architectural debt should be managed early in the process because the early introduction of architectural debt shows it persists during the whole software lifecycle [P2]. **7)** The estimation difficulties are proposed to be solved by a workflow, which "*provide more actual information including TD concepts to the stakeholders*" [P7, p. 600-601].

5.3 RQ3: What research agenda should the Digital Government scholars investigate in the context of TDM?

RQ3a What suggestions does the literature have for future studies? The papers proposing a model, tool, method, approach or technique for TD aim to continue expanding and validating their model against new datasets [P1, P5, P8, P12, P14-15, P18-19, P23, P27, P29-31, P34-35, P41-42, P47-48]. Suggestions for future research for other researchers are scarce. However, both MTD workshops encourage a slight change in direction and provide several suggestions for future studies [P10, P13].

Quantifying the value of TD. The report of MTD workshop of 2016 [P13] suggests the research agenda is on defining, understanding and operationalizing the value of TD. They urge researchers to understand the value that falls outside the core definition of TD, which is essential to how TD plays out in practice. Digital Government scholars can contribute to this with their experience. The 2017 report of MTD workshop [P10] recommends elevating the quantification of TD from low-level code to architectural opportunities. They identify a need to educate stakeholders to raise awareness level.

A better understanding of the metrics. Three papers contain the following suggestions for future work: 1) to research more important metrics in the future [P16], 2) to understand the factors leading to TD [P14], 3) to study more change features that can introduce TD [P31]. Other research should be undertaken to investigate if other types of TD (besides Architectural Debt) have a significant correlation to the estimated wasted time. Thus, it creates a better understanding of the negative impact different TD has on wasted software development time [P2-3]. Lastly, the organization's maneuverability can be increased by determining the types of debt incurred [P18].

Strategies. Specific suggestions are to investigate concrete architecture problems: how they contribute to file bug-proneness, and possible ways of refactoring [P6]. Vadja et al. encourage future studies to develop methods to assist the stakeholders to estimate their TD [P22] or increase the breadth of the experience [P33]. Additionally, Dong et al suggest exploration of TD recovery strategies due to a lack of discussed actual cases – particularly in the cross-disciplinary environment [P9]. Further research should focus on providing an in-depth understanding of the relationship between TD and developers' morale [P11]. Two papers aim to build a TD theory [P39, P45].

RQ3b What are the identified knowledge gaps in the literature? We identify five gaps which are presented in Table 4. 1) The papers suffer from a lack of theory and therefore cannot explain the relationship between events and concepts [26]. The debt metaphor contains some explanatory power; however, a metaphor cannot substitute for a theory [27]. Two papers present research as possible foundation for theory [P39, P45]. 2) The level of analysis is primarily focused on a project or system level, which leaves a gap in researching the organizational and individual levels. 3) Half of the papers have not specified the organizational context of the data, and the other half investigate open source projects or private companies. Only one paper use data from the public sector, this leaves a gap in research in this sector [P41]. 4) TD is explored through 16 different approaches and techniques - both quantitative and qualitative. Eight papers use more than one technique and combine both quantitative and qualitative methods. However, observation is not used actively as a data gathering technique.

Table 4. Identified gaps, their importance, and suggestions to how they can be addressed

Gap	Why is this a problem	Suggestion to how the gaps can be addressed
1. Lack of research from the public sector	The public sector has other restrictions than open source and the private sector. The TD challenges and management might differ.	Study TD in the public sector.
2. Lack of Theory	Understand the connection of events and why they occur will help in determining how to prevent them from happening [29].	Develop a new theory or import theory from other fields.
3. Lack of venue diversity	TD is relevant to other fields. Different disciplines may offer different perspectives and approaches to TD.	Introduce TD to other fields than Software Engineering
4. Limited abstraction level of analysis	Examining TD from different levels generates a fuller picture. TD's impact goes beyond the IT-system.	Examine TD from an organizational or individual level.
5. Lack of technique diversity	Observations may inform us about people's actual behaviour, rather than how people claim they act [3].	Use observation to address people's behaviour.

This may be a problem, because what people say during interviews, may differ from what they do [28]. 5) All the papers are published within the field of software engineering leaving a gap for venue diversity. Table 4 summarizes the identified gaps in the papers and suggests why and how these gaps could be addressed.

When we compare the TD literature's suggested research RQ3a and the gaps we identify, we find that they can easily be combined. Our gaps contain a high abstraction level, whereas the suggested future research provides very specific suggestions.

6 Discussion

First, we compare the findings of this review to comparable secondary studies. Second, we identify and discuss five findings further, 1) lack of research in the public sector, 2) empirical confirmation of negative effects of TD, 3) a tendency of reinventing the wheel, 4) gaps in diversity of approaches, and 5) the absence of theory.

Comparison to related work. Becker et al. [21] focus on decision-making and criticize the method and objective of the research. They find that the actual decision making was not studied. Ampatzoglou et al. [13] explore how financial aspects are defined and applied when studying TD. They encourage a balance between economic theories and software engineering. Li et al [14] find that code-related TD and its management gain the most attention and encourage future studies to explore the whole TDM process. This corresponds to our findings in the concept matrix, as research on the organizational level or the working environment is limited. The tertiary review by Rios et al. [3] underline several gaps, which are in line with the MTD reports [P13, 3]: more research on strategies and management.

This literature review includes management of all types of TD, in line with several of the secondary studies [5, 13, 14, 21, 30]. However, none of the secondary studies analyze theory nor the organization type the data was extracted from. Furthermore, this review offers insights into the recent literature and present the concept and a research agenda to Digital Government scholars.

Lack of research in the public sector. Half of the papers do not state the organizational context where data was collected. Only one paper appears to have conducted its data collection within the public sector [P41]. This is important, because the public sector is different from open source projects and the private sector. While some challenges may be similar, public organizations are subjected to specific requirements [32]. We suggest future Digital Government studies focus on studying TDM, both in terms of strategies, management and indicators.

A tendency to reinvent the wheel. About a third of the papers propose a tool to support technical debt management [P8, P16-20, P29-31, P34, P35, P41, P42, P47, P48]. They evaluate their presented tool and plan to apply it within a different context. They rarely apply other researchers' tools or methods, instead, they develop their own. This is problematic as it does not advance the field through joint effort; instead of providing and testing a few select tools for practitioners, the number of tools becomes overwhelming.

Empirical confirmations of the negative effects of TD. Technical debt harms software development work, both in terms of morale and wasted time [P2, P3, P4, P11].

However, community factors can also intensify TD [P46]. The size of the project correlates with the amount of TD [P14]. TD can be minimized by following some simple guidelines. However, if TD is addressed early it is less time consuming and the morale can be increased [P2, P3, P4, P11]. Thus, we need future studies to explore indicators of TD.

Gaps in the diversity of approaches. TDM is primarily studied from a quantitative approach, 30% of the papers used a qualitative approach. Six of the 31 papers use more than one method. This leaves a gap which can be addressed with the mixed method approach. Mixed methods may provide more comprehensive evidence [24]. None of the papers apply observation, this leaves people's actual behavior concerning TDM unstudied.

Absence of theory. Concerning results, we see an absence of the use of theory, a few papers import theory and approaches from other fields, e.g. portfolio finance and games. We need theories to explain the relationship between the concepts and why these exist [24–26]. The absence of native theory is natural for a young field, where researchers are prone to import theory from a different field instead [33, 34].

7 Conclusion, limitations and future studies

We have introduced TDM to the field of Digital Government and proposed a research agenda for Digital Government scholars. We conducted a systematic literature review to explore: 1) how and where technical debt management is studied, 2) what we know about technical debt and its management and lastly 3) a research agenda for scholars within Digital Government. Our findings are based on papers published from 2017-2020. We have discovered that 1) researchers focus on a specific type of technical debt and how it can be reduced. 2) TDM is still strongly rooted within the field of Software Engineering and 3) is primarily examined in open software projects or the private sector. 4) TDM is studied using primarily quantitative methods, finally, 5) there is a lack of theory to guide the studies and explain findings.

Limitations. We decided to only use the term "Technical debt management" in our literature search which can decrease the external validity, however the backward/forward searches strengthen the external validity. The papers were coded by the first author only which decrease the internal validity, in order to strengthen the internal validity, issues were discussed among all authors.

Research agenda. We suggest Digital Government scholars research technical debt management, TD strategies and TD indicators, so the field is advanced further.

8 Acknowledgement

This study is funded by the Research Centre for Government IT, which is a collaboration between the IT University of Copenhagen, the Danish Digitization Agency and the self-owned institution ATP. The Research Centre's external funding partners were not involved in the research presented herein, or its dissemination.

Appendix A: Previous TD literature reviews

Authors	Title	Year	No. of papers	Period covered
Alves, Mendes, De Mendonça, Spinola, Shull, Seaman	Identification and management of technical debt: A systematic mapping study	2016	100	2010-2014
Ampatzoglou, Ampatzoglou, Chatzigeorgiou, Avgeriou	The financial aspect of managing technical debt: A systematic literature review	2015	69	2009-2014
Behutiye, Rodríguez, Oivo, Tosun	Analyzing the concept of technical debt in the context of agile software development: A systematic literature review	2017	38	1992-2014
Ribeiro, de F. Farias, Mendonça, Spínola	Decision Criteria for the Payment of Technical Debt in Software Projects: A Systematic Mapping Study.	2016	38	2010-2014
Becker, Chitchyan, Betz, McCord	Trade-off decisions across time in Technical Debt Management: a systematic literature review	2018	240	2012-2016
Li, Avgeriou, Liang	A systematic mapping study on technical debt and its management	2015	94	1992-2013
Fernández-Sánchez, Garbajosa, Yagüe	A framework to aid in decision making for Technical Debt Management	2015	51	2011-2015
Besker, Martini, Bosch	Managing architectural technical debt: A unified model and systematic literature review	2018	42	2005-2016
BenIdris, Ammar, Dzielski	Investigate, identify and estimate the technical debt: A systematic mapping study	2018	43	2014-2017

Appendix B: The pool of papers analyzed in the literature review

ID	Author	Title	Year
P1	Besker & Martini	Embracing Technical Debt, from a Startup Company Perspective	2018
P2	Besker, Martini & Bosch	Impact of Architectural Technical Debt on Daily Software Development Work — A Survey of Software Practitioners	2017
P3	Besker, Martini & Bosch	Technical debt cripples software developer productivity: a longitudinal study on developers' daily software development work	2018
P4	Besker, Martini &Bosch	The Pricey Bill of Technical Debt: When and by Whom will it be Paid?	2017
P5	Bowlds, Fossaceca & Iammartino	Software obsolescence risk assessment approach using multicriteria decision-making	2018
P6	Cai, Xiao, Kazman, Mo, Feng,	Design Rule Spaces: A New Model for Representing and Analyzing Software Architecture	2018
P7	Cha, Dong & Vogel-Heuser	Preventing Technical Debt For Automated Production System Maintenance Using Systematic Change Effort Estimation With Considering Contingent Cost	2018
P8	Chicote	Startups and Technical Debt: managing technical debt with visual thinking	2017
P9	Dong, Ocker & Vogel-Heuser	Technical Debt as indicator for weaknesses in engineering of automated production systems	2019
P10	Fontana, Chatzigeorgiou, Trumler, Izurieta, Avgeriou & Nord	Technical Debt in Agile Development: Report on the Ninth Workshop on Managing Technical Debt (MTD 2017)	2017
P11	Ghanbari, Besker, Martini, & Bosch	Looking for peace of mind?: manage your (technical) debt: an exploratory field study	2017
P12	Huang, Shihab, Xia, Lo & Li	Identifying self-admitted technical debt in open source projects using text mining	2018
P13	Izurieta, Ozkaya, Seaman & Snipes	Technical Debt: A Research Roadmap Report on the Eighth Workshop on Managing Technical Debt (MTD 2016)	2017

P14	Jesus & Melo	Technical Debt and the Software Project Characteristics. A Repository-Based Exploratory Analysis	2017
P15	Klotins, Unterkalmsteiner, Chatzipetrou, Gorschek, Prikladnicki & Tripathi	Exploration of technical debt in start-ups	2018
P16	Kosti, Ampatzoglou, Chatzigeorgiou, Pallas, Stamelos & Angelis	Technical Debt Principal Assessment Through Structural Metrics	2017
P17	Liu, Huang, Xia, Shibab, Lo & Li	SATD detector: a text-mining-based self-admitted technical debt detection tool	2018
P18	Magnusson, Juiz, Gómez & Bermejo	Governing technology debt: beyond technical debt	2018
P19	Maldonado, Shibab & Tsantalis	Using Natural Language Processing to Automatically Detect Self-Admitted Technical Debt	2017
P20	Martini	Anacondebt: a tool to assess and track technical debt	2018
P21	Martini, Fontana, Biaggi & Roveda	Identifying and Prioritizing Architectural Debt Through Architectural Smells: A Case Study in a Large Software Company	2018
P22	Martini, Vajda, Vasa, Jones, Abdelrazek & Grundy	Technical debt interest assessment: from issues to project	2017
P23	Mensah, Keung, Svajlenko, Bennin & Mi	On the value of a prioritization scheme for resolving Self-admitted technical debt	2018
P24	Mera-Gómez, Ramírez, Bahsoon & Buyya	A Debt-Aware Learning Approach for Resource Adaptations in Cloud Elasticity Management	2017
P25	Mera-Gómez, Ramírez, Bahsoon & Buyya	A Multi-Agent Elasticity Management Based on Multi-Tenant Debt Exchanges	2018
P26	Rojas, Izurieta & Griffith	Toward Technical Debt Aware Software Modeling	2017
P27	Sierra	Is Self-Admitted Technical Debt a Good Indicator of Architectural Divergences?	2019
P28	Sierra, Shihab, Yasutake & Kamei	A survey of self-admitted technical debt	2019
P29	Skourletopoulos, Mavromoustakis, Mastorakis, Sahalos, Batalla & Dobre	A game theoretic formulation of the technical debt management problem in cloud systems	2017
P30	Wattanakriengkrai, Maipradit, Hata, Choetkiertikul, Sunetnanta & Matsumoto	Identifying Design and Requirement Self-Admitted Technical Debt Using N-gram IDF	2018
P31	Yan, Xia, Shihab, Lo, Yin & Yang	Automating Change-level Self-admitted Technical Debt Determination	2018
P32	Hacks, Hofert, Salentin, Yeong, Lichter.	Towards the definition of enterprise architecture debts	2019
P33	Fagerholm, Becker, Chatzigeorgiou, Betz, Duboc, Penzenstadler, Mohanani Venters	Temporal Discounting in Software Engineering: A Replication Study	2019
P34	Reboucas De Almeida, Treude, Kulesza	Tracy: A Business-Driven Technical Debt Prioritization Framework	2019
P35	Reboucas De Almeida	Business-Driven Technical Debt Prioritization	2019
P36	Njima, Demeyer	Value-based technical debt management: An exploratory case study in start-ups and scale-ups	2019
P37	Rindell, Bernsmed, Jaatun	Managing security in software or: How I learned to stop worrying and manage the security technical debt	2019
P38	Rindell, Holvitie	Security risk assessment and management as technical debt	2019
P39	Becker, Fagerholm, Mohanani, Chatzigeorgiou	Temporal discounting in technical debt: How do software practitioners discount the future?	2019

P40	Brenner R.	Balancing resources and load: Eleven nontechnical phenomena that contribute to formation or persistence of technical debt	2019
P41	Perez, Correal, Astudillo	A proposed model-driven approach to manage architectural technical debt life cycle	2019
P42	Pavlič, Hliš	The technical debt management tools comparison	2019
P43	Aragão, Andrade, Santos, Castro, Lelli, Darin	TestDCat: Catalog of Test Debt Subtypes and Management Activities	2019
P44	Amanatidis, Mittas, Chatzigeorgiou, Ampatzoglou, Angelis	The Developer's Dilemma: Factors affecting the Decision to Repay Code Debt	2018
P45	Besker, Martini, Bosch	Technical Debt Triage in Backlog Management	2019
P46	Palomba, Tamburri, Fontana, Oliveto, Zaidman, Serebrenik,	Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells?	2018
P47	Li, Liang, Avgeriou	Architectural Technical Debt Identification Based on Architecture Decisions and Change Scenarios	2018
P48	Dai, Kruchten	Detecting Technical Debt through Issue Trackers	2017
P49	Rios, Spínola, Mendonça Seaman	Supporting Analysis of Technical Debt Causes and Effects with Cross-Company Probabilistic Cause-Effect Diagrams	2019

References

1. Cordella, A., Iannacci, F.: Information systems in the public sector: The e-Government enactment framework. J. Strateg. Inf. Syst. 19, 52–66 (2010).
2. Cunningham, W.: The WyCash portfolio management system. ACM SIGPLAN OOPS Messenger. 4, 29–30 (1992).
3. Rios, N., Mendonça Neto, M.G. de, Spínola, R.O.: A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. Inf. Softw. Technol. 102, 117–145 (2018).
4. Griffith, I., Taffahi, H., Izurieta, C., Claudio, D.: A simulation study of practical methods for technical debt management in agile software development. In: Proceedings - Winter Simulation Conference (2015).
5. Alves, N.S.R., Mendes, T.S., De Mendonça, M.G., Spínola, R.O., Shull, F., Seaman, C.: Identification and management of technical debt: A systematic mapping study. Inf. Softw. Technol. 70, 100–121 (2016).
6. Niehaves, B.: Innovation Processes in the Public Sector – New Vistas for an Interdisciplinary Perspective on E-Government Research? Electronic Government. EGOV 2007. (2007).
7. Scholl, H.J.: Digital Government : Looking Back and Ahead on a Fascinating Domain of Research and Practice. Digit. Gov. Res. Pract. 1, 1–12 (2020).
8. Dečman, M., Jukić, T.: Editorial for EJEG Volume 15 Issue 1. 15, 1 (2017).
9. Danish Ministry of Finance: Regeringens kasseeftersyn på it-området. Copenhagen, Denmark (2017).
10. The Swedish National Audit: Föråldrade it-system – Hinder för en effektiv digitalisering, Stockholm (2019).
11. Holvitie, J., Licorish, S.A., Martini, A., Leppänen, V.: Co-existence of the "technical debt" and "software legacy" concepts. CEUR Workshop Proc. 1771, 80–83 (2016).
12. Holvitie, J., Licorish, S.A., Spínola, R.O., Hyrynsalmi, S., MacDonell, S.G., Mendes, T.S., Buchan, J., Leppänen, V.: Technical debt and agile software development practices and processes: An industry practitioner survey. Inf. Softw. Technol. (2018).
13. Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P.: The financial aspect of managing technical debt : A systematic literature review. Inf. Softw. Technol. 64, 52–73 (2015).

14. Li, Z., Avgeriou, P., Liang, P.: A systematic mapping study on technical debt and its management. J. Syst. Softw. 101, 193–220 (2015).
15. Jeffery, M., Leliveld, I.: Best practices in IT portfolio management. MIT Sloan Manag. Rev. (2004).
16. Magnusson, J., Juiz, C., Gómez, B., Bermejo, B.: Governing technology debt. Proc. 2018 Int. Conf. Tech. Debt - TechDebt '18. 76–84 (2018).
17. Schlichter, B.R., Kraemmergaard, P.: A comprehensive literature review of the ERP research field over a decade. vol. 23 No. 4, pp. 486-520 (2010).
18. Webster, J., Watson, R.T.: Analyzing the past fo prepare for the future: writing a literature review. MIS Q. Vol. 26, No. 2, pp. xiii-xxiii (2002).
19. Boell, S.K., Cecez-Kecmanovic, D.: A hermeneutic approach for conducting literature reviews and literature searches. Commun. Assoc. Inf. Syst. Vol 34. (2014).
20. Harzing, A.W.: Publish or Perish. https://harzing.com/resources/publish-or-perish (2007).
21. Becker, C., Chitchyan, R., Betz, S., McCord, C.: Trade-off decisions across time in technical debt management. In: Proceedings of the 2018 International Conference on Technical Debt - TechDebt '18, pp. 85-94. (2018).
22. BenIdris, M., Ammar, H., Dzielski, D.: Investigate, Identify and Estimate the Technical Debt: A Systematic Mapping Study. Int. J. Softw. Eng. Appl. 9, pp. 01–14 (2018).
23. King, N.: Doing Template Analysis. In: Symon, G. and Cassell, C. (eds.) Qualitative Organizational Research: Core Methods and Current Challenges. SAGE (2012).
24. Blaikie, N., Priest, J.: Designing Social Research: The logic of anticipation. Cambridge (2019).
25. Gregor, S.: The nature of theory in Information Systems. MIS Q. Manag. Inf. Syst. vol. 30, pp. 611–642 (2006).
26. Whetten, D.: What Constitutes a Theoretical Contribution. Acad. Manag. Rev. vol. 14, pp. 490–495 (1989).
27. Derman, E.: Metaphors, Models & Theories. Q. J. Financ. vol. 1, pp. 109–126 (2011).
28. Blomberg, J., Burrell, M.: An Ethnographic Approach to Design. In: Jacko, J.A. (ed.) Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. Boca Raton, FL (2012).
29. Sutton, R.I., Staw, B.M.: What theory is not. Adm. Sci. Q. vol. 40, pp. 371–384 (1995).
30. Fernandez-Sanchez, C., Garbajosa, J., Yague, A.: A framework to aid in decision making for technical debt management. 2015 IEEE 7th Int. Work. Manag. Tech. Debt, MTD 2015 - Proc. pp. 69–76 (2015).
31. Polit, D.F., Beck, C.T.: Generalization in quantitative and qualitative research: Myths and strategies. Int. J. Nurs. Stud. vol. 47, pp. 1451–1458 (2010).
32. Danish Competition and Consumer Authority: Hvor lang løbetid må en rammeaftale have? https://www.kfst.dk/faq/udbud/nyt-udbud/fase-1/hvor-lang-loebetid-maa-en-rammeaftale-have/ last accessed 2020/05/19
33. Bannister, F., Connolly, R.: The great theory hunt: Does e-government really have a problem? Gov. Inf. Q. vol. 32, pp. 1–11 (2015).
34. Mathiassen, L.: Designing Engaged Scholarship: From Real-World Problems to Research Publications. Engag. Manag. Rev. vol. 1, iss. 1, Article 2. (2017).

Exploration of metaphors as a way to understand socio-technical phenomena: An emergent analytical framework

Author 1[1], Author 2[2], and Author 3[3]

[1]Details for Author1

[2]Details for Author2

[3]Details for Author3

Abstract

Socio-technical phenomena are difficult to understand and communicate since they are intangible and neonatal. Researchers and practitioners have applied metaphors to understand them, prescribing a structure through their thinking. While metaphors are frequently applied, researchers have not discussed their application and limitations. We apply a mixed-method approach, exploring communities' application of metaphors for conceptual development of socio-technical phenomena, using two communities: Open Government Data and IT Development and Maintenance. We synthesized 21 articles and two books into the Communities' Applications Of Metaphors

(CAOM) framework, which serves as our analytical lens. We collected empirical material for each community's academic and practical parts between 2015–2020, resulting in 100 articles and 263 documents. We conducted a word frequency analysis and an in-depth analysis of the empirical material. The contributions are the CAOM analytical framework with the key concepts of metaphors' usage, expression, and metaphor complexes. We conclude that the application of metaphors in communities is influenced by community type and metaphors' role, while the topic influences metaphor selection. Metaphors are combined to create new ways of reasoning. We recommend that governments draw on action-oriented metaphors when writing policies for socio-technical phenomena, while digital government researchers should include an action-orientation when they develop metaphors.

Key points for practitioners:

- Practical communities can approach metaphors as tools to discover new ways of working and acting, departing from habits and traditional thinking.
- Combining metaphors is essential to create new ways of reasoning and understanding socio-technical phenomena.
- The CAOM framework provides a base for combining and exploring metaphors.

Keywords: Metaphor; Community Development; Socio-Technical; Analytical Framework; Open Government Data; Technical Debt

1 Introduction

As IT is emerging in peoples' work and research, new socio-technical phenomena are arising with an intangible nature, making them difficult to understand and communicate. People rely on their existing vocabulary and knowledge of known things and apply the similarities to describe the neonatal phenomena. Here, people can use metaphors to extend their vocabulary to encompass the neonatal socio-technical phenomena. Researchers, as senders, communicate about a socio-technical phenomenon to other researchers (receivers), using metaphors. This process has been taking place for years in research disciplines, such as information systems, management, and IT Development and Maintenance (e.g., Kendall & Kendall, 1993; Magnusson, Juiz, Gomez, & Bermejo, 2018; Morgan, 1998). For example, LNyhlen and Gidlund (2021) use 'Narratives' as a syntax to examine policy documents, more precisely, government's visions for digitization; Schedler, Guenduez, and Frischknecht (2019) use 'Smart' as a frame to describe new digital governmental initiatives based on AI, IoT, big data analytic, and sensor systems; and Gerrits (2021) use 'Soul' in the title as a rhetorical device to capture the receivers' attention for the black box of self-learning algorithms. Senders and receivers can also be practitioners (e.g., Accenture Federal Services, 2018; Carrara, Chan, Fischer, & Steenbergen, 2015; ODI, 2019).

Metaphors can help senders and receivers enrich their vocabularies (Black, 1955; Lakoff & Johnson, 2008a) and be viewed *"as essential to human understanding and as a mechanism for creating new meaning and new realities in our* *lives"* Lakoff and Johnson (2008a, p.196). Morgan (1998) explain that a metaphor is an implicit or explicit assertion that something is like something else that brings a specific structure to thoughts (Lakoff & Johnson, 2008b; Morgan, 1998), but it is not meaningful in the literal sense (M. Davies, 1982). Metaphors can enable senders and receivers to understand and address phenomena with their preceding logic syntax (Mészáros, 1966). As such, metaphors are an approach for conceptual development in communities.

While previous digital government research has applied metaphors for conceptual development (e.g., Custers & Bachlechner, 2017; Smith & Sandberg, 2018; Zuiderwijk, Janssen, Van De Kaa, & Poulis, 2016), it lacks a constructive and conscious discourse. A problem is that senders assume they share sufficient knowledge with receivers. If it is insufficient, receivers can misunderstand or fail to grasp the metaphors (Glucksberg, 1989; Ortony, 1975). Metaphor usage in ambiguous and knowledge-intensive situations with unclear purposes can hinder learning and lead to confusion (Hekkala, Stein, & Rossi, 2018). For example, 'Technical Debt' explains the impact of suboptimal solutions made during the development of IT systems to cut costs or time (Cunningham, 1992). It is a metaphor describing a socio-technical phenomenon and the name of said phenomenon, which has created confusion in the conceptual development (Fairbanks, 2020). Metaphors can also strongly influence how senders and receivers conceptualize and approach problems. They tend to search for information that confirms their metaphorical framing while seldom recognizing the metaphors' influence on their behaviors (Thibodeau & Boroditsky, 2011). Unsuccessful metaphors can generate confusion and despair (Ortony, 1975), while inappropriate metaphors can constrain thinking along undesirable lines. These metaphors embedded in formal communications may mislead or inhibit the learning of receivers (Hamilton, 2000). As such, it is important for digital government researchers to study and discuss the application of metaphors in their communities and related conceptual development.

Previous digital government research has not explored communities' applications of metaphors. We need this knowledge to improve the application of metaphors to better comprehend new socio-technical phenomena (e.g., smart government and smart cities), but also to better develop concepts, such as open government data (OGD) (e.g., Altayar, 2018; Dawes, Vidiasova, & Parkhimovich, 2016; Susha, Grönlund, & Janssen, 2015). OGD is data shared by public organizations for anyone to reuse without restrictions (Handbook, 2015). Therefore, this article provides knowledge on how metaphors are applied in two digital communities; such knowledge can improve our application of metaphors. It is currently possible that the present application of metaphors could constrain thinking and lead to confusion, hampering conceptual development while being difficult to recognize (Thibodeau & Boroditsky, 2011).

The purpose of this research is to explore conceptual development in communities by studying researchers' and practitioners' application of metaphors about socio-technical phenomena in their communications. Therefore, we synthesize literature on metaphors into an analytical framework, which we call the 'Communities Application of Metaphors' framework (CAOM framework). We select the Open Government Data (OGD) and technical debt (TD) communities (see subsubsection 3.2.1). We study the communities for a five year period using the CAOM framework as an analytical lens (see Section 4). Our research was guided by the following research questions:

- How have the OGD and TD communities applied metaphors in their conceptual development between 2015 and 2020, and why in this manner?
- What are the lesson learned for analyzing the application of metaphors for the OGD and TD communities?

The article starts with an overview of previous research and our research approach, and then we present the CAOM framework. It continues by giving the quantitative and qualitative findings (in a section for each). We discuss the findings as researching metaphors, the usage of metaphors, and the implications for researchers and practitioners. The article ends by presenting the conclusions with limitations and future research.

2 Background

This section presents the theoretical grounding for the analytical framework synthesized in this research. It starts by explaining metaphors and then giving an approach to understanding how they spread within a community.

2.1 Metaphors

Information polity, digital government, and information system communities, in general, use metaphors frequently to comprehend and communicate socio-technical phenomena.

The anatomy of a metaphor is display Fig. 1, where a **sender** and a **receiver** possess **origin thoughts** and **origin words** that they can use to communicate the **origin phenomenon** to the receiver. A sender who encounters a new **target phenomenon** and experiences **target thoughts** about it but lacks **target words** can find it challenging to communicate. The sender can solve this problem by abstracting their thoughts about similar phenomena to produce a metaphor as a generalization (Ogden & Richards, 1923).

The sender borrows origin words to use as target words (**metaphors**). A metaphor borrows partial thoughts of one phenomenon to apply to a target phenomenon, a form of generalization. Morgan (1998) describes that *"[... a] metaphor proceeds through implicit or explicit assertions that A is (or is like) B."* (Morgan, 1998, p. 4). As such, metaphors are a conceptual approach to recycling and expanding previous knowledge of the unknown. The receiver infers similarities between an origin phenomenon and a target phenomenon from a metaphor that exclude others (Lakoff & Johnson, 2008b). The receiver must **sort out** the differences and focus on the similarities (Glucksberg, 1989).

However, some similarities may be only superficial, which can make a metaphor misleading (Mars, Bronstein, & Lusch, 2012). A metaphor aids by providing a specific perspective on a target phenomenon. However, the receiver's understanding of a metaphor depends on its experiences with the origin phenomenon (Lakoff & Johnson, 2008b). A metaphor implies a way of thinking as it brings a specific structure to target thoughts based on origin thoughts and a way of understanding the world (Lakoff & Johnson, 2008b; Morgan, 1998). For example, technical debt refers to a debt that should be repaid; however, this is not always the case for technical debt. As a result, a sender and a receiver can evaluate a metaphor as beneficial or disadvantageous based on well it conveys target thoughts about the target phenomenon.

2.2 Metaphors from a Meme Perspective

The application of metaphors in a community can be understood from the meme perspective (Dawkins, 2016). A meme is a convenient unit of cultural transmission (e.g., an idea or a tune) that can jump from one sender to another receiver through a process of imitation. A meme can be a metaphor, which we presume in this paper. A metaphor spreads from sender to receiver through imitations where the metaphor can mutate or blend with other ideas (or metaphors). A metaphor spreads based on whether the receiver perceives the metaphor as having utility and appeal (Dawkins, 2016). A metaphor's appeal relates to how well it can communicate a target phenomenon. The metaphor borrows the structure of the origin phenomenon to apply to the target phenomenon initially, and the focus is on the similarities between the origin phenomenon and the target phenomenon (Glucksberg, 1989; Lakoff & Johnson, 2008b; Ogden & Richards, 1923). A high appeal metaphor entails being a well-known origin phenomenon and where the similarities between the origin and target phenomena are easily identified. The metaphor loses its appeal if the dissimilarities between the origin and target phenomenon create more confusion than the similarities provide understanding. For example, the 'e' (electronic) is slowly dying as digital, online, and virtual is taking over (e.g., the eGovernment is becoming Digital Government). A beneficial and high-appeal metaphor can be characterized as self-explanatory (Ortony, 1975), while a low-appeal metaphor can generate confusion and despair (Ortony, 1975). However, a metaphor can have high-appeal but also be disadvantageous, as such spreading quickly with consequences. On the other hand, a metaphor can lie dormant for years in various mediums (Dawkins, 2016), such as the metaphor "Technical debt", which was first introduced in 1992 by Cunningham. It took a decade before it was picked up and discussed in the academic community (Tom, Aybuke, & Richard, 2012). A metaphor can co-adapt to ideas and other metaphors in its environment where associations are entrenched. It grows interdependent on other ideas and metaphors and can no longer exist without them. Possibly, they self-reinforce each other. This situation can lead to stable structures of metaphor complexes, making it difficult for other metaphors to be adopted or accepted by receivers (Dawkins, 2016).

3 Research Approach

Our study was iterative and oscillates between solo and group work, using a mixed methods approach to explore the application of metaphors within communities (see Creswell & Creswell, 2017). It followed three phases: (1) we synthesised an analytical framework from previous research, (2) we chose and collected empirical material about two communities, and (3) we analysed the empirical material with word frequency analysis and then in-depth analysis based on the analytical framework. We used triangulation to increase the validity of the research (Saunders & Lewis, 2012). The triangulation was achieved by comparing empirical sources. Fig. 2 presents an overview of our study.

3.1 Phase 1: Synthesis of An Analytical Framework

We used a hermeneutic literature review with citation tracking and citation analysis (Boell & Cecez-Kecmanovic, 2014) to review previous metaphor research. We combined the keywords "metaphor" or "metaphor development" with "communities" or "organizations" in Google Scholar. We skimmed titles, abstracts, and possibly the contents of literature to determine whether it enables us to understand metaphors and their development. We classified relevant literature following title, purpose, research problem, keywords, limitations, implications, explanatory powers, and interesting statements. In total, we classified 21 articles and 2 books. We synthesized an analytical framework based on the relevant literature, using concept mappings where we defined concepts and their relationships (Maxwell, 2012). An analytical framework, like a theoretical framework, draws concepts from a theory to shed light on a phenomenon Imenda (2014). We understand that an analytical framework leaves certain concepts open to be filled with empirical material and, as such, allow for the generation of new explanations and insights. The analytical framework guided and enabled us in explaining and interpreting the empirical material Imenda (2014). The analytical framework is labeled the CAOM framework (see Section 4).

3.2 Phase 2: Data Collection

We selected two communities based on accessibility, differences in community structure, active metaphor usage for a present socio-technical phenomenon, and containing a mix of researchers and practitioners. We decided to study the OGD and TD communities.

3.2.1 Introduction to the OGD and TD Communities

We understood the OGD and TD communities to be subcommunities of the overarching Information System community. The OGD community is public, with key actors, and expresses itself to an abundance on the Internet. While TD is less prominent on the Internet, contextual bounded to the IT system and local practitioners. They differ in level of analysis, OGD contains a system perspective and focuses on actors and their interactions, whereas TD focuses on a local phenomenon related to a specific IT system. OGD has been understood as ecosystems (e.g., Dawes et al., 2016; Pollock, 2011), lifecycles (e.g., Attard, Orlandi, Scerri, & Auer, 2015; Folmer, Reuvers, Quak, van den Broek, & van Veenstra, 2011), and chains (e.g., Albano, 2013; Carrara, Chan, et al., 2015), while TD has been understood as loan (Kruchten, Nord, & Ozkaya, 2012), investments (Guo & Seaman, 2011), and shortcuts (Cunningham, 1992). The communities also use a mix of metaphors. For example, in OGD, T. Davies (2011) uses ecosystem and infrastructure, Lindman, Kinnari, and Rossi (2015) use ecosystem, network, and chain, and Harrison, Pardo, and Cook (2012) use ecosystem, platform, network, and market metaphors. On the other hand, in TD, Kruchten et al. (2012) use loan, evolution, landscape, and development, Cunningham (1992) use shortcut, and Guo and Seaman (2011) use investment metaphors.

3.2.2 Data Collection in the Communities

Our data collection was divided into researchers and practitioners communities. Fig. 7 and Fig. 8 present a summary of the number of empirical documents over time collected in our study.

Research Communities. We created a list of outlets (journals and conferences) covering OGD or TD that contains English peer-reviewed, international research between 2015 and 2020. We excluded outlets that lacked a search engine and the ability to limit the searches by year. From this list, we chose three OGD outlets and three TD outlets based on the number of articles on their respective topics with a mixed sample in mind. We searched through the outlets to filter articles in two rounds of selection. First, we selected articles published within 2015 to 2020, written in English, and about the topics of OGD and TD. Second, we included accessible, full research articles, and excluded literature reviews, technical research, methodological research, and non-machine readable articles. In total, we identified 50 OGD articles and 50 TD articles for TD. Table 1 presents the articles divided over the outlets of respective community with the two selection rounds. We analysed these articles in phase 3 of our study.

Practical Communities. We explored the practical communities through desk research and identified data sources. We downloaded empirical material from the data sources, if it had sufficient quality and met our selection criteria (Table 2). We adapted our data collection to the differences in community structure of the OGD and TD communities. In the **OGD community**, we chose to study the European Data Portal (EUDP) and the Open Data Institute (ODI), as they provide rich streams of English public documents, act as international nexuses of OGD knowledge, and show some use of metaphors. We explored the actors' website with a focus on knowledge-intensive documents, as we believed they could capture the application of metaphors in the OGD community. In total, we identify 34 documents for EUDP and 126 documents for ODI. In the **TD community**, we extended our exploration to include English and Danish as of limited search results. Danish had the highest number of hits amongst the Scandinavian languages. We chose 11 legitimate actors and identify a total of 103 documents. The TD community's structure is semi-private and fragmented in nature, to enter the community, we conducted 17 semi-structured interviews in a Danish Agency. The interviewees worked with the development and maintenance, either as business or legal perspective, IT architect, developer, or application manager. The IT systems are developed by the agency and software consultancies as onsite vendors. The agency has a good history with IT-development and works closely with their software consultancies in developing and maintaining IT systems.

3.3 Phase 3: Data Analysis

We first analysed the empirical material in MaxQDA using a word frequency analysis and then in-depth based on the CAOM framework (see Section 4). Both methods used metaphor density to inform us about possible shifts in communal perspectives. Density was calculated per metaphor for each year by summing metaphor applications in the year and dividing this amount by the number of documents for that year.

Metaphor Frequency Analysis. The word frequency analysis (WFA) (see Izwaini, 2003) resulted in statistics about the frequency of certain words in the empirical material. It provided us with an estimation for metaphor usage, as the method does not consider the word's context. This analysis followed three steps. First, we identified search terms for a dictionary. We understand categories as metaphor groups where the search terms are possible metaphorical words (Table 3). To remove unnecessary words, we applied WFA with filters of word frequency above 19, length above two, and the standard stop list of MaxQDA. We sorted out everyday, non-metaphorical, fragmented words into a stop list or our dictionary and included metaphorical words into a dictionary. We discussed uncertain words and decided their status. For OGD, we started with 373.297 possible words and ended with 190 metaphorical words. For TD, we began with 393.103 possible words and ended with 76 metaphorical words. Second, we discussed and rearranged the metaphorical words into metaphor groups, and we decided to focus on six OGD groups and six TD groups. Third, we applied the metaphorical groups on the empirical material for respective community using MaxQDA's category matrix browser. We examined the matrix for patterns. If we identified anything unexpected or unusual, we browsed the metaphor group and its search terms in the respective documents. If a word is used more in a non-metaphorical manner than in a metaphorical manner, we divided it into more specific metaphorical words. For example, the search term *climate* counts *climate change*, as such we divide it into the metaphorical words *political climate*, *business climate*, and *organizational climate*. On the other hand, some words could not be specified because of complexities in context, these words were deactivated in our dictionary. For example, in TD, we deactivated *negative* as it is primarily used to describe results rather than as a metaphor for 'Emotion'. We visualized the final statistics as charts in Excel to reveal patterns. The charts were based on the metaphor's usage density. We discussed the charts to draw tentative conclusions about the communities' application of metaphors. The charts presented the statistics by year divided over the practical and researcher parts of the communities. Then, we applied the CAOM framework and examined the charts for metaphor *usage* and *expression.* We analyzed *usage* by studying changes in metaphor density for and between years (short-time changes, revealing speed of replication), while we analyse *expression* by examining the usage over time (illustrated by six patterns Section 5, revealing stability over time). The metaphor *complex* could only be explored in the qualitative analysis.

Qualitative Analysis. We discuss the patterns to decide on one metaphor to deeper explore per community. We decided to explore the application of 'Ecosystem' (for OGD) and 'Technical debt' (for TD) based on their differences in application and metaphor density. We create an analytical template based on our analytical framework (see Section 4) that we fill out with the empirical material for the selected metaphors. We apply the framework for each year and examined the density and application of the metaphor. We use the analytical framework to examine how it has changed over time within a community. We pay attention to deviant cases or negative cases to ensure the quality of our findings and framework. We examine the individual items or case which appear to deviate from the developed framework (Mays & Pope, 2000).

4 The CAOM-Framework

The Communities' Application Of Metaphors (CAOM) Framework is a final product and contribution of this article (see Walsham, 1995). It can be iteratively used to analyze the application of metaphors within a community's communication. It assumes the application of metaphors happens towards the community's conceptual development to understand socio-technical phenomena. The framework follows a quantitative and a qualitative approach.

The quantitative approach provides high-level insight into how the application of metaphors in a community develops over time. The framework conceptualizes this communication as a stream of messages containing metaphors. The continued application of metaphors forms **flows** within this stream, such as an ecosystem or technical debt flow. This conceptualization enables us to analyze a flow for increases and decreases in the application of metaphors, which provides us with a quantitative estimation of a metaphor's longevity (*stability* over time) and fecundity (*speed* of replication) (Dawkins, 2016).

A metaphor's copying-fidelity, referring to the senders' ability to accurately replicate the metaphor (Dawkins, 2016), requires a study of changes and blends in the application of this metaphor. The qualitative approach employ us to analyze the community's application of a metaphor in detail. Specifically, we analyze the application of a metaphor as usage, expression, and complex.

Usage. We examine how a sender uses a metaphor in its communication. Whether it is merely mentioned, discussed and defined, or used as a tool. A sender or a receiver can use a metaphor for several purposes, such as (1) enhance its understanding and learning of a target phenomenon, (2) guide to possible directions for refinement and specification of thoughts, (3) describe shapes, (4) explain the unfamiliar or unknown, (5) communicate abstract concepts, (6) as an invitation to conversations, and (7) provide constraints to its thinking (Glucksberg, 1989; Haack, 1988; Hamilton, 2000; Ortony, 1975).

Expression. A metaphor's expression entails examining where the metaphor appears in a sender's communication and to what extent. An appearance in the introduction with no further explanation indicates the sender perceives the metaphor to have a high appeal and be immediately understood by receivers (Glucksberg, 1989; Lakoff & Johnson, 2008a). On the other hand, the metaphor can also appear several times in the text if the senders use the metaphor for one of the various purposes, such as (1) an analytical lens in qualitative research (Kaarst-Brown & Robey, 1999), (2) help in methodology (Kendall & Kendall, 1993), and (3) a framework to conceptualize a targeted phenomenon in a particular way (Oates & Fitzgerald, 2007).

Complex. A sender's communication contains complex metaphors and concepts (see Section 2). A metaphor can highlight *some* of the similarities between an origin phenomenon and a targeted phenomenon (Glucksberg, 1989; Lakoff & Johnson, 2008b; Ogden & Richards, 1923). A sender can, in their communication, use and combine several metaphors (e.g., Izwaini, 2003). As such, we understand that it is possible for metaphors to be combined or supplement each other, highlighting different aspects of the target phenomenon where metaphors can have diffident roles. A complex could, therefore, reveal what similarities a sender is attempting to highlight and dissimilarities it is trying to abate.

5 Metaphor Frequency Analysis as Patterns

In this section, we explore the density of estimated metaphor usage and identify patterns through the period 2015–2020 for the OGD and TD communities.

Six metaphor groups were analyzed for the OGD community: 'Ecosystem', 'Silo', 'Creature', 'Market', 'Industry', and 'Adventure' (Table 4). No clear common pattern is visible. The metaphor density of the OGD researchers (see Fig. 4) is first stable with few fluctuations, ending with a sharp increase in density between 2019 and 2020. On the other hand, for OGD practitioners (see Fig. 3), the metaphor density fluctuates following a decreasing trend.

The TD community's six metaphor groups were: 'Economics', 'Emotion', 'Loan', 'Person', 'Purchase', and 'Repair'(Table 4). Similar to the OGD community, no clear common pattern is visible. For TD researchers (see Fig. 6), the metaphor density first fluctuates, followed by a sharp increase of all metaphor groups between 2019 and 2020. The metaphor density of the TD practitioners (see Fig. 5) is first stable, then expresses a sharp increase followed by a lower metaphor density for all metaphor groups.

In the metaphor frequency analysis, six common patterns were identified for the OGD and TD communities. First, the **waver pattern** starts with a slow increase in metaphor density, reaching a plateau or peak, followed by a slow decrease in density. Second, the **bumper pattern** has first a higher stable period of metaphor density, which is abrupted by a bump (an increase, peak, and a decrease in density), ending with a lower stable density period. Third, the **quaker pattern** starts with a valley (decrease in density, bottom, and increase density). A solid increase follows the valley in density, peak, and a substantial decrease in density. Another valley follows this peak and then a smaller increase in density, peak, and decrease in density. Fourth, the **climber pattern** is a slow moving plateau that has either increasing or decreasing density over several years. Fifth, the **duner pattern** starts with a quick increase in density, reaching a peak, followed by a slow stable decrease in density, ending with a rapid decrease in density. Sixth, the **rocket pattern** starts with an increase in density, reaching a peak followed by a valley. It then expresses a sharp, high increase in density.

Table 4 gives an overview of the identified patterns for the OGD and TD communities. The researcher communities have a higher metaphor density than the practitioner communities. This difference is possibly a consequence of researchers' focus on concept development, while practitioners' focus could be on using the metaphor. In addition, only the researcher communities contain the rocket pattern. Likewise, the waver pattern is only apparent in the practitioner communities. Whereas the Bumper pattern is not visible in the TD researcher community, but it appears in the remaining communities. Thus, there are similarities in patterns within the researchers' communities and practitioners' communities but not within the OGD community or the

TD community. This result indicates the community's use of metaphors is more aligned between practitioners and between researchers than the community topic. Furthermore, in a cross-comparison, the TD researchers and OGD practitioners fluctuate in density. They are likely experimenting with metaphors or at least discussing them. A possible explanation is that a useful metaphor complex has not been generated. On the other hand, the OGD researchers and TD practitioners seem stable. They have likely found a working complex of metaphors or do not need metaphors to describe the target phenomenon. This similarity could mean that different communities can express similar behaviors using metaphors for conceptual development. The two communities lack clear common patterns, which requires an in-depth analysis.

6 The In-depth Analysis of Metaphor Application

In this section, we present the in-depth analysis following the CAOM-Framework (see Section 4), ending with a comparison.

Usage. Senders' usage of metaphors ranges from everyday, syntax, tool, and development. **Everyday** usage assumes the metaphor to be accepted amongst or known by the receivers. For example, Susha et al. (2015) use the 'Ecosystem' metaphor to describe a research stream with a particular position. **Syntax** usage frames thinking, helping senders and receivers to perceive or approach a socio-technical phenomenon. For example, Charalampidou, Ampatzoglou, Apostolos, and Avgeriou (2017) use 'Technical debt' to create context and relevance for studying code smell; *"[...] we explore code smells, which according to practitioners are the most commonly occurring type of TD in industry, by assessing the associated interest probability."* (Charalampidou et al., 2017, p. 1). **Tool** usage help senders and receivers to see new ways to act or work with a socio-technical phenomenon. For example, ODI (2019) presents a tool to map an 'Ecosystem'. The tool follows four steps: (1) map actors, (2) map the formal value exchanges, (3) map the 'soft' value exchanges, and (4) find opportunities. **Develop** usage can combine tool and syntax usage but focus mainly on refining the metaphor for some purpose. For example, Siebra, Oliveira, Seaman, Silva, and Santos (2016) use Grounded theory to develop 'Technical debt' into a theory;

> *"Thus, this work contributes to the effort in building a formal theory about TD and provides directions to assist the work of developers/managers who intend to identify and monitor TD items in their projects, given the practical nature of this study."* (Siebra et al., 2016, p. 219).

Expression. Senders can express metaphor differently in their communication: at the start, the end, as crumbles, or a mix. At the **start**, a metaphor helps to position the communication within a larger discourse or supply initial reasoning about a target phenomenon. For example, Carrara, Chan, et al. (2015) use 'Ecosystem' to frame the context of value creation. In the **end**, a metaphor helps to wrap up the communication or encourage the use of the metaphor. For example, while Altayar (2018) studies motivations for open data adoption from an institutional theory perspective, the author ends with the following recommendation: *“Regarding the specific context of the study, it is important for policy makers in Saudi to take into consideration the development of OGD using an ecosystem approach [...]”*(Altayar, 2018, p. 642). As **crumbles**, the communication entails sporadically mentions of the metaphor. For example, Carrara, Fischer, Oudkerk, van Steenbergen, and Tinholt (2015) mention ’Ecosystem’ in the middle of the report. A **mix** is a combination of the the above expressions (e.g., Bonina & Eaton, 2020; Dawes et al., 2016; McBride, Olesk, Kütt, & Shysh, 2020).

Complex. A sender can use and combine several metaphors and concepts in their communication, forming a complex. The focus is on the role of a studied metaphor within its complex. A metaphor can have a **core** role, providing a specific perspective on the targeted phenomenon. In TD, several sub-categories of the metaphor emerge, such as architectural debt and code debt (e.g., Fontana, Ferme, Zanoni, & Roveda, 2015; Verdecchia, Malavolta, & Lago, 2018). The role of a metaphor can be **supplementary**, helping to create a new metaphor or to enhance the meaning of concepts, phenomena, or other metaphors. For example, Khayyat and Bannister (2017) focus on co-creation, but draws on ’Ecosystem’ to put the activity within a larger whole; *”Co-creation takes many forms. The type of co-creation that emerged in this research can best be described as a type of ecosystem in which various members or stakeholders interact and interde-*

pend in various ways." (Khayyat & Bannister, 2017, p. 14). A metaphor can have a **peripheral** role where it is briefly related to other metaphors, phenomenons, or concepts. On the other hand, Fontana et al. (2015) apply TD to set the scene for code smell's relevance as a research topic; *"Code smells can be used to capture symptoms of code decay and potential maintenance problems that can be avoided by applying the right refactoring. They can be seen as a source of technical debt. "* (Fontana et al., 2015, p. 16). Lastly, the metaphor can have an **insignificant** role where it is merely used once or twice in a reference list, footnote, related posts, or without metaphorical meaning in the text.

Table 5 gives an overview of the in-depth analysis of 'Ecosystem' and 'Technical Debt', drawing on the above dimensions. The in-depth analysis highlights that 'Ecosystem' is often used in an everyday manner where researchers develop the metaphor while practitioners use it as a tool. Its role is often insignificant or peripheral, with some instances of being a core role. In comparison, 'Technical Debt' tends to be developed by researchers, and it has everyday usage by practitioners. Its role is often core within research, while it varies for practitioners.

7 Discussion

This research has explored the conceptual development of the OGD and TD communities by studying researchers' and practitioners' application of metaphors about socio-technical phenomena in their communications. While previous research tends to focus on a metaphor and its related target phenomenon (e.g., Glucksberg, 1989; Hamilton, 2000; Ogden & Richards, 1923; Ortony, 1975; Thibodeau & Boroditsky, 2011), our study broaden this research by perceiving metaphors as part of a community's conceptual development for a socio-technical phenomenon (see Section 4). The CAOM framework combines previous research about metaphors with a meme perspective, resulting in an original approach to understanding and studying metaphors. This section discusses the usage of metaphors within the OGD and TD communities, our applied research approach for studying metaphors, and implications for practice and research.

7.1 The Usage of Metaphors

This section discusses the communities' usage of metaphors, focusing on the identified patterns and metaphor complexes (see Section 5 and Section 6). Our findings show that researchers discuss, share metaphors, and apply them for conceptual developments, whereas practitioners primarily apply the metaphors. Researchers display a higher metaphor density than practitioners and cite previous research. It is, as such, easier to trace the development and changes of metaphors in the research-part of the communities. This difference is most likely a reflection of the fact that researchers develop knowledge about less known socio-technical phenomena while practitioners are encountering or creating these phenomena.

Communal Usage of Metaphors. While Section 5 and Section 6 present how the OGD and TD communities applied metaphors in their conceptual development between 2015 and 2020, answers partly the first research question. The findings need a further highlight to give an idea about why do the TD and OGD communities apply metaphors in a certain manner? The communities apply several metaphors each playing a specific role within a complex. OGD is a name for the core socio-technical phenomenon (the data that are published and reused following certain principles (Attard et al., 2015)), but needs metaphors to be further developed and give context. For example, 'Ecosystem', 'Market', and 'Industry' give terms to describe the larger systems of actors and interactions surrounding OGD. On the other hand, 'Technical Debt' is both a developing metaphor, but also the everyday name for a specific socio-technical phenomenon. TD community develops the metaphor more compared to OGD because of TD being the core of the field. While the socio-technical phenomenon influences the selection of possibly applicable metaphors, as it together with a sender give rise to a metaphor (see Section 2), the findings show that it has a negligible influence on the application of these metaphors. As such, the role of a metaphor within a complex influences the senders' application of said metaphor. For example, within the researcher community, TD has become the central concept and is aided by financial and economic metaphors and to a lesser degree other metaphors (e.g., refactor, code smell, and bugs). In the practitioner community, the TD has become an everyday metaphor and the phenomenon is not developed on its own but used with other concepts (e.g., decoupling, general IT maintenance, legacy systems). Practitioners combine metaphors and concepts, while researchers expand upon metaphors and concepts with other metaphors and concepts, meaning they express different community types.

Additionally, the OGD and TD communities varies further in community type. OGD is open public community with more complex interactions which makes it challenging to separate concepts and metaphors. We expected that this openness would have contributed to a higher degree of development for the ecosystem metaphor, as ideas and knowledge can easily flow in and out of the community. However, this was not the case. Instead, they had a high usage of various metaphors. On the other hand, TD is a semi-private community, with a strong core of researchers and software developers who are developing the metaphor. They appear to take an ownership of the phenomenon and try to solve the challenges it poses. The phenomenon is expanding to other communities (e.g., engineering) and other job functions (e.g., management). A surprising insight is then that open communities everyday use peripheral metaphors more, while closed communities developed core metaphors more.

As an answer to the 'why'-question, the application of metaphors varies based on community type and the metaphors' role within a complex. The socio-technical phenomenon influences the selection of applied metaphors, but has a negligible influence on the application of these metaphors.

Patterns in Metaphor Frequency. In the period between 2015 and 2020, we identified six patterns in the application of metaphors for the OGD and TD communities (see Section 5). While we identified no clear common patterns between the communities, we found that there are similarities in the patterns within the researchers' communities and within the practitioners' communities but not within the OGD communities nor the TD communities. This difference indicates that the community's application of metaphors is more aligned between practitioners and between researchers than based on a common socio-technical phenomenon. At the same time, only the researcher communities contain the Rocket pattern; likewise, the Waver pattern is only apparent in the practice communities. Researchers could be more sensitive to the appeal of a metaphor, seeking to experiment and test it, while practitioners take longer to accept a metaphor because they need to have practical utility. Whereas the Bumper pattern is not visible in the TD researcher community, but it appears in the remaining communities. The other communities might, thus, be more open to experimentation with metaphors. In a cross-comparison, the TD research community and OGD practice community display fluctuation in density. They are likely experimenting with metaphors or at least discussing them. It is possible that a useful metaphor complex has not been generated. On the other hand, the findings show that the OGD research community and TD practice community are more stable in their metaphor density. They have likely found a useful metaphor complex or do not need metaphors to describe the target phenomenon.

This similarity could mean that different communities can express similar behaviors towards using metaphors for conceptual development. This conclusion is in contradiction to Stubblefield (1998)'s statement that metaphors can have lifecycles, as our findings show that metaphors are part of ever-evolving processes following similar patterns.

Metaphor Complexes. Our findings show that metaphor complexes can be used to create a new logical syntax. For example, 'Technical Debt' is combined with 'Repair' rather than 'Repay'. The metaphors fit with the target phenomena but could be understood as incompatible if compared to each other. However, Mészáros (1966) warns that the metaphorical fallacy is to draw a conclusion from something that is only metaphorically established. It is, as such, possible that senders and receivers create similarities and reasoning, which are not (fully) applicable to the target phenomenon. For example, in the OGD community, ecosystem is combined with business and health. Where businesses and ecosystems behave differently (Mars et al., 2012) and health is a metaphor that originates in medicine for an organism and is now stretched to groups (Rapport, Gaudet, Constanza, Epstein, & Levins, 1998). As a result, senders' and receivers' construction of metaphor complexes is similar to using their minds as a lab to discover new ways to express understandings and thinking about a target phenomenon by drawing on previous knowledge. In addition, our research indicates that senders can employ complexes to help receivers sort similarities and differences. These complexes can oscillate over time with some recurrent configurations (e.g., in the OGD community, 'Ecosystem' tends to be paired with 'Business' or 'Production'). At the same time, Mészáros (1966) explains that a metaphor is more than a word and is similar to an island within an ocean of communication that helps receivers to better understand the message. A metaphor can provide a logical syntax. The metaphor-complex approach opens questions about inter-metaphorical emergence, meaning senders and receivers could create new meanings through combinations of metaphors that could not be expressed by a single metaphor (as shown in the 'Technical Debt' example above). It is possible that the oscillations of the complexes are attempts by the senders to capture target phenomena difficult to express. In addition, senders and receivers can use a metaphor as a tool to reason and make sense of new or complex phenomena (e.g., Hekkala et al., 2018; Stubblefield, 1998; Thibodeau & Boroditsky, 2011). Therefore, a complex could help senders create new ways of reasoning and understanding (a new logical syntax) about a target phenomenon, which a single metaphor could not express even if the metaphors when literally compared would be incompatible.

7.2 Studying Metaphors

The study of metaphors in a community is complicated as several senders can participate in the communication through various outlets. Some senders explain their metaphors, while other senders assume the receivers will spring to the right spontaneous conclusions (Ogden & Richards, 1923). In addition, the metaphors can differ in usage, expression, and complex where they are sometimes not used as metaphors or at least border between being a metaphor and being everyday language or an unfettered concept. Therefore, it can be difficult to determine the exact meaning of a metaphor. We can draw some lessons learned about researching metaphors related to the OGD and TD community structures, the emergent CAOM framework, the frequency analysis, and the in-depth analysis.

Community Structure. The community structure influences how senders communicate to receivers but is also acts as an entry barrier for newcomers to the community. The OGD community is open, public, and international, allowing easy access for receivers (and researchers). The challenge was to find relevant key outlets where the majority of essential communication happens (relevant empirical material). The format and quality of this communication varied, which made the research time-consuming, giving less gain per resource invested (as seen as an increase in the 'Ecosystem's role as insignificant in complexes. See Table 5). Moreover, the TD community is semi-private and local, which can create an entry barrier for newcomers but also increase the quality of the material. It requires more effort from the researcher to access or become part of the community but also provides greater gains. The entrance barriers include cultural and language barriers (e.g., the researchers are very strict on what can be defined as technical debt).

The Emergent CAOM Framework. We synthesized the CAOM framework from previous research about metaphors and memes. The metaphor research gave a foundation for the framework, while the meme research provided an approach to study and understand the spread and application of metaphors within a community. This combination is an original contribution to previous research. The framework focused our attention on important parts of the empirical material while giving structure to the analysis. It enabled us to bridge the use of a quantitative and a qualitative method.

Frequency Analysis. The quantitative analysis used word frequency analysis to estimate the application of metaphors within the empirical material over time. While it required several iterations to become satisfactory, it was useful as a tool to gauge the senders' application of metaphors, as it helped to identify patterns. However, this method does not consider the context of words, giving a rough estimation where the researchers need to decide to exclude or include words in the grey zone (words used in both a metaphorical and non-metaphorical manner). The analysis method is fast to apply but time-consuming to set up and difficult to get right. At the same time, the lack of consideration for context meant that metaphor complexes were only analyzable in the in-depth analysis. This exclusion allowed the frequency analysis to focus on the usage and expression of metaphors over time. It revealed when a metaphor had a high potential to influence the conceptual development of a community but also when it could have limited potential.

In-depth Analysis. The in-depth analysis focused on a metaphor per community using the CAOM framework. It allowed a richer understanding of a metaphor's usages, expressions, and complexes. We saw a correlation between the density, usage, expression, and metaphor complex. For example, the TD researcher community has a high density, tends to primarily use the metaphor for development purposes, the expression is mixed, and the metaphor has a core role. Although the in-depth analysis was time-consuming, it helped to explain the usage and trends of metaphors within the two communities. It was an essential part to understand metaphor complexes.

7.3 Implications

Our research has some implications for practice and research. Practitioners and researchers should be aware that metaphor complexes can cause unwanted associations for receivers. Metaphor development allows the sender to experiment with different configurations of metaphors to create new logical syntax and meanings. Inter-metaphorical emergence can be beneficial but can lead to metaphorical fallacies or nonsensical communication. Receivers are the catalyst who interpret the metaphor and its metaphor complex. Practitioners and researchers should consider the potential impacts metaphors could have on each other.

Practice. Practitioners have a repertoire of actions and interpretations for the application of metaphors. They should reflect on what a metaphor entails for their thinking and actions, but also its possibilities. Senders could use the CAOM framework to analyze metaphors, which could help them avoid pitfalls in their communication and create complexes that foreground what they seek to communicate. Receivers can use the CAOM framework to gain a deeper insight into a metaphor. Especially, if the sender is attempting to communicate something only expressed through the emergence of a metaphor complex. We recommend that practitioners approach metaphors as a serious tool that they can use to quickly expand previous knowledge to encompass new and unknown socio-technical phenomena. This process can involve evaluations and reports on the success of a metaphor and its complex. We recommend that governments who seek to communicate to practitioners through policies or documents should draw on action-oriented metaphors, which can guide the practitioners towards an intended goal.

Research. Researchers can use the CAOM framework as an analytical lens. It captures the underlying mechanisms of metaphor usage, expression, and complex. The framework is developed to be used in mixed methods and qualitative research; as such, it is ample for use in case study research. Researchers could use the CAOM framework as part of their analysis, as an introductory guide to metaphors, and as a starting point for future research that seeks to theorize metaphor usage in communities (see Eisenhardt, 1989; Walsham, 1995). On the other hand, our findings imply that researchers should be aware of metaphors and their complexes. We believe complexes can give insight into what is perceived as new by practitioners and researchers, which could direct future research to original contributions. Similarly to practitioners, researchers could discuss the emergence of new meanings when combining various metaphors to increase the speed of conceptual developments. We recommend that digital government researchers include an action-oriented element when they develop or use metaphors, as it can improve practitioners use of their research.

8 Conclusion

Our study aimed to explore conceptual development in communities by studying researchers' and practitioners' application of metaphors about socio-technical phenomena in their communications. We employed a mixed-method approach, analyzing the application of metaphors between 2015 and 2020, focusing on the TD and OGD communities. We synthesize metaphor literature into the CAOM analytical framework with the key concepts of metaphors' usage, expression, and metaphor complexes, which is a contribution of our study (see Section 4). Our study was guided by two research questions, which it answered:

First, how have the OGD and TD communities applied metaphors in their conceptual development between 2015 and 2020, and why in this manner?

- The OGD community applies metaphors to supplement the main concept, while the TD community applies TD as a syntax to develop the main metaphor further. Practitioners combine metaphors and concepts, focusing on action, while researchers expand upon metaphors and concepts with other metaphors and concepts, focusing on knowledge development. Generally, researchers and practitioners blend metaphors to create new ways of reasoning and understanding. The metaphors could be understood as incompatible if literally compared to each other. Our findings also show that metaphors are part of ever-evolving processes following similar patterns even between communities and socio-technical phenomena.

- The application of metaphors for conceptual development in communities is influenced by community type (e.g., researcher or practitioner; open or semi-private) and metaphors' role (e.g., core or peripheral). The communities' socio-technical phenomena influence their selection of metaphors but have a negligible influence on their application of metaphors.

Second, what are the lesson learned for analyzing the application of metaphors for the OGD and TD communities?

- The boundary between metaphor and concept is blurry where some terms could be both, making them challenging to capture in both quantitative and qualitative analysis.

- The community structure influences how senders communicate to receivers but also acts as an entry barrier for newcomers to the community. It impacts the quantity and quality of empirical material.

- The CAOM framework provides analytical dimensions, enabling the exploration of metaphors' application in communal communication. It helped us bridge the use of quantitative and qualitative analysis.

- The quantitative analysis revealed when a metaphor had a high potential to influence the conceptual development of a community but also when it could have limited potential.

- The qualitative analysis explained the usage and trends of metaphors within the two communities. It was an essential part of understanding metaphor complexes.

8.1 Limitations and future research

Our research has some limitations and opens new avenues for future research. In our research, we followed the suggestions of Mays and Pope (2000) to improve our research's validity. We have used methodological triangulation, fair dealings, attention to negative cases, and lastly, clear exposition of methods of data collection and analysis. We also followed Creswell and Creswell (2017); Myers (2013); Saunders and Lewis (2012) to further ensure research quality. Our synthesis of the CAOM framework focused on understanding, as such it is possible that further previous research could be used to enhance the framework. We believe the framework gives sufficient understanding to analyze the application of metaphors. However, our research has mainly focused on public empirical sources (e.g., blogs and reports) with some interviews from the TD community. This approach excludes forums and events where practitioners and researchers could meet to discuss and develop metaphors. It is possible that these social gatherings are an important location for the application of metaphors. On the other hand, if so, their output should sooner or later be expressed in the texts we have analysed; allowing us to identify them indirectly. Moreover, we used word frequency analysis (WFA) and then a qualitative analysis (QA). WFA is limited in its ability to capture the context of used words. It is possible that some instances of non-metaphorical use have been counted. QA should have offset some of these problems by allowing us to study the metaphors in their context, but this correction is not included in the statistics. We believe the combination of WFA and QA has mitigated some limitations of each other and contributed to better validity of our research. In the end, we believe our research to have good quality, but is one step towards understanding the development of metaphors. It has opened for future research about (1) the application and combination of metaphors, (2) comparative, longitudinal studies for the application of metaphors in communities, and (3) metaphors as a method for knowledge development about socio-technical phenomena between practitioners and researchers.

Acknowledgements

[Blinded for peer-review]

References

Accenture Federal Services. (2018). *Decouple to innovate – how federal agencies can unlock it value & agility by remediating technical debt.*

Albano, C. S. (2013). Open government data: a value chain model proposal. In *Proceedings of the 14th annual in-*

ternational conference on digital government research (pp. 285–286).

Altayar, M. S. (2018). Motivations for open data adoption: An institutional theory perspective. *Government Information Quarterly*, *35*(4), 633–643.

Attard, J., Orlandi, F., Scerri, S., & Auer, S. (2015). A systematic review of open government data initiatives. *Government Information Quarterly*, *32*(4), 399–418.

Black, M. (1955). Xii-metaphor. In *Proceedings of the aristotelian society* (Vol. 55, pp. 273–294).

Boell, S. K., & Cecez-Kecmanovic, D. (2014). A hermeneutic approach for conducting literature reviews and literature searches. *CAIS*, *34*, 12.

Bonina, C., & Eaton, B. (2020). Cultivating open government data platform ecosystems through governance: Lessons from buenos aires, mexico city and montevideo. *Government Information Quarterly*, *37*(3), 101479.

Carrara, W., Chan, W., Fischer, S., & Steenbergen, E. v. (2015). Creating value through open data: Study on the impact of re-use of public data resources. *European Commission.*

Carrara, W., Fischer, S., Oudkerk, F., van Steenbergen, E., & Tinholt, D. (2015). *Analytical report 1: Digital transformation and open data.*

Charalampidou, S., Ampatzoglou, Apostolos, A., Chatzigeorgiou, & Avgeriou, P. (2017). Assessing code smell interest probability: a case study. *In Proceedings of the XP2017 Scientific Workshops*, 1–8.

Creswell, J. W., & Creswell, J. D. (2017). *Research de-*

sign: Qualitative, quantitative, and mixed methods approaches. Sage publications.

Cunningham, W. (1992). The wycash portfolio management system. *ACM SIGPLAN OOPS Messenger*, *4*(2), 29–30.

Custers, B., & Bachlechner, D. (2017). Advancing the eu data economy: Conditions for realizing the full potential of data reuse. *Information Polity*, *22*(4), 4291–309.

Davies, M. (1982). Idiom and metaphor. In *Proceedings of the aristotelian society* (Vol. 83, pp. 67–85).

Davies, T. (2011). Open data: infrastructures and ecosystems. *Open Data Research*, 1–6.

Dawes, S. S., Vidiasova, L., & Parkhimovich, O. (2016). Planning and designing open government data programs: An ecosystem approach. *Government Information Quarterly*, *33*(1), 15–27.

Dawkins, R. (2016). *The selfish gene.* Oxford university press.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review*, *14*(4), 532–550.

Fairbanks, G. (2020). Ur-technical debt. *IEEE Software*, *37*(4), 95–98.

Folmer, E., Reuvers, M., Quak, W., van den Broek, T., & van Veenstra, A. F. (2011). Walking the extra byte: A lifecycle model for linked open data. In E. Folmer, M. Reuvers, & W. Quak (Eds.), *Linked open data–pilot linked open data nederland* (pp. 95–111). Remwerk.

Fontana, F. A., Ferme, V., Zanoni, M., & Roveda, R. (2015).

Towards a prioritization of code debt: A code smell intensity index. *In 2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 16–24.

Gerrits, L. (2021). Soul of a new machine: Self-learning algorithms in public administration. *Information Polity*, *26*(3), 237–250.

Glucksberg, S. (1989). Metaphors in conversation: How are they understood? why are they used? *Metaphor and Symbol*, *4*(3), 125–143.

Guo, Y., & Seaman, C. (2011). A portfolio approach to technical debt management. *Proceedings of the 2nd Workshop on Managing Technical Debt*, 31–34.

Haack, S. (1988). Surprising noises: Rorty and hesse on metaphor. In *Proceedings of the aristotelian society* (Vol. 88, pp. 293–302).

Hamilton, A. (2000). Metaphor in theory and practice: the influence of metaphors on expectations. *ACM Journal of Computer Documentation (JCD)*, *24*(4), 237–253.

Handbook, O. D. (2015). *What is open data?* Retrieved 2020-08-31, from http://opendatahandbook.org/guide/en/what-is-open-data/

Harrison, T. M., Pardo, T. A., & Cook, M. (2012). Creating open government ecosystems: A research and development agenda. *Future Internet*, *4*(4), 900–928. Retrieved from http://www.mdpi.com/1999-5903/4/4/900/ doi: 10.3390/fi4040900

Hekkala, R., Stein, M.-K., & Rossi, M. (2018). Metaphors in managerial and employee sensemaking in an information systems project. *Information Systems Journal*, *28*(1), 142–174.

Imenda. (2014). Is there a conceptual difference between theoretical and conceptual frameworks? *Journal of Social Sciences*, *38*(2), 185–195.

Izwaini, S. (2003). A corpus-based study of metaphor in information technology. In *Proceedings of the workshop on corpus-based approaches to figurative language, corpus linguistics* (pp. 1–8).

Kaarst-Brown, M. L., & Robey, D. (1999). More on myth, magic and metaphor. *Information Technology & People*.

Kendall, J. E., & Kendall, K. E. (1993). Metaphors and methodologies: Living beyond the systems machine. *MIS quarterly*, 149–171.

Khayyat, M., & Bannister, F. (2017). Towards a model for facilitating and enabling co-creation using open government data. *Information Polity*, *22*(4), 211–231.

Kruchten, P., Nord, R. L., & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *Ieee software*, *29*(6), 18–21.

Lakoff, G., & Johnson, M. (2008a). *Metaphors we live by.* University of Chicago press.

Lakoff, G., & Johnson, M. (2008b). *Metaphors we live by.* University of Chicago press.

Lindman, J., Kinnari, T., & Rossi, M. (2015). Business roles in the emerging open-data ecosystem. *IEEE Software*, *33*(5), 54–59.

LNyhlen, S., & Gidlund, K. L. (2021). In conversation with digitalization: Myths, fiction or professional imagining? *Information Polity*, *Pre-press*(Pre-press), 1–11.

Magnusson, J., Juiz, C., Gomez, B., & Bermejo, B. (2018). Governing technology debt: beyond technical debt. *Proceedings of the 2018 International Conference on Technical Debt*, 76–84.

Mars, M. M., Bronstein, J. L., & Lusch, R. F. (2012). The value of a metaphor: Organizations and ecosystems. *Organizational Dynamics*, *41*(4), 271–280.

Maxwell, J. A. (2012). *Qualitative research design.* SAGE Publications Inc.

Mays, N., & Pope, C. (2000). Assessing quality in qualitative research. *Bmj*, *320*(7226), 50–52.

McBride, K., Olesk, M., Kütt, A., & Shysh, D. (2020). Systemic change, open data ecosystem performance improvements, and empirical insights from estonia: A country-level action research study. *Information Polity*, *25*(3), 377–402.

Morgan, G. (1998). Images of organization: The executive edition. *Thousand Oaks, CA*.

Myers, M. D. (2013). *Qualitative research in business and management.* Sage Publications Limited.

Mészáros, I. (1966). Metaphor and simile: Preliminary notes to a discussion of norms and values. In *Proceedings of the aristotelian society* (Vol. 67, pp. 127–144).

Oates, B. J., & Fitzgerald, B. (2007). Multi-metaphor method: organizational metaphors in information systems development. *Information Systems Journal*, *17*(4), 421–449.

ODI. (2019). *Data ecosystem mapping tool – the odi.* Retrieved 2020-11-09, from https://theodi.org/article/data-ecosystem-mapping-tool/#1561999285194-8defa99a-39a5

Ogden, C. K., & Richards, I. A. (1923). *The meaning of meaning: A study of the influence of thought and of the science of symbolism.* Harcourt, Brace & World, Inc.

Ortony, A. (1975). Why metaphors are necessary and not just nice. *Educational Theory*, *25*(1), 45–53.

Pollock, R. (2011). Building the (open) data ecosystem. *Open knowledge foundation Blog*, *31*.

Rapport, D., Gaudet, C., Constanza, R., Epstein, R., & Levins, R. (1998). Defining ecosystem health. *RAPPORT, D.; COSTANZA, R.; EPSTEIN, PR; GAUDET, C*, 18–33.

Saunders, M. N., & Lewis, P. (2012). *Doing research in business & management: An essential guide to planning your project.* Pearson.

Schedler, K., Guenduez, A. A., & Frischknecht, R. (2019). How smart can government be? exploring barriers to the adoption of smart government. *Information Polity*, *24*(1), 3–20.

Siebra, C., Oliveira, R., Seaman, C., Silva, F., & Santos, A. (2016). Theoretical conceptualization of td: A practical perspective. *Journal of Systems and Software*, *120*, 219–237.

Smith, G., & Sandberg, J. (2018). Barriers to innovating with open government data: Exploring experiences across service phases and user types. *Information Polity*, *32*(3), 249–265.

Stubblefield, W. A. (1998). Patterns of change in design metaphor: a case study. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 73–80).

Susha, I., Grönlund, A. k., & Janssen, M. (2015). Driving factors of service innovation using open government data: An exploratory study of entrepreneurs in two countries. *Information Polity*, *20*(1), 19–34.

Thibodeau, P. H., & Boroditsky, L. (2011). Metaphors we think with: The role of metaphor in reasoning. *PloS one*, *6*(2), e16782.

Tom, E., Aybuke, A., & Richard, V. (2012). A consolidated understanding of technical debt. In *European conference on information systems (ecis)* (pp. 1–13).

Verdecchia, R., Malavolta, I., & Lago, P. (2018). Architectural technical debt identification: The research landscape. *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 11–20.

Walsham, G. (1995). Interpretive case studies in is research: nature and method. *European Journal of information systems*, *4*(2), 74–81.

Zuiderwijk, A., Janssen, M., Van De Kaa, G., & Poulis, K. (2016). The wicked problem of commercial value creation in open data ecosystems: Policy guidelines for governments. *Information polity*, *21*(3), 223–236.

A Tables

Table 1: The number of articles in the selected outlets from the OGD and TD communities.

Outlet (Community)	**Round 1**	**Round 2**
Government Information Quarterly (OGD)	79	27
Information Polity (OGD)	36	10
International Conference on Electronic Government (OGD)	34	13
Journal of systems and software (TD)	20	15
International Conference on Technical Debt (TD)	40	31
International Conference on Software Engineering (TD)	6	4

Table 2: Selection criteria for practical literature

+/-	Selection Criterion	Reason
+	Published within 2015-2020	Capture the recent longitudinal development of the metaphors
+	Entailing 'Open Government Data' or 'Technical Debt'	Only include articles focusing on relevant for the study
+	Public and knowledge-oriented documents	Increase the chance for metaphorical use and developments
+	Only legitimate and credible organisations as publishers	Authenticity and credibility that helps to ensure the quality of our research (Myers, 2013)
+	Possible to download the content	Accessibility to the empirical material to practitioners

Table 3: Our metaphor group dictionary.

Open Government Data Metaphor Groups	
Ecosystem	Cultivate, Element, Ecosystem, Evolve, Climate, Evolution, Cycle, Cultivation, Environment
Silo	Silo, Access, Close, Unlock, Lock, Release, Capture, Share
Creature	Lifecycle, Body, Heart, Interact, Consume, Adapt, Sit, Smart, Maturity, Capability, Mature, Life cycle
Market	Provider, Competitive, Consumer, Asset, Competition, Provision, Trade, Supply, Market, Demand, Supplier
Industry	Survey, Resource, Oil, Extract, Landscape, Plot
Adventure	Journey, Myth, Obstacle, Struggle, Launch, Hero, Champion

Technical Debt Metaphor Groups	
Economics	Portfolio, Forecast, Capital, forudsi, forudse, progn, kost, penge
Emotion	Dark, Shadow, Visible, Density, Bad, Negative, synliggøre, synlighed, tæt, dårlig, negativ
Financial	Financial, Spend, Save, Income, Finans, spare, forbrug, gemt, gemme, indtægt, bankruptcy
Investment	Invest, Return, afkast
Loan	Repay, Principal, Interest, rente

Table 3 continued on next page.

Table 3 continued from previous page.

Person	Behavior, Immature, Lifecycle, Sprint, Smell, Brain, Grow, Retirement, Maneuver, Survival, Mature, Legacy, Inheritance, anatomy, heritance, infancy, candidate, Handsome, adfærd, cyklus, vokse, levetid, modern, hjern, arvet, bevæge
Purchase	Discount, Pay, Cost, omkost, betal
Repair	Shortcut, Hack, Self-fix, Maintain, Break, Fix, Refactor, genvej, vedlig, refak, brud, stykker, fiks

Table 4: Overview of patterns in the metaphor frequency analysis for the OGD and TD communities.

	OGD Community		***TD Community***	
Pattern	**Researcher**	**Practitioner**	**Researcher**	**Practitioner**
Bumper	Creature	Adventure, Silo		Person, Purchase, Repair
Climber	Silo, Adventure, Industry		Economics, Emotions	Emotion
Duner		Creature		
Quaker		Market, Industry	Purchase	
Rocket	Ecosystem, Market		Loan, Person, Repair	
Waver		Ecosystem		Economics, Loan

Table 5: Overview of the in-depth qualitative analysis of the ecosystem and technical debt metaphors.

		Open Government Data											
		Researcher						**Practitioner**					
	Year(20-)	*-15*	*-16*	*-17*	*-18*	*-19*	*-20*	*-15*	*-16*	*-17*	*-18*	*-19*	*-20*
	Documents	4	7	5	8	5	6	8	3	12	23	13	35
	Density	4.3ρ	28.4ρ	4.6ρ	5.4ρ	5.2ρ	87.5ρ	1.5ρ	9.7ρ	2.6ρ	13.7ρ	9.0ρ	8.0ρ
Usage	*Everyday*	75.0%	57.1%	60.0%	75.0%	80.0%	50.0%	100.0%	66.7%	66.7%	52.2%	46.2%	60.0%
	Syntax	25.0%	0.0%	20.0%	25.0%	20.0%	0.0%	0.0%	33.3%	33.3%	17.4%	23.1%	14.3%
	Tool	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	26.1%	30.8%	17.1%
	Develop	0.0%	42.9%	20.0%	0.0%	0.0%	50.0%	0.0%	0.0%	0.0%	4.3%	0.0%	8.6%
Expressi.	*Start*	0.0%	14.3%	0.0%	0.0%	20.0%	0.0%	12.5%	0.0%	0.0%	4.3%	0.0%	0.0%
	End	50.0%	14.3%	40.0%	50.0%	0.0%	33.3%	50.0%	0.0%	41.7%	17.4%	7.7%	25.7%
	Crumbles	0.0%	0.0%	0.0%	12.5%	20.0%	0.0%	25.0%	33.3%	50.0%	34.8%	38.5%	28.6%
	Mix	50.0%	71.4%	60.0%	37.5%	60.0%	66.7%	12.5%	66.7%	8.3%	43.5%	53.8%	45.7%
Complex	*Core*	0.0%	28.6%	0.0%	12.5%	20.0%	50.0%	0.0%	33.3%	0.0%	25.0%	15.4%	14.3%
	Supplementary	0.0%	14.3%	40.0%	12.5%	0.0%	0.0%	0.0%	33.3%	0.0%	12.5%	0.0%	0.0%
	Peripheral	50.0%	42.9%	40.0%	37.5%	60.0%	16.7%	50.0%	33.3%	75.0%	50.0%	76.9%	54.3%
	Insignificant	50.0%	14.3%	20.0%	37.5%	20.0%	33.3%	50.0%	0.0%	25.0%	12.5%	7.7%	31.4%

		Technical Debt											
		Researcher						***Practitioner***					
	Year(20-)	*-15*	*-16*	*-17*	*-18*	*-19*	*-20*	*-15*	*-16*	*-17*	*-18*	*-19*	*-20*
	Documents	9	8	4	13	8	7	13	7	8	23	41	8
	Density	64.2ρ	55.0ρ	59.0ρ	39.8ρ	62.4ρ	52.9ρ	11.8ρ	12.3ρ	5.8ρ	21.1ρ	7.1ρ	12.8ρ
Usage	*Everyday*	22.2%	12.5%	50.0%	30.8%	22.2%	57.1%	76.9%	42.9%	62.5%	65.2%	68.3%	50.0%
	Syntax	0.0%	25.0%	0.0%	0.0%	11.1%	28.6%	7.7%	14.3%	12.5%	21.7%	7.3%	12.5%
	Tool	0.0%	0.0%	0.0%	7.7%	0.0%	0.0%	0.0%	0.0%	12.5%	0.0%	0.0%	0.0%
	Develop	77.8%	62.5%	50.0%	61.5%	55.6%	14.3%	15.4%	42.9%	12.5%	13.0%	24.4%	12.5%
Expressi.	*Start*	0.0%	0.0%	0.0%	0.0%	0.0%	14.3%	0.0%	14.3%	12.5%	4.3%	4.9%	0.0%
	End	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	7.7%	0.0%	12.5%	0.0%	2.4%	0.0%
	Crumbles	33.3%	0.0%	0.0%	15.4%	33.3%	0.0%	53.8%	42.9%	50.0%	47.8%	63.4%	25.0%
	Mix	66.7%	100.0%	100.0%	84.6%	55.6%	85.7%	38.5%	42.9%	25.0%	47.8%	31.7%	75.0%
Complex	*Core*	77.8%	100.0%	75.0%	76.9%	88.9%	71.4%	46.2%	42.9%	12.5%	26.1%	14.6%	62.5%
	Supplementary	0.0%	0.0%	25.0%	7.7%	11.1%	0.0%	15.4%	0.0%	12.5%	21.7%	26.8%	25.0%
	Peripheral	22.2%	0.0%	0.0%	15.4%	0.0%	28.6%	23.1%	14.3%	50.0%	26.1%	26.8%	0.0%
	Insignificant	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	15.4%	42.9%	25.0%	26.1%	36.6%	12.5%

Figure Captions

Fig. 1: An anatomy of a metaphor.

Fig. 2: The research process.

Fig. 3: Practitioners use of metaphors within the OGD community.

Fig. 4: Researchers use of metaphors within the OGD community.

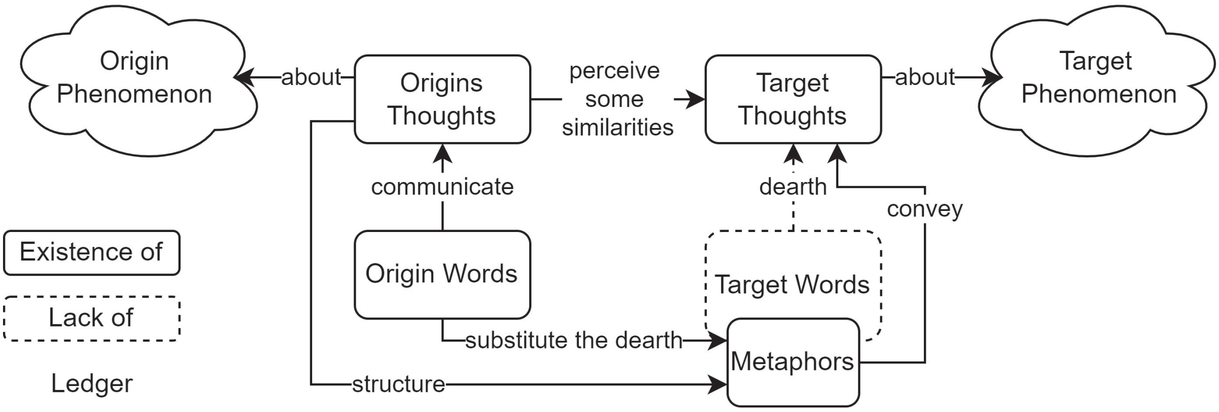
Fig. 5: Practitioners use of metaphors within the TD community.

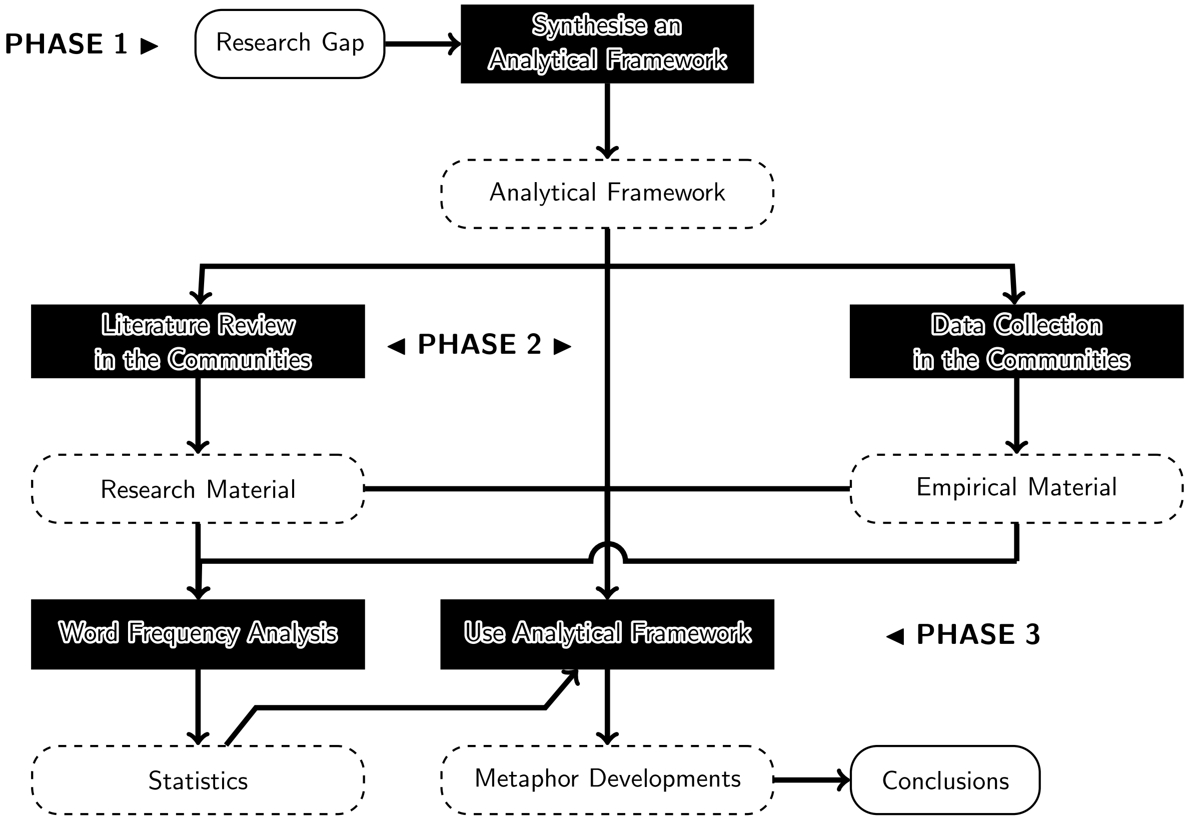
Fig. 6: Researchers use of metaphors within the TD community.

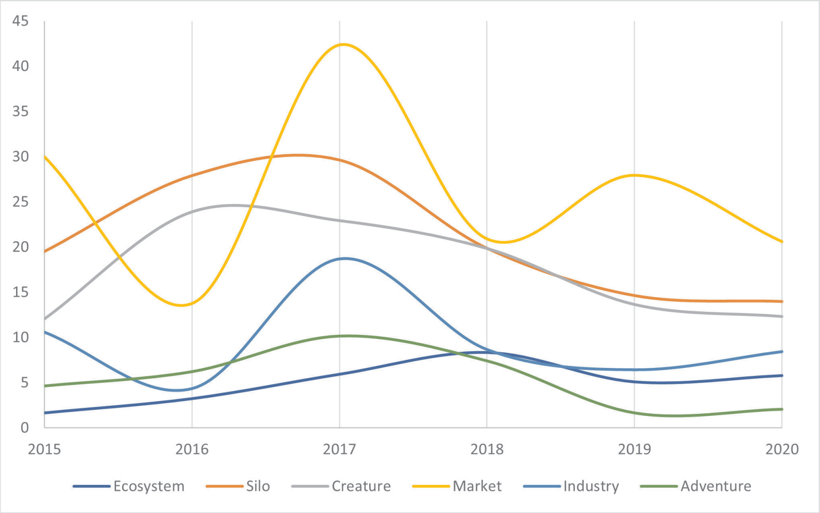
Fig. 7: Documents per actor over year for the OGD community.

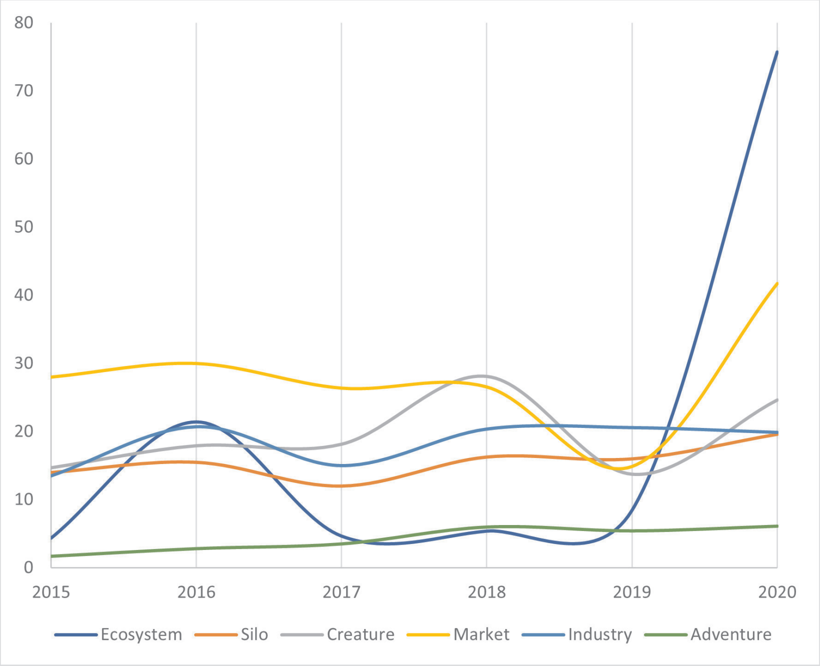
Fig. 8: Documents per actor over year for the TD community.

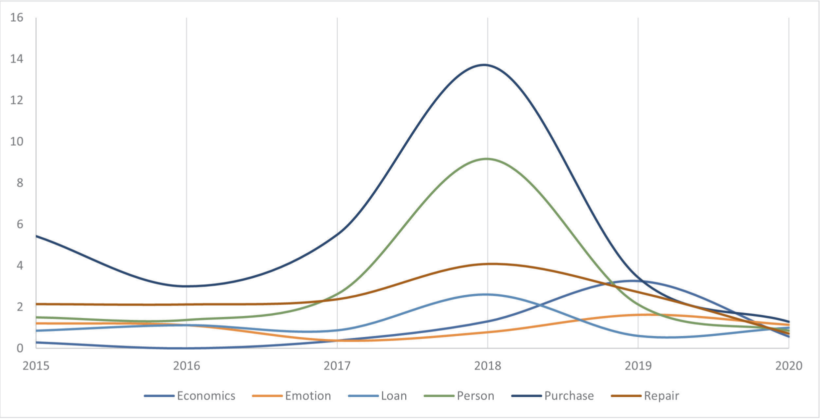
Figures

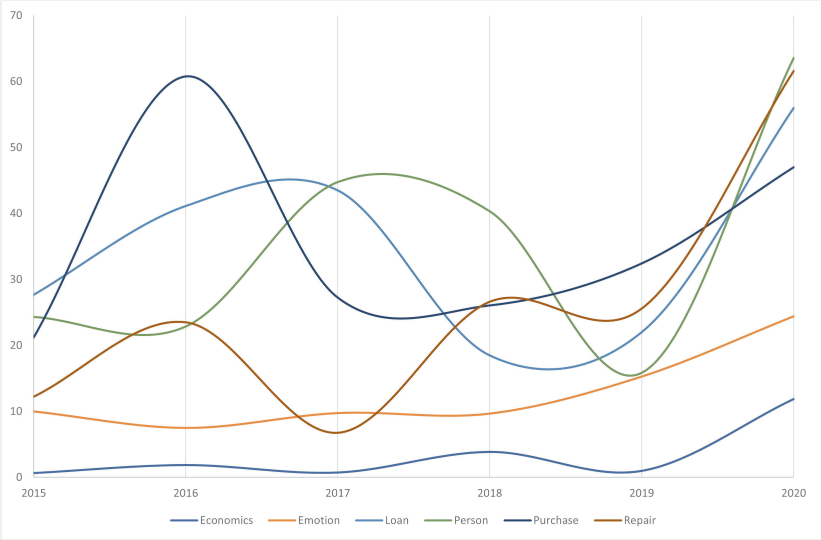

Origin Phenomenon
about
Origins Thoughts
perceive some similarities
Target Thoughts
about
Target Phenomenon
communicate
dearth
convey
Existence of
Origin Words
Target Words
Lack of
substitute the dearth
Metaphors
Ledger
structure

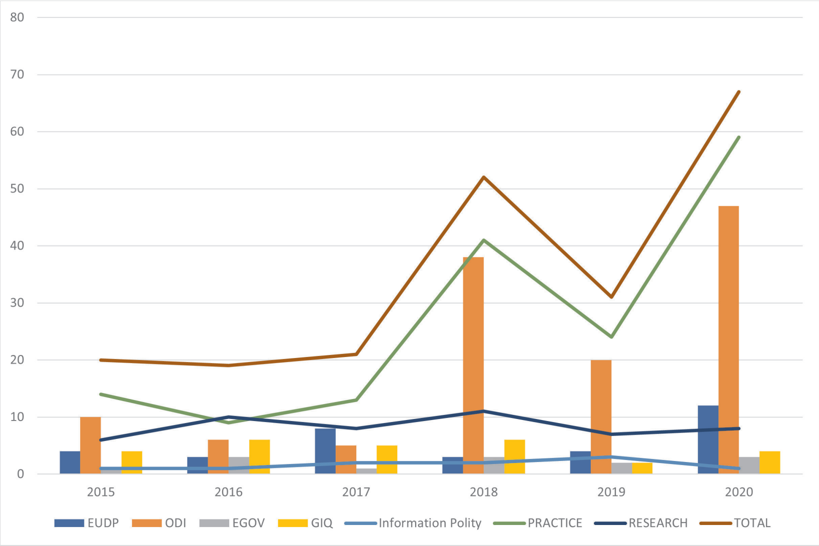

PHASE 1 ▶
Research Gap
Synthesise an
Analytical Framework
Analytical Framework
Literature Review
in the Communities
◀ PHASE 2 ▶
Data Collection
in the Communities
Research Material
Empirical Material
Word Frequency Analysis
Use Analytical Framework
◀ PHASE 3
Statistics
Metaphor Developments
Conclusions

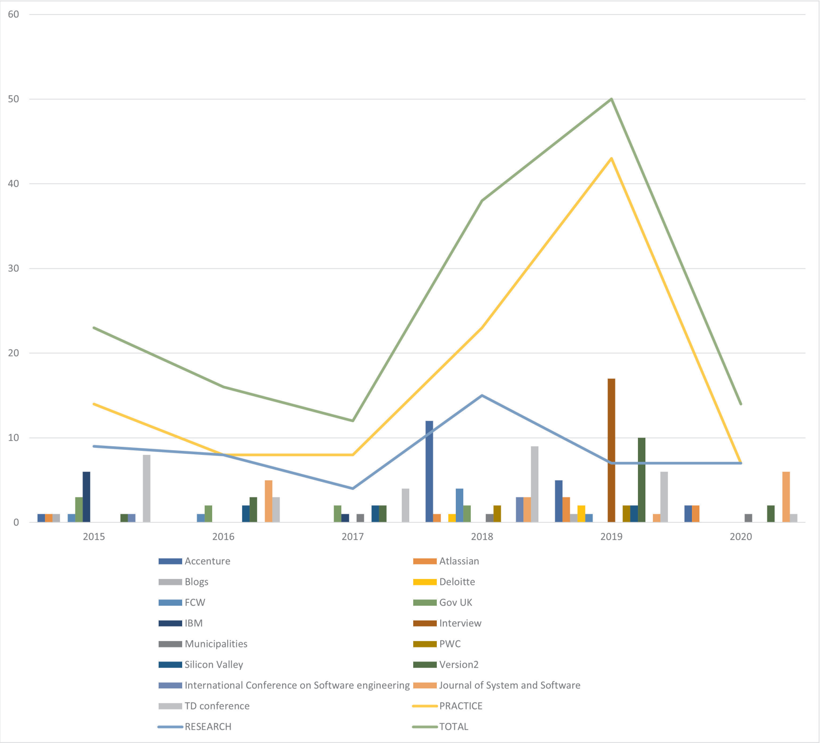

45
40
35
30
25
20
15
10
5
0
2015
2016
2017
2018
2019
2020
Ecosystem
Silo
Creature
Market
Industry
Adventure


80
70
60
50
40
30
20
10
0
2015
2016
2017
2018
2019
2020
Ecosystem
Silo
Creature
Market
Industry
Adventure


16
14
12
10
8
6
4
2
0
2015
2016
2017
2018
2019
2020
Economics
Emotion
Loan
Person
Purchase
Repair


70
60
50
40
30
20
10
0
2015
2016
2017
2018
2019
2020
Economics
Emotion
Loan
Person
Purchase
Repair


80
70
60
50
40
30
20
10
0
2015
2016
2017
2018
2019
2020
EUDP
ODI
EGOV
GIQ
Information Polity
PRACTICE
RESEARCH
TOTAL


60
50
40
30
20
10
0
2015
2016
2017
2018
2019
2020
Accenture
Atlassian
Blogs
Deloitte
FCW
Gov UK
IBM
Interview
Municipalities
PWC
Silicon Valley
Version2
International Conference on Software engineering
Journal of System and Software
TD conference
PRACTICE
RESEARCH
TOTAL

IT Portfolio management as a framework for managing Technical Debt

Theoretical framework applied on a case study

MILLE EDITH NIELSEN

SØREN SKAARUP

IT University of Copenhagen, Research Centre for Government IT, {miln, skaa}@itu.dk

Technical debt (TD) originally names the reoccurring phenomenon of shortcuts and quick fixes within IT development. TD saves time and resources in the short run but may cause problems and require additional resources in the long run. Managing TD is difficult, whether it is seen as a local phenomenon (residing in individual applications) or seen in a portfolio perspective. However, TD is not always caused by the individual developer and it does not always just affect one application. It may affect several parts of the IT portfolio; it may arise in the portfolio and it may be resolved by decisions on an IT-portfolio level. In this paper, we develop a theoretical framework that incorporates TD into IT portfolio management (ITPM). We apply the framework on a case study to explore how TD can be created, reside, resolved and managed in practice in a portfolio-perspective. This paper contributes with a framework integrating TD with ITPM. The paper also provides empirical insights on practice regarding ITPM and TD and highlights implications for research and practice.

CCS CONCEPTS • Applied Computing – E-government • IT Portfolio management • Technical debt management

Additional Keywords and Phrases: Technical debt, Technical debt management, IT portfolio management, IT portfolio framework, IT governance

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2021. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2021, Woodstock, NY. ACM, New York, NY, USA, 12 pages.

1 INTRODUCTION

IT development often takes place under a strict budget or timeframe, and the scope and the resources available may change during development. These restrictions and changes can lead to short-cuts or suboptimal solutions in order to save resources and deliver on time. Cunningham called this phenomenon "Technical Debt" (TD) [7]. TD has traditionally been understood as a short-term solution which needs to be addressed at later point in time to ensure software stability.

TD can be beneficial in the short run, however, if the debt is not managed it may pose a risk in the long run [7]. TD can have negative effects on IT maintenance, as it can become more time consuming to resolve the debt or develop the IT application further. TD can slow IT applications down and cause breakdowns, and it can bring entire organizations to a stand-still [7,27]. The costs and constraints introduced by IT can also be an impediment for business-development and growth [7,23]. TD is often seen as an aspect of individual systems [22] arising from choices made during system-development [7,22]. However, TD may also be created during IT maintenance [32] and may evolve (increase or decrease) during the application lifecycle [25].

TD has primarily been researched in the context of the private sector and in the software engineering field, thus leaving a gap for TD research in the public sector and from an eGovernment perspective [22]. TD is a serious issue in the public sector. A third of societal or business-critical IT systems in the Danish Government have been found to be in an inadequate condition [8]. Likewise, the Swedish National Audit Office found that 70% of their IT systems were outdated [34]. This shows the importance of understanding the creation and management of TD in the public sector. While the private and the public sector shares a constant drive to increase efficiency and effectiveness, to a large extent through the use of technology, there are also important differences in what drives change in the two sectors [6]. New political initiatives and priorities may require new development or changes in IT-systems within a short timeframe, and this may induce shortcuts and lead to the creation of TD in the systems affected and/or in other systems from where resources may be taken.

The research questions we seek to answer then are: *How can technical debt be conceptualized in an IT- portfolio-management perspective? And how is technical debt created and managed in the IT portfolio perspective in a government organization?* To answer this, we propose a theoretical framework combining IT portfolio management (ITPM) and TD management, and we apply this framework to a public sector case study to explore how TD is created and governed at a portfolio level in practice. This study makes a contribution to the e-Government literature by offering a tool for understanding TD management in an ITPM context, and by providing empirical insights into TD management practices in a government organization. The framework is also useful for IT-practitioners for increasing awareness of TD challenges in the IT portfolio. The study thus bridges and has implications for both practice and research and advances digital governance research and practice agenda [36].

The paper is structured as follows: First, we present related work on TD management, secondly, we synthesize the literature on IT portfolio management and present a theoretical framework. We then introduce our case study, and our methodological approach, before applying the framework on the case study. Finally, we discuss the application of the framework and the relation between IT portfolio and TD.

2 RELATED WORK: TECHNICAL DEBT MANAGEMENT

The term "technical debt" (TD) was coined by Cunningham in 1992 [7]. Since then, TD research has focused on further defining the concept and develop tools to identify and quantify TD items. This research has generally confirmed the negative effects of TD [22].

Rios et al [27] conducted a tertiary literature study on TD examining 13 secondary TD studies and 185 TD primary studies. Therefore, in our conceptualization of TD we include problems deriving from incomplete documentation, not following the IT-architecture, and from short term solutions which are not sustainable in the long run. To limit the consequences of the negative effects of TD, it is crucial to focus on the management of TD. Griffith, Taffahi, Izurieta, and Claudio's [4, p. 1016] define TD management as comprising "*the actions of identification, assessment, and remediation of TD throughout a software system*." From this definition TD management can appear to be limited to cover one software system at a time.

Avgeriou, Ernst, Nord and Kutchen [1, p. 40] find that "*… [what] the software technical-debt community has not discussed is that of technical debt crossing multiple disciplines. In a sense, technical debt is incurred in one discipline,* *but it is burdened and has to be repaid in another discipline*." For example management may make decisions that accumulate TD in applications, which then become the developers' task to solve [1].

Rios et al. [28] conducted a survey of 107 practitioners from the software industry to identify the experienced causes of TD. They identified the most frequent causes as: deadlines, inappropriate planning, lack of knowledge, non-adoption of good practices, ineffective project management, lack of qualified professionals, lack of experience, outdated/incomplete documentation, and lack of commitment. These causes are not restricted to the individual developer or application and includes IT-management and the management of the whole IT-portfolio among the sources of TD. They also indicate that TD may arise from the wider organization, e.g., when deadlines and resource-allocations are decided there.

Klinger et al [14] examines TD from an enterprise perspective, they find that "*enterprise technical debt occurs in the context of a larger portfolio*". For example, different non-technical stakeholders in the portfolio have an impact on TD, however, it is difficult for non-technical stakeholders to understand TD and their effect on it. [14].

TD can be introduced by changes outside the IT-development and -maintenance activities [1]. It may, for example, arise from the relation between IT applications and the infrastructure on which they are built and run. E.g., if the IT application requires a specific infrastructure which is no longer supported, or there is an increased use of the IT application so it exceeds what the infrastructure can deliver (known as infrastructure debt [27]). TD may also arise from changes in the business context of the organization or in the technological environment in which it operates [26]. In the business environment, several factors may affect TD, such as, the business's need for high system-reliability, user/customer satisfaction [26].

With their concept of "technological debt", Magnusson & Bygstad [19] add further dimensions to this growing set of factors that can influence the accumulation of TD. Applying institutional theory, they point to the factors of path-dependency (that previous decisions and established practices and principles may guide future decisions), lock-in (when a technology is embedded in standards, contracts, IT-staff and user-skills and in systems-integrations, making it more difficult to switch), and institutional logics (e.g., a focus on cost savings and standardizations). All these factors, while not strictly rooted in technical considerations, may influence decisions made about and around the systems.

Thus, a wide set of factors may influence TD, as TD occurs in the context of a larger portfolio [2]. There is, as indicated by Klinger [14], Rios et al [27] and by Magnusson and Bygstad [19], a need for a more holistic approach to TD and how to manage it, something which appears to be missing in the literature. Therefore, we find it relevant to view TD in a portfolio perspective, and to view the management of TD as part of the overall IT-portfolio management in order to benefit from collective resources within the organization.

3 METHODOLOGICAL APPROACH

This section presents the methodological approach of the paper. First, we describe how the framework was synthesized, then we describe the selection of the case and the case itself. Lastly, we present the data gathering techniques we applied.

3.1 Framework

We contextualize TD management in an ITPM framework based on a reading of IT Portfolio literature [12]. The result is a framework consisting of the core elements of the IT-portfolio, and the core management activities across the portfolio, including the management of TD. Thus, the framework could be extended with further portfolio elements and further activities. The purpose of applying a framework is to have it as a guidance for exploring, explaining and interpreting the empirical material [12]. A framework can be used as a mirror to the findings and the findings can be used to improve and adjust the framework and show its possible limitations. A theoretical framework draws on theory or concepts from a theory, to shed light on a particular phenomenon or research problem [12]. Therefore, the researcher first chooses the theory they want to guide the analysis and then they synthesize it into a framework. We choose to review the literature on IT portfolio and ITPM as presented in section 4. We develop a theoretical framework in which we apply a deductive approach in synthesizing the literature on ITPM. Secondly, we apply the framework on the empirical material (see section 5.1).

3.2 Case study

We conducted a case study [37] to explore the relation between IT portfolio and TD and between ITPM and TD management. The case study research approach is recommended when the researcher wishes to explain "how" or "why" a social phenomenon works [37], as it enables a detailed understanding of a phenomenon [9]. We have chosen a deviant case to obtain knowledge on an especially good practice [8], as this allowed us to observe good ITPM and how TD management is performed in the organization.

Agency X is a mature IT organization and considered among the best government organizations in Denmark in the development and operation of IT-systems. The agency has a large IT-portfolio containing internally developed as well as off the shelf applications. The agency has not employed their own developers, instead they have partnerships with software companies. The IT-development is caried out in a collaboration between the agency and software companies' developers placed onsite. Agency X works actively with ITPM; there are monthly meetings between the IT architects and top management, and a quarterly two-day workshop with the IT department, top management, and business managers. The latter is inspired by SAFe's[1] Program Increment (PI) planning, with a focus on planning of IT development and the interdependencies between projects and maintenance [30]. The purpose of these quarterly workshops is to plan the next couple of months of IT development and maintenance and to ensure dependencies between tasks, projects, and systems are taken into account and to mitigate any potential risks. The business managers are present in these workshops to help prioritize tasks, clarify needs and answer funding questions. These workshops primarily focus on internally developed systems. The PI planning format was implemented shortly before to the data collection toke place and the agency was working toward a final form.

Generalizing case studies can be problematic [37], although Flyvbjerg argues it is possible to generalize on the basis on a single case [8]. Our case selection is information-oriented and we have selected an extreme case, more precisely a good case [8]. in Agency X, which is considered among the most mature public sector IT-organizations in Denmark. We would expect that we in this case would be able to find TD managed at portfolio level.

3.3 Techniques for data collection

Data was collected applying several qualitative techniques: participatory observation, semi-structured interviews, and document analysis. The data collection took place from August 2019 to December 2019.

Participant observation [5] was carried out by the first author who was present at Agency X three full days a week during the data collection period. Additionally, the first author attended three of the quarterly PI workshops, in which the following months of IT development were presented and prioritized, and dependencies coordinated.

[1] The Scaled Agile Framework® (SAFe®) is a system for implementing agile practices at enterprise scale it includes structured guidance on roles and responsibilities [30]

This provided a rich insight into how ITPM took place. Conducting observations allows the researcher to address the say/do problem, where people may say one thing during interviews and do something else in practice [3]. These observations were captured through continuous notetaking.

Next, the first author conducted 15 semi-structured interviews [16] and two in situ interviews [4]. During the in situ interviews the interviewee performed their tasks while the interviewer asked about their practice. The interviews consisted of open-ended questions framed around the employee's collaboration process with the other employees maintaining and managing the IT system. From the interviews, we gained insights into the employees' perception and reflection of the maintenance processes and ITPM.

Together these techniques provide an in depth and detailed empirical insights. Particular, the participatory observation was beneficial in gathering rich insights. The observation notes were coded, together with the transcribed interviews. The empirical material was analyzed by applying the theoretical framework (section 4).

4 FRAMEWORK: IT PORTFOLIO MANAGEMENT

In this section we synthesize the IT-portfolio literature to establish a theoretical framework (figure 1) which can be used to investigate the creation and solution of TD. In this framework we propose to include TD management as a distinct portfolio-management activity. This framework will then be applied to an empirical case (section 5).

A "portfolio" can be defined as "*A collection of items grouped together to facilitate their efficient and effective management*" [23, p. 37]. Typically, a portfolio is comprised of a group of business-investments with something in common, e.g., contributing to the same overall organizational goal(s). A portfolio is a dynamic entity which has to be continuously managed, monitored and maintained as a set of interrelated assets/activities which *address the business needs* efficiently [13]. IT-portfolio management (ITPM) includes the management of *applications*, *infrastructure*, *projects* and people and their mutual dependencies [15,33]. Additionally, Verhoef [35] argues that *IT-operations* and maintenance are aspects of an IT portfolio.

The main goal of managing the IT-portfolio is to ensure that it supports the activities and goals of the business [18,20]. Thus, a key activity across the portfolio is to ensure that it is well aligned with business needs and goals. Changes in business-needs may be driven by changes in the business environment where, in the case of public sector organizations the political environment plays an important part [6,21].

Another key activity across the portfolio is to manage personnel-resources in a global perspective, which may be a challenging task [21]. Closely related to this is deciding what parts of the portfolio to operate, maintain and develop inhouse and what parts of these tasks to buy from outside vendors, including which applications to buy and which to develop internally [31]. These decisions are not only a matter of economics but are closely related with considerations of what resources and skills are, can and should be available in-house [29]. Thus, the management of "people" across the portfolio and the management of sourcing/procurement is closely related [29]. For the purpose of this framework, we have combined the two in what we call the *re-sourcing* activity.

The purpose of ITPM is to apply a holistic perspective on these sub-portfolios as one interconnected portfolio [10,15,24]. And based on our practical experience they are indeed interconnected. IT-applications are either bought or developed inhouse through IT-projects – a re-sourcing decision. The applications run on infrastructure that is either bought or operated in-house (a re-sourcing decision) and applications and infrastructure is maintained by IT-operations (which again can be outsourced or in-house)[2]. Changes in applications and infrastructures are

[2] This inhouse/outsourced distinction is a considerable simplification of a much more nuanced set of options [17]

(ideally) primarily driven by business needs which may in turn be driven by changes in the *business environment*, but changes may also be driven by changes in the external *technology-environment* [31]. In the technological environment, new technologies may make it possible to discard old systems, including the debt they may have accrued [26] or new technologies may reveal inadequacies in the existing technological setup [2].

As indicated by the discussion in the related work section, TD comes in many forms, and may be created in all parts of this portfolio and in the interplay between its elements [14,19,27]. Therefore, we find it relevant to include *TD management* in the ITPM framework. We combine these elements and present them in a theoretical framework as displayed in figure 1.

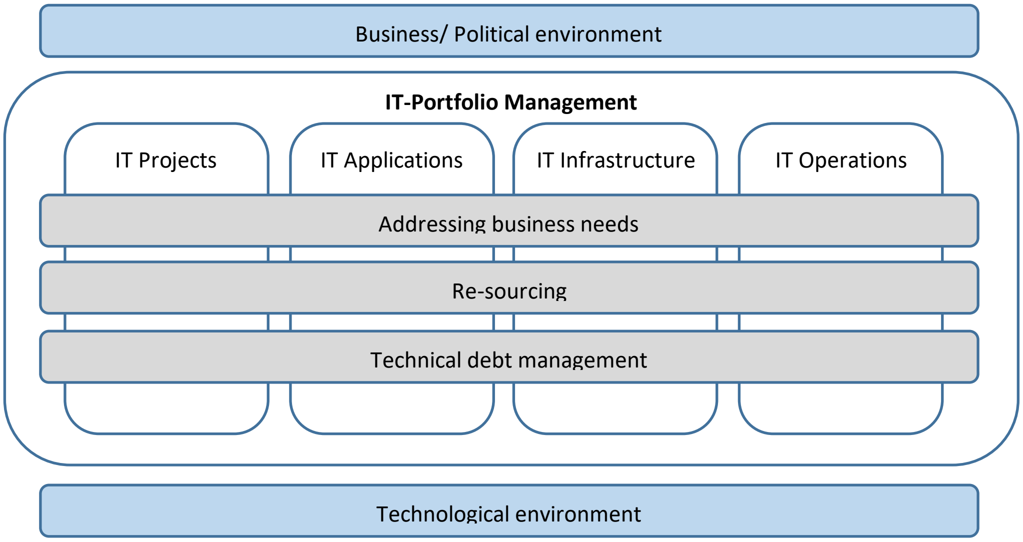

Business/ Political environment
IT-Portfolio Management
IT Projects
IT Applications
IT Infrastructure
IT Operations
Addressing business needs
Re-sourcing
Technical debt management
Technological environment

Figure 1: Theoretical framework of the components of an IT-portfolio (white boxes), key management activities across the portfolio (grey boxes) and the most important external factors affecting the portfolio (blue boxes)

5 FINDINGS

We apply the framework (figure 1) on the case of agency X to investigate if and how TD is created in the portfolio and to what extent the agency managed TD as part of their ITPM. Through our application of the framework, we find several examples of portfolio-related TD. Here, we present these examples and relate them to the framework. ITPM decisions lead to the creation of TD within the portfolio, however, portfolio related decisions also resolve some of the TD. Also, agency X have implemented tools and procedures to prevent and resolve TD on an IT portfolio level. We present five overarching findings: 1) TD exist in the IT portfolio, 2) TD may not always be created and reside in the application it is handled, 3) re-sourcing decisions impact and reveal TD creation, 4) TD is created by the change of TD needs, 5) successful TD management strategies which are implemented in Agency X.

5.1 TD resides, not in one application but in the application portfolio.

An older internally developed Content Management System (CMS) only partly supported the need for CMS functionalities that had been developed in Agency X. Therefore, several other CMS applications had been added to the application portfolio supporting different needs but with overlapping functionalities. This created unnecessary complexity in the application portfolio which constitutes a type of TD. This TD does not reside in the individual CMS applications but in the application-portfolio. The Agency had to decide whether they should continue with all of these CMS applications or if they should replace them with one off the shelf application. This would eliminate the TD arising from the complexity of having many CMS systems. Resolving the TD in this way would have re-sourcing effects across the portfolio. Operations would have to develop new integrations for the different business-applications to the new system, and to take care of the decommissioning of the old systems. In addition, if the new systems required different server-resources than the old systems, changes would have to be made in the infrastructure portfolio.

5.2 TD may not always be created and reside in the application where it is handled

TD in an application can be created by the way that application is used by other applications. A case handling component served several business applications. The teams developing the business applications did not always comply with the data formats required by the case-handling component due to lack of knowledge or lack of resources. This created performance problems for the case-handling component. The TD was in this case created and also resided in the business-applications, but the issues were seen as problems in the case-handling component. Therefore, it was left to the developers maintaining that application to fix the problems that arose by swiftly changing their priorities in order to clean up the data before it affected the citizens who relied on that data. TD created in business applications to support business needs created problems for operations because they had to change focus and prevent this TD from affecting citizens (another business need).

5.3 Re-sourcing decisions in one part of the portfolio creates or exacerbates TD in another part

A change was needed in a self-service application to improve the user-experience, however, needs in other parts of the portfolio resulted in the application being deprioritized and striped of its developers on several occasions. Every time a new developer came onboard, they had to be trained to work on the application. The lack of developers with knowledge of the application constituted a type of TD which was exacerbated by re-sourcing decisions, depriving the application of resources and making the change more costly and time-consuming than originally planned.

Another example is the case handling component mentioned in section 5.2. It was stripped of developer resources several times, as they were needed elsewhere. Shortly after these resources where removed errors started occurring in the system. As this affected both internal users and citizens, finding a solution became a priority. Once the component was prioritized and had developers assigned to it, problems where again addressed quickly, and the errors stopped.

These examples of TD were not created in the application or in the application portfolio but in the re-sourcing activities across the whole IT-portfolio. However, this was not a result of any overall prioritization of resources, but of ad-hoc decisions which focused on where the resources where needed, not on the consequences of moving them.

5.4 TD created by re-sourcing decisions and then transferred to operations

The case handling system was belatedly transferred from development to operations. It turned out that this transfer included a backlog of more than six hundred unresolved tasks of varying severity, partly with unknown status. This backlog constituted a significant amount of TD and had accumulated because the resources had been moved from the development-project too soon. It was now up to operations to clean up this backlog, but they were not given resources to do this. This example illustrates a tension in the re-sourcing going from project to operation.

5.5 Re-sourcing decisions reveal existing but unknown technical debt in the infrastructure

Re-sourcing-decisions may also reveal existing but unknown TD. An operations' provider decided to change the hosting of servers from a subcontractor to themselves. This change in hosting did not "create" TD as much as it revealed and actualized TD that existed but was not known; logic that should have been implemented at the application level and thus should not have been affected by the move turned out to be implemented elsewhere and would have to be re-developed for the new setup. Several applications turned out to be running older versions that would have to be updated and there was a lack of documentation which made it difficult and more costly to perform the move. This TD did not reside in individual applications but in the infrastructure setup.

5.6 Technical debt created by changes in business needs

Agency X is a key implementer of high prioritized government policies and also has a long tradition of being very sensitive to the (changing) needs of the citizens they serve. This drives change, sometimes rapid change, in the business needs and goals the IT-portfolio is expected to support.

5.6.1 Technical debt created by assumed business-needs

The old CMS application which we mentioned in section 5.1, was created to support an assumed business need of supporting multiple language versions of the texts it contained as well as a detailed version-history of the texts; however, these functionalities were never actually fully utilized. The consequence was an unnecessarily complicated code-based which was more difficult for developers to navigate and to perform maintenance and further developments on. This TD was not created by shortcuts in the development process but by business needs that never fully materialized. Had the business fully utilized this functionality, the relative complexity of the code would probably not have been considered problematic and there would be no TD here.

5.6.2 Technical debt created by changes in political priorities

Changes in the business environment may create or reveal TD in the IT-portfolio. In a political environment such as the one in which agency X operates, changes in political objectives and the need for fast implementation of new regulations can create shorten deadlines and lead to cutting corners and the creation of new TD. Sudden changes in priorities can lead the organization to postpone or give up on ongoing changes, which would have reduced TD, in order to free up resources to handle the new top priorities.

When a general election was coming up, Agency X had to take a possible change in government into account in their planning as it could change the political priorities in the areas where agency X operate. This in turn could have significant consequences for how resources where prioritized across the IT-portfolio. While some types of new regulations were created through lengthy process that allowed the agency ample time to plan and prioritize, other types appeared suddenly and unexpectedly e.g., as the result of late-night political compromises. This could mean that the entire IT-planning had to be redone. Not only could this lead to resources being shifted away from ongoing projects or operation, but also to this being done with such haste, that there was no time to document or consider the consequences. Not only could new TD be created but also new unknown TD. In addition, the deadline for the implementation of the new regulations, and consequently for the development and implementation of new IT, could be so tight, that development would have to begin before the details of the new regulation was finalized. Not only could this lead to resources being wasted because things would have to be redone, but this deadline could also lead to the creation of TD. These examples refer to addressing business needs (in form of the politicians), resourcing, IT projects and possibly operations.

5.7 Managing technical debt

The examples above show how TD was created in the IT-portfolio of agency X in a number of ways. However, we also observed how the agency was beginning to handle TD in a more systematic way and in a portfolio-perspective.

5.7.1 Bundling technical debt into projects to be prioritized with other projects

Agency X has a focus on managing TD. The head of the IT department encourage application managers to bundle TD tasks and make a small business case so these tasks can be addressed and prioritized alongside with IT development projects. One example was a project that arose from a TD problem, which affected several parts of the organization. The project solved the problem and was prioritized and funded like other IT projects. Thus, TD management appears to be in the process of becoming an IT portfolio activity as suggested by the framework. In this example, the activity covers IT projects and applications as the TD debt in the applications is reduced in the project.

5.7.2 Integrating technical debt resolution into development projects

We also saw examples of how Agency X resolved old TD during IT development. One example was in the development of new functionality for an existing system. The resolution of a list of older TD items in the system was integrated into the development project as a separate task and prioritized at the same level as the development of new functionality. Thus, the TD management of an application is addressed in the project.

5.7.3 Towards portfolio-focused re-sourcing planning

As we have seen, TD may be created in one part of the portfolio by resourcing decisions made in other parts. A continuous overview and management of resources used in projects and on maintenance across the portfolio can contribute to reduce existing TD as well as the risk of creating new TD. The overview facilitates the allocation of resources to new tasks in a timely manner, and it may to some extent prevent the striping of resources from unfinished tasks, or at least do this in a way where the consequences are considered, and any resulting TD is catalogued for later resolution.

The beginnings of such a holistic portfolio-wide planning process was seen in agency X in the shape of the Program Increment (PI) planning process. The PI planning is a reoccurring event in the SAFe framework, adapted to Agency X's needs [30].

In the PI planning sessions, a "dependency board" displayed the dependencies between the project-, maintenance- and operation-teams over time. Through this visualization of the dependencies, it would for example become apparent that within the same short time period several projects relied on the same application-team to deliver on some tasks. This realization caused the projects to reschedule and postpone these tasks, thus, freeing up the application-team to handle one dependency at the time. The potential consequences of not rescheduling could be not meeting deadlines, having to reschedule much later or creating TD. Rescheduling is not an easy task as a dependency could reach further than one step, it could be multiple steps through various actors. The dependency board illustrated the complexity and final goal of this trail of dependent tasks, also illustrating that scheduling the dependencies is important and they can be complex. This example demonstrates how PI planning involves resource planning (of projects and operations). In effect it also involves TD management because resource planning can prevent TD.

The organizers of the PI planning process stressed the importance of business owners being present in the planning sessions, to facilitate the resolution of any doubts that might arise regarding funding and priorities. The planning process also included the employees' own assessments of their workload and how thinly stretched they were, in order to catch any potential problems before they occurred. One possible shortcoming in the PI planning process could be that while it includes a resource overview of the internal employees it does not include external resources, which means that it does not include any development resources as these are all external in agency X. Prioritization of these resources are made elsewhere.

5.8 Summary

This case shows that even in a large IT-mature organization as Agency X, TD can be created unwittingly. This TD can result from more than code related decisions made by individual developers in individual applications and it can reside in the IT-portfolio rather than in individual applications. However, resolving this TD may still end up as a task for an application- or operations-team, even though it may be caused by another project, a different application, or decisions regarding, business needs, sourcing or infrastructure in other parts of the portfolio. Unfortunately, even if the application team or operations are aware of this TD, they rarely have the necessary resources to solve this debt on top of their other tasks. The political stakeholders provide another level of complexity in planning IT development, an unavoidable factor which political sensitive organizations such as Agency X has to include in their planning.

Agency X have already implemented several methods to address TD continuously and in an IT portfolio perspective. However, TD will never be truly eliminated nor should it, but it is important to recognize it, and its potential consequences, when it is created. Thus, the methods Agency X already has implemented are a crucial step to prevent and resolve TD. The methods include: 1) PI planning, 2) encouraging application managers to bundle TD tasks so they can be prioritized along with IT projects, and 3) including TD resolution in IT projects working on existing applications. Additionally, we encourage to adopt a portfolio focused approach to re-sourcing planning.

6 DISCUSSION, CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

In this paper, we explore how TD (management) can be conceptualized in an ITPM perspective, and how TD is created and managed in the IT portfolio perspective in a government organization. We synthesize the IT portfolio literature and propose a framework including the TD management activity, and we examine how TD is created and managed on a portfolio level in Agency X, a Danish government agency.

TD has primarily been researched in the private sector and focusing on the debt created in individual projects or applications [22]. Bygstad & Magnusson [19] and Klinger et al. [14] expand the TD metaphor to include a larger part of IT development.

We contribute with a theoretical framework which incorporates TD management into an IT portfolio and IT portfolio management (ITPM) context. The framework stimulates a different way of thinking of TD and how to manage it at a portfolio level. In addition, the framework can serve as a communication tool for addressing and communicating problem as described by Klinger et al. [14]. The framework only covers the core-elements and activities of an IT-portfolio and of ITPM and we suggest that future studies apply and further develop the framework. We confirm that it can be productive to see TD in a larger and more holistic perspective [14,19]. TD can be created by and in an IT portfolio and not only in source-code and within IT projects and that a narrow view on resource-planning in the IT-portfolio affects TD creation and management. Thus, we argue that it can be beneficial to integrate TD and TD management into ITPM.

Organizations can benefit from expanding their view of TD from something which is created and dealt with in the respective applications through development projects and primarily resides in the source-code, to something which is also created and live in the IT portfolio. They could also benefit from a holistic portfolio-wide approach to resource planning and from considering the full consequences of ad-hoc changes in resource allocations. This can enhance management's understanding of how TD emerges at an IT portfolio level, for example caused by short deadlines introduced by changes in political priorities which introduces new business-requirements to the portfolio. Among the benefits could be improved planning, more informed decisions when prioritizing resources in the IT portfolio, as well as improved productivity and quality. Agency X exemplify such ITPM activities in their PI planning, as well as when they bundle TD tasks into TD-projects so they can prioritize along with other projects and when they include TD resolution during IT development of existing applications. Better TD management could also be achieved by careful documentation of TD that may be created when withdrawing resources from a project or by extending deadlines when the resources are partly removed. Or by addressing TD during IT-development projects, rather than "exporting" TD-issues to maintenance and operations which rarely has the resources to deal with it.

The analysis of the case shows that the framework developed here can be useful for understanding TD in a portfolio perspective. However, the framework does not, by design, include all possible sub-portfolios of the overall IT-portfolio, nor all possible ITMP activities, only those that were considered the most important in the literature. A broader set of sub-portfolios and activities could possibly help in identifying even more portfolio-related sources of TD and provided a more detailed understanding of the sources and remedies identified in the empirical case. This we leave for future studies.

ACKNOWLEDGMENTS

This study is funded by a Research Centre Government IT, which is a collaboration between the IT University of Copenhagen, the Danish Digitization Agency, and the self-owned institution ATP. The Research Centre's external funding partners were not involved in the research presented herein, or its dissemination. Furthermore, while Agency X has fact checked the article, the Agency has not funded the article nor part taken in analyzing or writing the article.

REFERENCES

[1] Paris Avgeriou, Neil A. Ernst, Robert L. Nord, and Philippe Kruchten. 2016. Technical Debt. ACM SIGSOFT Software Engineering Notes 41, 2: 38–41. https://doi.org/10.1145/2894784.2894800

[2] B. Bahli and S. Rivard. 2003. A validation of measures associated with the risk factors in technology outsourcing. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, HICSS 2003. https://doi.org/10.1109/HICSS.2003.1174792

[3] Jeanette Blomberg and Mark Burrell. 2012. An Ethnographic Approach to Design. In Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications (3rd ed.), J. A. Jacko (ed.). Boca Raton, FL.

[4] Keld Bodker, Finn Kensing, and Jesper Simonsen. 2004. Participatory IT design: designing for business and workplace realities. MIT Press.

[5] Matthew Brannan and Teresa Oultram. 2012. Participant Observation. In Qualitative Organizational Research: Core Methods and Current Challenges, Gillian Symon and Catherine Cassell (eds.). Sage Publications, 296–305.

[6] John Campbell, Craig McDonald, and T Sethibe. 2009. Public and private sector IT Governance: identifying contextual differences. Journal of Information Systems 16, 2: 5–18.

[7] Ward Cunningham. 1992. The WyCash portfolio management system. ACM SIGPLAN OOPS Messenger 4, 2: 29–30. https://doi.org/10.1145/157710.157715

[8] Danish Ministry of Finance. 2017. Regeringens kasseeftersyn på it-området. Copenhagen, Denmark.

[9] Bent Flyvbjerg. 2006. Five misunderstandings about case-study research. Qualitative Inquiry. https://doi.org/10.1177/1077800405284363

[10] Chip Gliedman and Adam Grown. 2004. Defining IT Portfolio Management: Holistic IT investment Planning. Forrester Research.

[11] Isaac Griffith, Hanane Taffahi, Clemente Izurieta, and David Claudio. 2015. A simulation study of practical methods for technical debt management in agile software development. In Proceedings - Winter Simulation Conference. https://doi.org/10.1109/WSC.2014.7019961

[12] Sitwala Imenda. 2014. Is There a Conceptual Difference between Theoretical and Conceptual Frameworks ? Is There a Conceptual Difference between Theoretical. 8923, January. https://doi.org/10.1080/09718923.2014.11893249

[13] Jimmi Kellerman and Patrik Löfgren. 2008. Application Portfolio Management - A Framework for Application Destiny Determination. University of Gothenburg.

[14] Tim Klinger, Peri Tarr, Patrick Wagstrom, and Clay Williams. 2011. An enterprise perspective on technical debt. In Proceeding of the 2nd working on Managing technical debt - MTD '11. https://doi.org/10.1145/1985362.1985371

[15] R Kumar, H Ajjan, and Y Niu. 2008. Information Technology Portfolio Management: Literature Review, Framework, and Research Issues. Information Resources Management Journal 21, 3. https://doi.org/10.1039/TF9696500098

[16] Steinar Kvale. 2008. Doing Interviews. Sage Publications.

[17] Mary Lacity, L. Willcocks, and David Feeny. 1996. The Value of Selective IT Sourcing. Sloan Management Review 37, 3: 13–25.

[18] Juha Lemmetti. 2016. Construction of Enterprise Architecture in Discourses Within the Public Sector. Electronic Government: Proceedings of the 15th IFIP WG 8.5 International Conference, EGOV 2016, 287–298.

[19] Johan Magnusson and Bendik Bygstad. 2014. Technology debt: Toward a new theory of technology heritage. ECIS 2014 Proceedings - 22nd European Conference on Information Systems: 0–15.

[20] James D. McKeen and Heather A. Smith. 2010. Developments in practice XXXIV: Application portfolio management. Communications of the Association for Information Systems 26, 1: 157–170. https://doi.org/10.17705/1cais.02609

[21] Jeppe Agger Nielsen and Keld Pedersen. 2014. IT portfolio decision-making in local governments: Rationality, politics, intuition and coincidences. Government Information Quarterly 21: 467–483. https://doi.org/10.1016/j.giq.2014.04.002

[22] Mille Edith Nielsen, Christian Østergaard Madsen, and Mircea Filip Lungu. 2020. Technical Debt Management: A Systematic Literature Review and Research Agenda for Digital Government. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12219 LNCS: 121–137. https://doi.org/10.1007/978-3-030-57599-1_10

[23] Ariadi Nugroho, Joost Visser, and Tobias Kuipers. 2011. An empirical model of technical debt and interest. Proceedings - International Conference on Software Engineering: 1–8. https://doi.org/10.1145/1985362.1985364

[24] Joe Peppard. 2003. Managing IT as a portfolio of services. European Management Journal 21, 4: 467–483.

[25] Narayan Ramasubbu and Chris F. Kemerer. 2014. Managing technical debt in enterprise software packages. IEEE Transactions on Software Engineering 40, 8: 758–772. https://doi.org/10.1109/TSE.2014.2327027

[26] Narayan Ramasubbu, Chris F. Kemerer, and C. Jason Woodard. 2015. Managing technical debt: Insights from recent empirical evidence. IEEE Software 32, 2: 22–25. https://doi.org/10.1109/MS.2015.45

[27] Nicolli Rios, Manoel Gomes de Mendonça Neto, and Rodrigo Oliveira Spínola. 2018. A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. Information and Software Technology 102, June: 117–145. https://doi.org/10.1016/j.infsof.2018.05.010

[28] Nicolli Rios, Rodrigo Oliveira Spínola, Manoel Mendonça, and Carolyn Seaman. 2018. The most common causes and effects of technical debt: First results from a global family of industrial surveys. International Symposium on Empirical Software Engineering and Measurement, October: 1–10. https://doi.org/10.1145/3239235.3268917

[29] Vital Roy and Benoit A. Aubert. 2002. A Resource-Based Analysis of IT Sourcing. Data Base for Advances in Information Systems 33, 2: 29–40. https://doi.org/10.1145/513264.513271

[30] Scale Agile Inc. 2021. PI Planning. Retrieved April 19, 2021 from https://www.scaledagileframework.com/

[31] Hans Jochen Scholl. 2006. Electronic government: information management capacity, organizational capabilities, and the sourcing mix. Government Information Quarterly 23, 1: 73–96.

[32] Carolyn Seaman and Yuepu Guo. 2011. Measuring and Monitoring Technical Debt. https://doi.org/10.1016/B978-0-12-385512-1.00002-5

[33] Daniel Simon, Kai Fischbach, and Detlef Schoder. 2010. Application Portfolio Management—An Integrated Framework and a Software Tool Evaluation Approach. Communications of the Association for Information Systems 26. https://doi.org/10.17705/1cais.02603

[34] The Swedish National Audit. 2019. Föråldrade it-system – Hinder för en effektiv digitalisering. Stockholm.

[35] C. Verhoef. 2002. Quantitative IT portfolio management. Science of Computer Programming 45, 1: 1–96.

[36] Charalabidis Yannis, Cunha Alexandra Maria, and Sarantis Demetrios. 2020. Foreword. In Proceedings of the 12 th International Conference on Theory and Practice of Electronic Governance, i–xxi.

[37] Robert K Yin. 2018. Case Study Research and Applications: Design and Methods. Sage Publications.

Stakeholder influence on technical debt management in the public sector:

An embedded case study

Abstract

Technical debt (TD) entails the shortcuts and unsuitable choices made during the development or maintenance of an IT system, which can result in negative consequences such as inefficiency and instability of the IT system. Digitalizing the government has led to the development of numerous IT systems which must be maintained to prevent decay, standstill, additional costs, and a decline in software quality. Previous studies on TD have primarily focused on the private sector, while TD in the public sector has largely been ignored. Therefore, this case study investigated TD management in relation to two IT systems in a Danish agency. Through participant observations and in-situ interviews we studied actual TD behavior, while stakeholder theory combined with a categorization of TD types and activities served as our theoretical lens. Thus, our study (1) identifies the stakeholders influencing an agency's TD management, (2) maps stakeholders' actions, and (3) identifies stakeholders' influence on TD. We found that TD extends beyond the influence of software developers and is also influenced by the behavior of several non-technical stakeholders, e.g., the European Parliament. We offer practical recommendations for TD management based on these findings.

Keywords: Technical debt, technical debt management, Stakeholder theory, e-government, digital government

1. Introduction

Technical debt (TD) was introduced as a concept by Cunningham (1992) to explain the impact of suboptimal solutions made during the development of IT systems to cut costs or time. TD can make it more time-consuming for software developers to perform maintenance and further develop an IT system. Furthermore, TD can affect an IT system's performance by making it slower or by causing breakdowns (Rios, de Mendonça Neto, & Spínola 2018). Therefore, TD management (TDM) is essential to reduce TD's potential negative impact on IT systems' efficiency, stability, and costs (Alves, 2016; Ribeiro, 2016).

We follow Rios et al.'s (2018) definition of TD: "The concept of Technical Debt (TD) contextualizes problems faced during software evolution considering the tasks that are not carried out adequately during software development" (p. 117). TD fulfills all the requirements of a theory (Gregor, 2006). The TD literature *describes* TD types and how it is visible (Rios et al., 2018), as well as *explains* and *predicts* why and how TD is created. Moreover, it briefly *prescribes* that if TD is repaid, the potential negative consequences will not occur. The concept of TDM is defined by Griffith, Taffahi, Izurieta, and Claudio (2014, p. 1016) as comprising "the actions of identification, assessment, and remediation of TD throughout a software system."

TD is a relevant research topic for the digital government field because digital government research focuses on how the public sector delivers services to citizens and businesses further innovates and develops government services using emergent technologies (Lindgren, Madsen, Hofmann & Melin, 2019; Lindgren, Melin & Sæbø, 2021). TD management is required to deliver digital services and ensure that the emergent technologies work as they build upon existing IT systems. Without proper TD management, government organizations cannot provide and develop digital services for citizens and businesses. TD is a structural barrier to digital government (Wilson & Mergel, 2022). Lack of TD management can be critical for digital government transformation (Eom & Lee, 2022; Omar, et al., 2020) and implementation of new technologies such as big data (Guendueza, Mettler & Schedler, 2020), Internet of Things (Janssen & Helbig, 2018), AI and Machine Learning (Mikalef, Patrick, et al., 2021), and blockchain (Ølnes, Ubacht & Janssen, 2017).

This rapid digitalization has resulted in increased TD in the public sector. For instance, a study by the Danish Ministry of Finance found that a third of societal or business-critical IT systems in the Danish government were in inadequate condition (2017). Examples of TD from the report include out-of-date or non-existing documentation, lack of IT security, and the need to upgrade outdated software and hardware. Similarly, the Swedish National Audit (2019) found that 70% of the IT systems in Sweden's government agencies were outdated. TD is thus an important topic for digital government research because it can prevent public organizations from benefiting from digitalization (Nielsen, Madsen & Lungu, 2020).

One of the goals of Digital Government is to increase efficiency, which makes TD particularly relevant for Digital Government research because TD can hinder efficiency gains and increase costs (Panagiotopoulos, Klievink & Cordellac, 2019; Eom & Lee, 2022; Omar, Weerakkody & Daowd, 2020). Despite TD's relevance to the public sector, most TD studies have been conducted outside of the digital government field, with researchers from software engineering and information systems focusing instead on IT systems in the private sector (Nielsen et al., 2020). However, the theory of TD enables and encourages researchers to examine the processes surrounding IT systems rather than focusing on an isolated IT system (Berenguer et al., 2021; Nielsen & Skaarup, 2021).

Therefore, we conduct a case study in a Danish agency analyzing TD and TDM in relation to two IT systems. To capture the surrounding processes, we employ observation, in-situ interviews, and document analysis (Kvale, 1997; Blomberg & Burrell, 2012; Brannan & Oultram, 2012). We apply stakeholder theory (Freeman, 1987; Scholl, 2001) combined with TD (Rios et al., 2018) as a theoretical lens through which to interpret and analyze our case data. We chose this theoretical frame during our analysis because the case data was closely associated with the organizational values (Rose et al., 2018), stakeholder behavior, stakeholder stance (Freeman, 1987; Blair & Whitehead, 1988), TD types, and TDM activities of the agency in question (Rios et al., 2018). Our study is guided by the following research question: *How do stakeholders influence technical debt and technical debt management in a public agency?*

By combining ST and TD, we follow Janowski and Janssen's (2015) recommendation for digital government researchers to offer both theoretical contributions and practical recommendations. Our study offers insights into how stakeholders affect TD. (1) We employ research methods suitable for problem-solving and the public sector (section 4); (2) advance knowledge in the field (section 6.1); and (3) make recommendations for practitioners on how to prevent TD (section 6.2).

The remainder of the paper is organized as follows. First, we present related work on TD and identify the gaps that this paper addresses. Secondly, we introduce our theoretical lens. Third, we describe the case setting and the methodological approach. Fourth, we present our findings on how stakeholders influence TD and TDM. Fifth, we discuss the implications for research and practice. Finally, we present our conclusions, outline the limitations of the study, and offer suggestions for future work.

2. Related work

TD as a concept was introduced by Cunningham in 1992. He sought to communicate the consequences of TD to managers in the financial sector. Cunningham explained the effects of TD: "Shipping first-time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite" (Cunningham, 1992, p. 30). Left outstanding, TD may lead to inefficiency in maintaining and developing IT systems, decreased performance, and increased risk of breakdowns. Unmanaged, it can bring entire engineering organizations to a standstill as accumulated TD makes it challenging to develop and maintain IT. Therefore, organizations must manage their TD (Cunningham, 1992).

The TD literature is growing rapidly, and researchers have made significant contributions in identifying and measuring TD. However, several knowledge gaps remain with regard to the management of TD. Nielsen et al. (2020) conducted a systematic literature review of 49 primary TDM studies published between 2017 and 2020. They found that TDM studies tend to reinvent the wheel; previous TD studies are limited as they mostly confirm TD's negative effects on IT development and inventing tools and models that can measure specific types of TD (Nielsen et al., 2020). Nielsen et al. (2020) also identified a knowledge gap: researchers have studied TDM indirectly through interviews, surveys, code comments, and source code but have not observed the actual behavior surrounding TDM. Furthermore, previous TD research has primarily focused on the private sector, leaving TD in the public sector unexplored (Nielsen et al., 2020).

Avgeriou, Ernst, Nord, and Kutchen (2016) argued that those in management are responsible for the decisions that incur TD in IT systems: what "the software technical-debt community has not discussed is that of technical debt crossing multiple [work] disciplines. In the sense, technical debt is incurred in one discipline, but it is burdened and has to be repaid in another discipline" (Avgeriou et al., 2016, p. 40). While managers may cause TD, it is the responsibility of developers to solve the problems it creates. Yli-Humo et al. (2016) used ST to identify TD stakeholder groups. They concluded that addressing TD requires that the developers first increase awareness of TD and then apply more advanced TDM techniques.

Rios et al. (2018) observed a lack of applied theory in a tertiary literature review of TD research encompassing 185 primary and 13 secondary studies. They identified three research directions: TD identification, the concept of TD, and TDM. Moreover, they synthesized the literature into a TD landscape, mapping out TDM activities, strategies, tools, and 15 different TD types. These TD types identified by Rios et al. (2018) include the following: design debt, code debt, architecture debt, test debt, documentation debt, defect debt, infrastructure debt, requirement debt, people debt, build debt, process debt, automation test debt, usability debt, service debt, and versioning debt. The macro TDM activities were (1) those designed to *prevent*

TD; (2) those designed to *identify TD*; (3) those designed to *monitor TD* during the evolution of a project/ IT system; and (4) those designed to *pay TD* promptly (i.e., eliminate it). We will apply and extend this categorization by Rios et al (2018) in our analysis. Previous TD research has applied TD types and categorization to map TD types with TD causes, TD effects, and preventive and payment actions (Rios et al., 2020; Ramac et al.; 2022, Freire et al.; 2020; Pérez et al., 2021).

Klinger et al. (2011), meanwhile, conducted a study with IT architects and found that "the decision to acquire technical debt is not made by technical architects, but rather by non-technical stakeholders who cause the project to acquire new technical debt or discover existing technical debt that wasn't previously visible" (p. 35). Klinger et al. (2011) also state that future studies should focus on TD stakeholders. Furthermore, stakeholders are important in digital government studies as they influence the IT development in the government, e.g., digital government implementation (Ashayea & Irani, 2019). To the best of our knowledge, previous research has not investigated stakeholders' influence on TD in the public sector.

To summarize this section, Table 1 displays the three identified knowledge gaps in the extant TD literature, their importance, and how we will address these gaps in our study.

Table 1. Gaps in the TD literature

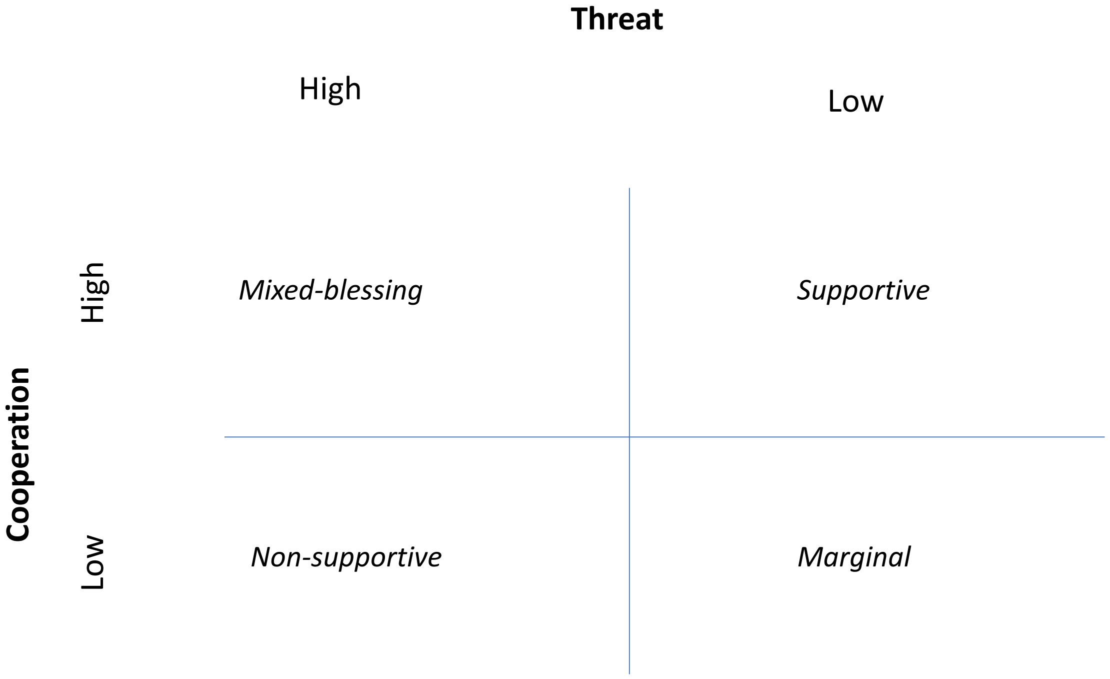
Gap	**Why is it important?**	**How do we address it?**
(1) Lack of knowledge on non-technical stakeholders	Klinger et al. (2011) find that non-technical stakeholders influence TD, but do not classify this group further.	We extend Klinger et al.'s (2011) study by identifying additional stakeholder groups, their actions, and the consequences on TD.

(2) Lack of TD research from the public sector	While TD exists in the public sector, it has mostly been studied in the private sector. Findings and recommendations from the private sector may not be applicable in the public sector.	We explore TD in the public sector in order to identify TDM strategies suitable for this specific context.
(3) Lack of knowledge on actual TD behavior	The existing knowledge on TD is based on interviews, experiments, and code analysis. Actual TD behavior may differ from what is stated in interviews.	We combine observations and in-situ interviews to gain insights into stakeholders' actual TD behavior (Blomberg & Burrell, 2012).

3. Theoretical lens: Stakeholder Theory (ST)

Freeman (1984) developed ST to increase company revenue in the private sector. He argued that organizational profits are higher when stakeholders are satisfied. An essential part of ST is the definition of a stakeholder: "A stakeholder in an organization is (by its definition) any group or individual who can affect or is affected by the achievement of the organization's objective" (p. 25). ST focuses on the firm's internal stakeholders (e.g., employees and managers) and external stakeholders (e.g., customers and suppliers). Freeman argued that "you must take your stakeholders into account in a systematic fashion" (p. 48)—identifying them, their actions, and their influence—to manage effectively. Stakeholders have different interests and goals that can be difficult to navigate when developing and maintaining IT systems. These competing priorities can lead to shortcuts, abandoned development, and similar problems that are likely to result in TD.

Blair and Whitehead (1988) operationalized stakeholder theory for analytical purposes by distinguishing the stakeholders' stance (Scholl et al., 2007) as displayed in Figure 1. They developed a diagnostic typology of stakeholders' potential to either cooperate with or threaten the organization. The typology comprises four distinct stakeholder categories: (1) mixed-blessing (high cooperation and high threat), (2) supportive (high cooperation and low threat), (3) non-supportive (low cooperation and high threat), and (4) marginal (low cooperation and low threat) (Blair & Whitehead, 1988).



		Threat	
		High	Low
Cooperation	High	*Mixed-blessing*	*Supportive*
	Low	*Non-supportive*	*Marginal*

Fig. 1. Blair and Whitehead's (1988) typology of stakeholder categories

Scholl (2001) introduced ST to the digital government field and identified how the theory could benefit "public-sector managerial decision-making" (Scholl, 2001, p. 745). Flak and Rose (2005) further explored the implications of applying ST in digital government studies. They highlighted an issue that results from transferring the theory from a private- to a public-sector context. Increasing profit is an important private-sector value, but does not apply in the public sector. Rose et al. (2018) contributed to the research on ST in digital government by identifying four values relevant to the public sector: professionalism, efficiency, service, and engagement. Although previous research had mapped stakeholder roles within digital government (Axelsson, Melin & Lindgren, 2013; Reinwald & Kraemmergaard, 2012), Rose et al. (2018) recommended future studies to better define "the specific roles of various stakeholders" (p. 372).

ST has been used to study TD and TDM, but only in a private-sector context. Yli-Huumo, Maglyas, and Smolander (2016) and Klinger et al. (2011) applied it in examining how a software team manages TD. They distinguished between technical and non-technical stakeholders and found that both these groups may influence TD. Yli-Huumo et al. (2016) contributed to the literature on TD by mapping stakeholders to specific TD activities.

We seek to build on this knowledge of TD by exploring stakeholders' influence on TD in a public-sector setting. To do this, we combine core concepts from ST with Rios et al.'s (2018) TDM activities in our analytical framework.

4. Method

Our study employed a qualitative and explorative approach. We conducted an embedded case study, which is suitable when the researcher seeks to explain how a social phenomenon works (Yin, 2018). An embedded case study is a case study that "may involve units of analysis at more than one level. This occurs when, within a single-case (the first level), attention is also given to a subunit or subunits (second level)" (p. 51). Specifically, we conducted an embedded case study with two subunits to explore TD on both an individual IT systems level and a holistic IT development level (Yin, 2018). The following section contains a description of the case setting, data collection, and data analysis.

4.1 Case description

Agency X is a mature IT organization, considered to be among the best government organizations in Denmark in terms of the development and operation of IT systems. The agency does not have its own developers. Instead, application development is carried out through a partnership between the agency and the software companies' developers placed on-site. Agency X also develops IT systems for other authorities. These authorities cover the costs, define the needs of the system, and are responsible for the underlying legislation, while Agency X sets the requirements for, develops, and maintains the IT system. In this way, these external systems become part of Agency X's IT landscape and use its back-end systems.

Internally developed IT systems can more easily be altered compared to off-the-shelf applications (Al-Ibraheem & Ruël, 2009). Hence, stakeholders are more likely to influence TD on these IT systems. We applied critical sampling to identify the two IT systems where the stakeholders' influence was the most evident (Yin, 2018). We chose the IT systems that we refer to as TEXT and REPORT. TEXT serves as a content management system (CMS) for the other IT systems, while REPORT offers a platform where citizens can report specific activities; the casework is handled by other authorities. An application manager and external developers are assigned to each of the two systems.

4.2 Data collection

We wanted to explore how stakeholders influenced TD. The first author gathered empirical data on-site from August to December 2019. Conducting case studies using multiple sources of evidence ensures that the research is of a high quality (Yin, 2018). Previous studies on TD have not studied actual TD behavior (Nielsen et al., 2020). Therefore, we conducted on-site interviews, in-situ interviews, and participant observations, as displayed in Table 2.

Table 2. Overview of the empirical studies

Technique	**Frequency**	**Amount of data**

Interview w/ application manager	4	210 minutes
Interview w/ architect	4	258 minutes
Interview w/ developer	4	223 minutes
Interview w/ law graduate	1	53 minutes
Interview w/ user representative	2	86 minutes
Interview (in situ) w/ user representative	1	35 minutes
Interview (in situ) w/ user	1	73 minutes
Total number of interviews	17	937 minutes
Participant observations		
On-site observation	3 days/week 5 months	13.555 words in field notes
2-day workshops	3	3.193 words in field notes
Document analysis	30 documents	

We conducted 15 semi-structured interviews at Agency X. We applied a purposeful sampling and interviewed only the employees that were part of the daily management of the IT systems (Kvale, 1997). The interviews were framed around the processes that the employees used to collaborate with others in maintaining and managing the IT system. From the interviews, we gained insights into the employees' perceptions of and reflections on the maintenance processes. The two in-situ interviews were conducted with a user and a user representative of the IT systems (Blomberg & Burrell, 2012). The interviewees performed their routine tasks on the IT system while the interviewer asked them about their actions and reflections. This gave us insights into the challenges that IT systems pose to their users and the users' reactions to these challenges.

The participant observations were conducted by the first author, who was present at the organization three days a week for five months. Additionally, she attended three of the agency's quarterly meetings, during which the following months of IT development were coordinated. She had been acquainted with Agency X before data collection began and therefore was able to blend in and take on the role of the participant as an observer (Brannan & Oultram, 2012). This also meant employees who were not interviewed initiated casual conversations with her on the research topic. They shared their experiences, thoughts, and concerns. This led to valuable insights on how other authorities and the Parliament influenced IT development. Both the observations and these conversations were captured as field notes observations (Blomberg & Burrell, 2012). The interviews were transcribed and, together with the field notes and the documents detailing tasks and system documentation, imported to MAXQDA 2020 (VERBI Software, 2019), where the coding took place.

4.3 Data analysis

Initially, we wanted to explore how and why TD arises. Our understanding of the phenomenon was sharpened through an open analysis, on-site data collection, and discussions with academic colleagues, before we conducted an open-ended exploration of the main themes and patterns of the interview transcripts, field notes, and documents (Glaser & Strauss, 1967). During the initial coding of the empirical material, we detected a pattern in how responsibility for prioritizing TD tasks was taken by or delegated to specific employees in Agency X. Moreover, we identified both internal and external stakeholders who wielded influence over TD management activities (Rios et al., 2018). Therefore, we chose to combine TD with ST as a theoretical lens to capture how stakeholders influence TD. We recorded the material using the new analytical lens of ST and through Rios et al.'s categorization of TDM macro activities. However, we added the category *create TD* to cover activities involved in creating TD (Tom, Aurum & Vidgen, 2013) because Rios et al.'s (2018) categories encompass the management of TD rather than the entire TD lifecycle. Furthermore, during our iterative coding, we expanded the *monitor TD* category to include the creation of workarounds and the postponement of dealing with TD (Glaser & Strauss, 1967).

We incorporated three dimensions of ST in our analytical strategy. First, we identified the stakeholders in IT systems, working holistically and including the external and internal stakeholders that we encountered (Freeman, 1984). Second, we explored the stakeholders' actions or interactions relating to the selected IT systems (Freeman, 1984). Third, we highlighted the TD consequences of stakeholders' actions using Rios et al.'s (2018) categories (prevent TD, identify TD, monitor TD, and pay TD), along with our new category, create TD.

5. Findings

In this section, we present the results of our analysis of stakeholders' influence on TD. First, we identify the stakeholders in the TEXT and REPORT systems related to TD. Next, we map them, their actions, and their influence on TD. Then we compare the stakeholders of the TEXT and REPORT systems and explore how the differences in stakeholders in the two systems affected TD. Lastly, we provide a high-level overview of the stakeholders and their influence on TD.

5.1 Stakeholder analysis of TEXT

The following section describes TEXT and its TD stakeholders. The Appendix contains a general stakeholder description.

5.1.1 System description of the TEXT system

The TEXT system is a content management system (CMS) that was built in 2012 as part of a new type of IT system. The vision was to have several IT systems instead of one large mainframe. Back-end IT systems like TEXT were introduced to reuse as many of the existing functionalities of the IT systems as possible. Thus, when the front-end IT system needs specific general functionality, it uses the back-end systems—in this instance, to pull text. This setup allows the users to quickly edit the text without the assistance of developers. The developer inserted a reference to the actual text in the TEXT system instead of hardcoding the text in the front-end system. Should the TEXT system have a breakdown, all the IT systems using the TEXT system would be filled with these references instead of actual text. A breakdown would therefore be critical for Agency X, although fortunately the TEXT system appears to run without such difficulties. We analyzed the TD types in the TEXT system and found seven of them (Rios et al., 2018). Additionally, we found that incorrect usage of the system blocked some of its functions. Table 3 shows how TEXT's TD challenges lie in over-complex code, insufficient processes, and disappointing usability, and offers examples of the potential negative consequences of each of these TD types.

Table 3. TD types identified in the TEXT system and their respective consequences

TD types	**Description**	**Consequence**
1) Code debt	Overcomplex code as half of the system was not used.	Requires more time for the developer to maintain and develop the system as it is difficult to navigate
2) Architecture debt	Incorrect and non-uniform usage of the system	Difficult to navigate the system and maintenance is challenged
3) Test debt	Lack of testing	Deployment of a new version removed previously established functionality
4) Requirement debt	The system is slow	Long response time when the user interacts with the system
5) People debt	Change of developers and few engaged users	Challenge the process of prioritizing TD tasks and address the users' needs

6) Process debt	Some manual processes could be made automatic	Reloading text is done manually, which requires additional resources
7) Usability debt	Some commands cause the system to timeout	Unable to conduct certain commands and long waiting time for the user

5.1.2 Stakeholder mapping of the TEXT system

We identified seven internal stakeholders and two external stakeholders that influenced TEXT's TD (Fig. 2). The internal stakeholders were as follows: (1) other IT systems' teams; (2) IT projects' teams that were integrated into the TEXT system to store the text snippets there; (3) the expert users that work with the TEXT system and ensure that it is comprehensible and actionable; (4) application managers who manage the maintenance of the organization's IT systems and their further development; (5) the IT architects who plan the overall architecture of the systems and ensure that future solutions fit the framework; (6) the IT operations staff who maintain the servers on which the IT systems run; and (7) the board of directors that prioritizes projects and resources. The two external stakeholders, meanwhile, are (1) the software companies that develop IT systems with the agency and (2) the Digitization Agency, which sends quality requirements for the board of directors to address.

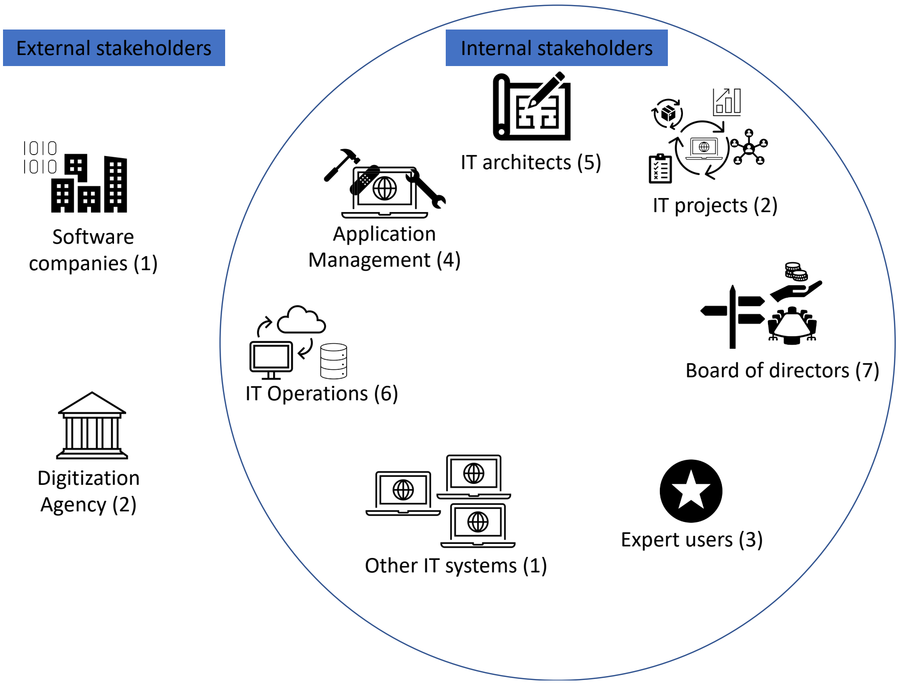


Fig. 2. The Internal and External Stakeholders of the IT System TEXT

We further analyzed the stakeholders' stance in terms of whether they posed a threat to the accumulation of TD and in terms of their cooperation level (Fig. 3) (Blair & Whitehead, 1988). Two of the stakeholders are considered a *mixed blessing* while three are considered *supportive*. We classify the teams working on other IT projects and other IT systems as *non-supportive* because their use of the TEXT system makes it more challenging to maintain the system. Only IT operation is identified as a *marginal* stakeholder.

We find that stakeholder actions can lead to a cascade of TD types. Some stakeholders can cause both TD creation and reduction, but this is not necessarily limited to a specific TD type, which means the TD types and stakeholders cannot be mapped directly.

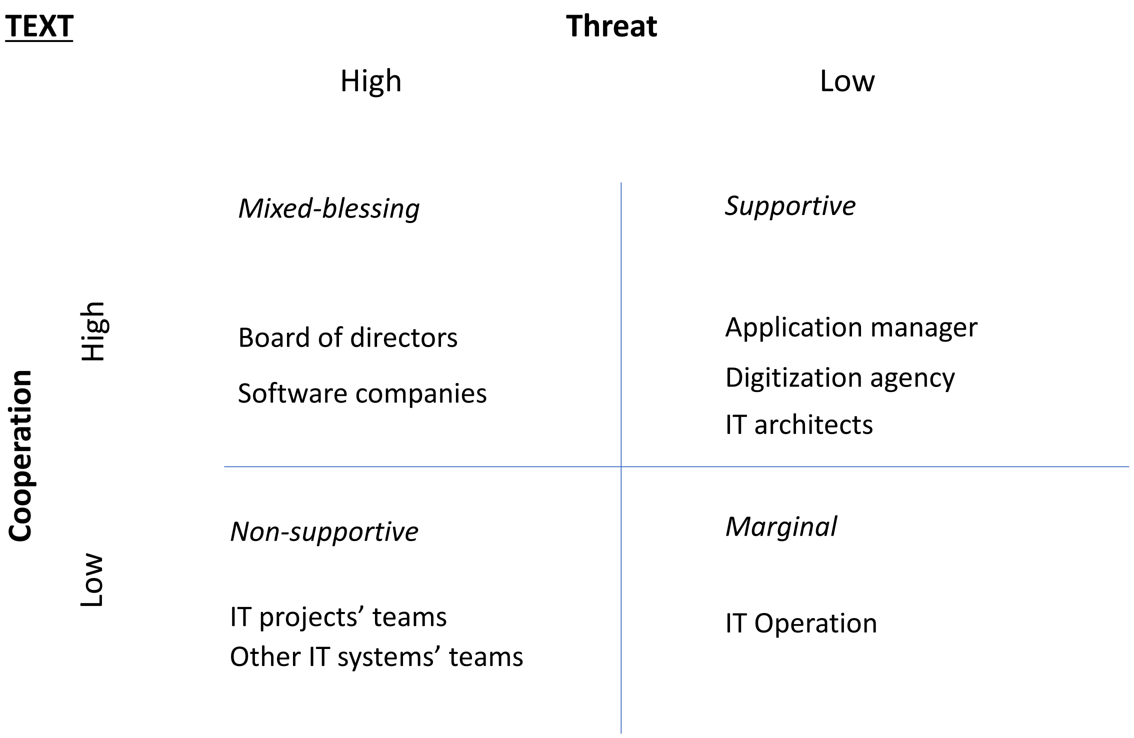


Fig. 3 Stance of TEXT's stakeholders

5.1.3 Stakeholders' actions and their TD consequences in the TEXT system

We mapped the identified stakeholders, the stakeholders' actions, and the TD consequences in Table 4. For example, the application managers (column 1) perform actions (column 2) that have TD consequences (column 3). We categorized the TD consequences according to Rios et al. (2018), as described in Section 4.3.

Table 4. A Mapping of Stakeholders' Actions and TD Consequences in the TEXT System

Stakeholder	**Description of actions**	**Consequence**
Application managers	Test the system, prioritize tasks, communicate business needs, and explore possible solutions, plan future maintenance on the system	Identify TD, pay TD, monitor TD, prevent TD

Board of directors	Determines priorities, makes architectural decisions, and allocates resources to projects that are given higher priority	Pay TD, create TD
Expert users	Provide feedback, prioritize tasks	Identify TD, pay TD, monitor TD
IT architects	Design the system architecture, approve solution descriptions, provide counsel on prioritization of technical tasks	Identify TD, pay TD, monitor TD, prevent TD
IT operations staff	Maintain the servers on which the systems run, escalate server capacity and power	Identify TD, monitor TD
Other IT systems'/IT projects' teams	Uses back-end systems	Create TD—if they do not follow the data formats and structure
Digitization Agency	Sets quality requirements for authorities' IT systems	Identify TD, pay TD
Software companies	Code systems, provide descriptions of possible solutions and feedback	Create TD, pay TD, identify TD

5.1.4 Stakeholder consequences on TD in TEXT: An example

The following example from the TEXT system involves several stakeholders and different types of TD consequences. The Digitization Agency (stakeholder) set quality requirements for the authorities (including Agency X). It required that all societal and critical applications be evaluated by the respective authorities (identify their TD). The authorities were then required to make an action plan to remediate issues identified in the evaluation (pay the TD), which included Agency X and the TEXT system. The TEXT system became a focal point of the board of directors (stakeholder). The IT architects (stakeholder) identified the different possibilities for paying TD, not only for the TEXT system but also for other CMS systems at Agency X. Consequently, the board of directors decided to centralize the CMS functionalities in one application. Thus, the TD in the TEXT system became obsolete and was thereby paid. The Digitization Agency, with its quality requirements, required Agency X to identify TD and pay TD. The TD in this case was paid as the board of directors decided to terminate the TEXT system and combine all of the CMS systems into one. Thus, this consequence ties into all of the TD types and helps to reduce TD.

5.1.5 Overview of the stakeholders and their influence

Table 5 maps the internal and external stakeholders of TEXT and their influence on TD. The most frequent consequence of stakeholder influence was the *payment of TD*; seven stakeholders' actions had this consequence. The least frequent consequence caused by the actions of stakeholders was the *prevention of TD*. Application managers and IT architects were the only stakeholders whose actions *prevented TD*.

Table 5. TEXT Stakeholders' Influence on TD

Stakeholder	**Prevent TD**	**Create TD**	**Identify TD**	**Monitor TD**	**Pay TD**
Application managers	X		X	X	X
Board of directors		X			X
Expert users			X	X	X
IT architects	X		X	X	X

IT operation team		X	X	
IT systems'/ IT projects' teams	X			
Digitization Agency		X		X
Software companies	X	X		X

5.1.6 Remarks on the TEXT system

In the TEXT system, we observed that the software developers were not alone in creating TD. Other stakeholders influenced which tasks software developers worked on, when they did so, and whether the work *created TD*. Most of the stakeholders *paid TD* and *identified TD* but only two *prevented TD*. After TEXT was built, further IT development was limited, although expert users were despondent towards the system. TEXT supports only text snippets, but Agency X's CMS needs exceeded this basic functionality, which is confirmed by the number of other CMS solutions in their IT landscape. Thus, TD can be created by a lack of IT development. The debt then continues to grow until a stakeholder ensures it is paid or the system is replaced. The consequences of the Digitization Agency's actions show that external quality requirements are beneficial for managing TD in public organizations.

5.2 Stakeholder analysis of REPORT

The following section describes the REPORT system and its TD stakeholders. The Appendix contains a general stakeholder description.

5.2.1 System description of REPORT

The REPORT system offers a platform where citizens can report specific activities, thus providing information that is then used by authorities to conduct casework. REPORT was developed on behalf of a different authority, Authority Z. Legally, Authority Z is responsible for providing the service to citizens and the authorities who need it to perform their casework. Should the legislation become obsolete, the IT system would be shut down. Because Authority Z is responsible for REPORT, Authority Z provides funding for the maintenance and development of the IT system. We analyzed the TD types in the REPORT system and found seven of them. Additionally, we found that short deadlines can lead to TD creation, because short deadlines cause fast-paced coding and low prioritization of maintenance. Table 6 illustrates how REPORT's TD challenges relate to a lack of maintenance (tests, documentation, and framework updates).

Table 6. TD types identified in the REPORT system and their respective consequences

TD types	Description	Consequence
Architecture debt	The IT system is not properly integrated into the existing IT landscape	Increases the need for maintenance
Test debt	Lack of testers and test cases	Can hinder development and lead to errors in the system
Documentation debt	Documentation is lacking, needs updating, consolidation, and clean-up	New employees and software vendors lack access to proper documentation
Infrastructure debt	Programming framework needs to be upgraded	The new versions provide security and functionality features

Requirement debt	Fast-paced requirements settings	Lead to hasty solutions which can be erroneous
Process debt	Data extractions could be automated	Instead, they are ordered with short deadlines and have high priority
Automation test debt	Tests have not been updated and maintained	Makes it difficult to identify errors in the system

5.2.2 Stakeholder mapping of the REPORT system

We identified four internal stakeholders and seven external stakeholders for the REPORT system's TD (Figure 4). The internal stakeholders were (1) application managers who maintain the IT systems and supervise their further development; (2) IT Operation responsible for maintaining the servers; (3) IT architects who plan the overall system architecture and ensure that future solutions fit the architecture; and (4) law graduates who ensure that the IT system adheres to the legislation. The seven external stakeholders were (1) software companies hired to build and maintain the IT system in collaboration with Agency X; (2) citizens using the services the authorities provide; (3) the EU Parliament, which sends out directives for governments to follow (e.g., the GDPR); (4) the Danish Parliament, which creates and changes legislation that the system supports or adheres to; (5) Authority Z, which serves as the funder and task setter; (6) authorities that act as collaborators and who contribute to and receive data from the IT system; and (7) authorities that act as users, performing casework using the IT system.

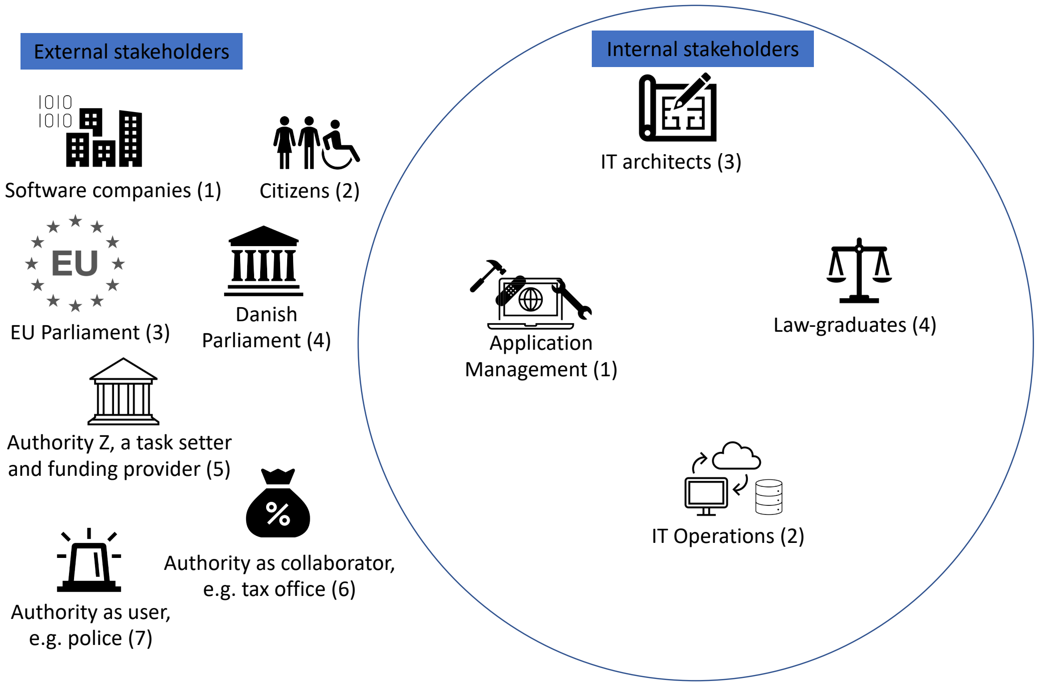


Fig. 4. The Internal and External Stakeholders of REPORT

We further analyzed the stakeholders' stance regarding whether they are a threat to the accumulation of TD or whether they cooperate to reduce TD (Fig. 5). Four of the stakeholders are considered a *mixed blessing*, five are considered *supportive*, and only IT operation was considered a *marginal* stakeholder. We found the citizens to be *nonsupportive* because they test the system by inserting incorrect data, which later has to be corrected.

5.2.3 Relation between stakeholders' stance and TD types in the REPORT system

Some stakeholders can cause TDM activities that create or reduce TD, but their actions are not necessarily targeted to a specific TD type. For example, the short deadlines for tasks imposed by the Danish or European Parliament can cause the agency to draw up requirements too quickly, meaning that they are not properly checked, tested, or documented (requirement debt, test debt, test automation debt, and documentation debt). Furthermore, these short deadlines can postpone automizing of data extractions, which the external authorities request using a short deadline (process debt).

REPORT **Threat**

Cooperation	High	Low
High	*Mixed-blessing* EU parliament Danish parliament Authority as task setter Software companies	*Supportive* Application manager Authority as collaborator Authority as user IT architects Law graduate
Low	*Non-supportive* Citizens	*Marginal* IT Operation

Fig. 5 Stance of REPORT's stakeholders

5.2.4 Stakeholders' actions and their consequences on TD in the REPORT system

Table 7 presents our mapping of the REPORT systems' stakeholders, their actions, and the consequences of these actions on TD. For example, law graduates (column 1) prioritize tasks, communicate other authorities' needs, and ensure compliance with legislation (column 2). These actions can *prevent TD* and *pay TD* (column 3).

Table 7. Stakeholders' Actions and Their Consequences on REPORT

Stakeholder	**Action**	**Consequence**
Application managers	Test the system, prioritize tasks, communicate the business's needs, explore possible solutions, plan future maintenance on the system	Identify TD, pay TD, monitor TD, prevent TD

Law graduates	Prioritize tasks, communicate other authorities' needs, ensure compliance with legislation	Prevent TD, pay TD
IT architects	Design the system architecture, approve proposed solutions, give counsel on the prioritization of technical tasks	Prevent TD, pay TD, monitor TD, identify TD
IT operations staff	Maintain the servers on which the systems run, escalate server capacity and power	Identify TD, monitor TD * Prevent system breakdowns
Citizens	Feedback, test the system	Pay TD, Create TD
Software companies	Code systems, provide descriptions of possible solutions, feedback	Create TD, Pay TD, identify TD
EU Parliament	Creates or changes legislation	Pay TD, Create TD *Depends on whether the system can be expanded and the length of the deadline
Danish Parliament	Changes prioritization and funding, creates or changes legislation	Pay TD, Create TD *Depends on whether the system can be expanded and the length of the deadline
Authority Z	Provides funding, sets requirements, prioritizes main tasks, requests a data extraction	Prevent TD, Create TD

Authority as collaborator	Provides feedback, suggests changes, requests a data extraction	Identify TD, pay TD, Create TD
Authority as user	Provides feedback, sets requirements	Identify TD

5.2.5 Stakeholder consequences on TD on the REPORT system: An example

The following scenario involves several stakeholders and the TD consequences of their actions. The authorities (as collaborators and users) require a specific dataset. This task has a very short deadline, meaning that other tasks are deprioritized and there is insufficient time to test. Thus, this requirement *creates TD*.

The REPORT system does not allow previous editions of the reported information. This functionality minimizes fraud but is inconvenient when the information is incorrect. The law graduate (a stakeholder) *identifies TD* caused by some *citizens* who *create TD* by running test reports. In these cases, the law graduate would like to be able to delete these reports so that they do not interfere with the data, thus *paying TD*. The data is expected to be reliable and accurate because Agency X uses the data to generate statistics and inform collaborating authorities.

Thanks to continuous development, REPORT has grown, but this growth has decreased usability for citizens. Furthermore, the EU (a stakeholder) has passed legislation that requires a new group of citizens to use the system. A new type of control must be implemented in the IT system to accommodate the citizens securely. Therefore, the law graduate argues that it is time to rethink the system and consider refactoring it. They suggest *paying TD* by refactoring the system and specifying the needs of the system to *prevent TD.*

5.2.6 Overview of the stakeholders and their influence on the REPORT system

In Table 8, we present the internal and external stakeholders of REPORT, their actions, and the TD consequences of those actions. The primary consequence of their actions is to *pay TD;* the actions of eight of the eleven stakeholders have this consequence on TD. The least frequent consequence is the *monitoring of TD*, which is caused by the actions of three stakeholders: application managers, IT architects, and IT operations staff.

Table 8. REPORT Stakeholders' Influence on TD

Stakeholder	**Prevent TD**	**Create TD**	**Identify TD**	**Monitor TD**	**Pay TD**
Application managers	X		X	X	X
IT architects	X		X	X	X
IT operations			X	X	
Law graduates	X				X
Citizens		X			X
Software companies		X	X		X
EU Parliament		X			X
Danish Parliament		X			X
Authority Z	X	X			
Collaborating authorities		X	X		X

Authorities as users	X

5.2.7 Remarks on the REPORT system

REPORT is a political system developed on behalf of Authority Z. Therefore, its stakeholders are mostly external. The political interest and external funding result in the REPORT system being in a state of continuous development. Thus, the risk of introducing TD is high; however, we found that the external funding and continuous development also enabled the TD to be paid more promptly. This case shows the power of the external stakeholders. They (1) create TD, (2) allocate funding for this specific system, and (3) influence the priority of the IT system because it is used by external users.

5.3 Comparison of the stakeholders' consequences on TEXT and REPORT

The two IT systems have different purposes and stakeholders. TEXT has predominantly internal stakeholders, whereas REPORT's stakeholders are predominantly external. The difference in stakeholders influenced the systems' further development. TEXT accumulated TD, as it was primarily visible to internal stakeholders, whereas REPORT appeared to pay back a large portion of its TD continually. REPORT has more external stakeholders than TEXT, and because Agency X strives for professionalism, TD affecting external stakeholders has a high priority. The external funding from Authority Z means that the board of directors cannot remove funding and create TD, and so the authority prevents TD. In contrast, TEXT lost resources because the shared funding was spent. This in turn led to the postponement of planned debt, which was meant to relieve the expert users.

We identify seven TD types for the IT systems in both TEXT and REPORT. TEXT is affected by a lack of resources and development, meaning that even debt affecting the usability is not prioritized by the agency. In contrast, the development of the REPORT system is sometimes fast-paced and oriented towards the users. Here the difference between the priority of front-end and back-end systems is clear. There is greater political focus and more development on the REPORT system, whereas the TEXT system received little attention until the Digitization Agency required public organizations to map the IT systems and the systems' vulnerabilities.

The stakeholder landscape, and thus their stance, also differed across the two systems. REPORT has one non-supportive stakeholder but four supportive stakeholders. In comparison, TEXT has two non-supportive stakeholders and three supportive stakeholders. Thus, REPORT's stakeholders appear more supportive and are more prone to cause TD reduction, whereas TEXT's stakeholders appear less supportive and more prone to cause TD creation.

The REPORT system is influenced continually by legislation passed by the EU and Danish Parliaments. Agency X prioritizes TD legislation because of its organizational values. The agency ensures that REPORT supports the legislation so that citizens can report the required information. Additionally, the REPORT system must adhere to the legislation (e.g., the GDPR). More resources for IT development were directed toward REPORT than TEXT; the IT deployment increased the creation of TD. This, along with the new requirements, caused the law graduate to consider refactoring the REPORT system and paying TD. In contrast, the lack of development activity in the TEXT system prevents the system from being sufficiently improved to be able to address Agency X's CMS needs. The TEXT system works even though the usability was low; eventually, however, the system is scheduled to be replaced as it is no longer of a sufficient quality.

To summarize, REPORT is a front-end system, TEXT is a back-end system, and they have different stakeholders. This difference caused different kinds of TD consequences. One solution for addressing TD is to perform stakeholder management. However, stakeholder management must be tailored to specific IT systems because such systems have different stakeholders.

5.4 The stakeholders in Agency X who affect TD

The IT systems we analyzed were distinct and involved a range of stakeholders. However, there was some overlap among stakeholders, and the stakeholders' actions did not appear to change between the systems. We have synthesized our findings from the case study by mapping the high-level categorization of stakeholders and their consequences on TD (Table 9).

Table 9. An Overview of Stakeholder Influence on TD in Agency X

Stakeholder			**Consequences**		
groups	Prevent TD	Create TD	Identify TD	Monitor TD	Pay TD
Other authorities	X	X	X		X
External political entities		X			X
Internal users		X	X	X	X
Legal administration	X				X
Citizens		X			X
Software companies		X	X		X
Board of directors		X			X
IT department	X		X	X	X

Note that all the stakeholders may contribute to paying TD. Surprisingly, we found that most of the stakeholders contributed to creating TD as well. The actions of certain stakeholders cause specific consequences (e.g., the prevention and monitoring of TD). The other authorities, legal administration, and IT department prevent TD and the internal users and IT department monitor TD.

Some TD is inevitable; it occurs in both frequently and seldom developed IT systems. Agency X pays the TD on an IT system when it is held accountable by different authorities. Furthermore, software developers are not the only stakeholder groups to create and pay TD. The actions of non-technical external stakeholders may have consequences on TD. Thus, TD can occur in different types of IT systems and be generated by a range of stakeholders.

6. Discussion

In this section, we discuss our findings and relate them to previous TD studies (Table 8). Our mappings provide an insight into the TD types found in the IT systems and the various stakeholders' stances towards TD. Our research shows how different stakeholders influenced the TD in an agency's IT systems. It provides empirical insights on how stakeholders influence TD in the public sector. Finally, we expand on Rios et al.'s (2018) TDM overview and recommend practical TDM strategies.

6. 1 Research implication

Previous TD research has focused on organizations in the private sector, leaving behind a knowledge gap on TD in the public sector (Klinger et al., 2011; Yli-Humo et al., 2016). Public organizations adhere to different values than private companies (Rose et al., 2018). This is important to stakeholder analysis because ST is based on creating value for the stakeholders. Our study is unique in combining ST with TD to explore how stakeholders in the public setting influence TD (Table 10).

Table 10. Major Findings

Topic	Previous TD studies	Our study
1. Setting of the study	Private sector	Public sector
2. TD stakeholder classification	Distinguish between technical and non-technical stakeholders	Identifies eight types of stakeholders who influence TD
3. TD types	There are 15 types of TD, which previous studies have identified based on code, surveys, and interviews with technical employees.	TD type mapping provides insight into the IT systems' challenges from the perspective of the non-technical and technical employees
4. Stakeholder Stance	N/A	Stakeholders' stance can influence on an IT system's TD
5. Stakeholder action	Identify four macro TDM activities	Expands TDM activities to include daily operation of the IT systems
	Stakeholder responsible for TD actions	Maps specific actions to specific stakeholders
6. Setting of TD action	Find that TD actions occur within an organization	Finds that TD actions may occur both within and outside of an organization
7. Stakeholders' influence on TD	Find that non-technical/business stakeholders influence TDM activities	Finds that stakeholders can influence TDM activities in different ways: preventing,

identifying, monitoring, and,
paying TD

Previous studies have distinguished between technical and non-technical stakeholders (Klinger et al., 2011; Yli-Humo et al., 2016). We expanded the classification of stakeholders by conducting a case study in a government agency (Axelsson, et al., 2013). Most of the agency's stakeholders were non-technical, e.g., other authorities, external political entities, and the legal administration. Furthermore, our categorization enabled us to illustrate that the two IT systems have different stakeholders who have distinctive stances and helped us to stage a nuanced exploration of the stakeholders' activities and their TD consequences. This implies that TDM strategies and activities should be tailored to the specific context.

Rios et al. (2018) map 15 types of TD and provide definitions and examples of these TD types. Previous TD research has identified TD types based on coding, surveys, and interviews with IT professionals (Rios et al., 2020; Ramac et al.; 2022, Freire et al.; 2020; Pérez et al., 2021). Furthermore, they map TD types with TD causes and TD effects (Rios et al., 2020; Ramac et al., 2022) and preventive and payment actions (Freire et al., 2020; Pérez et al., 2021). We find that mapping TD types provides an insight into an IT system's challenges and the corresponding TDM strategies that should be implemented. However, mapping TD types does not indicate exactly how extensive a threat each individual TD type poses. We gained an understanding of how each TD is perceived and which challenges TD causes on a daily basis, by interviewing and observing the technical and non-technical employees interacting with the system.

Rios et al. (2018) identified TDM actions and TDM macro actions and mapped them with existing tools and strategies for TD. We have regarded TDM actions as the consequences of actions by stakeholders. Furthermore, we expand on Rios et al.'s (2018) work by mapping stakeholders to TDM macro actions. Yli-Humo et al. (2016) propose a TDM framework for mapping TDM activities, the tools that support these activities, and those who are responsible for performing the activities. However, they do not map which stakeholders perform the activities. We found that the actions performed in two different IT systems were similar, and the actions led to the same TD consequences. Some of the stakeholders' actions overlapped; additionally, we found that different types of TD actions were performed by different stakeholders. Thus, the actions of a stakeholder are not unique, and this fact increases the robustness of TDM activities.

Klinger et al. (2011) found that non-technical stakeholders can contribute to creating TD, while Yli-Humo et al. (2016) discovered that business stakeholders may influence TDM activities. Our study confirms and elaborates on these findings. We also found that non-technical stakeholders potentially contribute to the prevention, monitoring, and payment of TD. The degree of different stakeholders' influence varied, and the same stakeholder could have several consequences on TD (e.g., both the creation and payment of TD). The consequence of stakeholders' actions on TD consequence influences how to manage and prioritize TD. Not just technical but also non-technical stakeholders influence TD. Therefore, the influence of non-technical stakeholders must be considered when managing TD.

Previous TD research has mostly focused on internal stakeholders, such as software developers, but we have also identified stakeholders external to the organization. We found that several groups of stakeholders (beyond software developers and other internal stakeholders) act in ways that have important consequences on TD. The authorities are required to collaborate to solve their tasks. Some authorities—for example, the Digitization Agency—set quality requirements that impact other authorities. Furthermore, the authorities must adhere to and support new legislation passed by the Parliament. The actions of these external stakeholders have consequences on TD. Rose et al. (2018) identify professionalism, efficiency, service, and engagement as important values in the public sector. We found that these values were central to Agency X and reflected the influence granted to its external stakeholders. As a result, the agency served the authorities quickly and risked creating TD, but also prioritized the IT system and TDM management. This is important because identifying how TD is created makes it possible to prevent or monitor the TD; thus, it constitutes a critical step in ensuring proper TDM.

We combine ST and TD to explore the relationship between stakeholders and TD activities. The two IT systems are distinctive, they have different TD challenges, and their stakeholders have different stances. Combining ST and TD enables us to explore how TD is created by the consequences of stakeholders' actions. The public organizational values have a significant impact on the prioritization of the IT systems, enabling front-end systems to gain a higher priority.

6.2 Practical implications

TD is essential for Governments to address as it can be a structural barrier to digital government (Wilson & Mergel, 2022). Our study has several implications for practitioners. We find that various stakeholders influence TD; it is, therefore, critical that the organization understands and mitigates the stakeholders' impact on TD. First, we found that the stakeholder landscape was larger than reported in previous studies. Second, we found that most of Agency X's stakeholders were non-technical. Thus, TDM cannot be handled by the IT department alone; some non-technical stakeholders must participate as well. As a first step, the various stakeholders must agree on a definition of TD. Such an agreement can make it easier to discuss TD, the difficulties it imposes, TDM activities, and create policy documents regarding TD and TDM.

Third, we found that stakeholder mapping could serve as a tool to identify stakeholders' influence and their respective stances on the system's TD. Numerous stakeholders in the public sector are external, and hence challenging to manage. However, by identifying the risk of a given stakeholder creating TD, an agency can take some precautions or simply increase the frequency of paying TD. Furthermore, identification of the TD types could improve the strategy for improving TD payment and prevention. It is vital that organizations realize and mitigate the risk of TD creation. We, therefore, recommend that the organizations create a stakeholder map, and map TD types to take appropriate actions to manage their TD. Our findings also show that the IT systems' stakeholders differ. Therefore, it is necessary to create stakeholder maps for each of an agency's IT systems. This information can then be incorporated into strategies that address TD, and for example, be integrated into policy documents.

7. Concluding remarks

This study explored how stakeholders influenced the TD in a public agency. The aim was to gather empirical insights on how stakeholders influence TDM. We conducted a longitudinal case study in an agency. The observations and interviews were performed over five months. We combined ST and TD to serve as our theoretical lens to identify the stakeholders, their actions, and the influence of their actions on TD.

Our study makes empirical, practical, and theoretical contributions on stakeholders' influence on TD in the public sector. First, by mapping stakeholders, it was revealed that the relevant stakeholders differ across IT systems. Most of the stakeholders we identified were non-technical and were both internal and external to the organization. Second, the TD types and TDM categorization gave insights into the TD challenges for each IT system, and how these challenges can be addressed by practitioners. Third, ST combined with TD can be used to explore how different stakeholders influence TD. For example, users of IT systems are inclined to identify TD while task and funding providers tend to both resolve and create TD.

Finally, we offer a note on the methodological and conceptual limitations of this study and present potential avenues for future research. Nielsen et al., (2020) identify a lack of research concerning actual TD behavior, therefore we employ a qualitative approach (including interviews and observations) to capture TD behavior. To further narrow our focus on practices surrounding TD, we chose an embedded case study approach and excluded source code examination, as this was already captured by previous studies (Nielsen et al., 2019). This methodological approach enabled us to capture the underlying factors leading to practitioners' TD behavior. A source code examination might have contributed to identifying the TD types, and allowing for a comparison between code quality and how the IT systems are perceived by the stakeholders. Another methodological limitation is caused by the fact that TD can take several years to surface. Thus, a five-month study may not capture all the actions which cause TD creation, or indeed all the consequences of TD.

Further, TD is an evolving concept with its own limitations. First, there is no uniform definition of TD in the extant literature, which may create confusion on what constitutes as TD (Fairbanks, 2020). To address this conceptual limitation, we have employed a broad understanding of TD combined with specific TD types and activities in our analysis. Moreover, the existing literature has primarily revolved around source code and the technical elements of TD (Nielsen et al., 2019). Therefore, existing TD research has focused on technical stakeholders, omitting the political and administrative level and stakeholders. We mitigate this limitation and contribute to TD research by analyzing the organizational processes surrounding TD and identifying non-technical stakeholders, such as the Danish and EU parliaments, legal administration, and caseworkers.

Future studies can extend our study and combine these techniques (observation, in-situ interviews, and document analysis) with code examinations. A case study can be generalized on an analytical and theoretical level, but it cannot be generalized on a statistical level (Yin, 2018). The agency we studied might not be representative and therefore the conditions of our study may impact the use of our findings and recommendations in other settings We encourage scholars to repeat our study and expand on it. Follow-up studies could be conducted on different IT systems or in different agencies. Studies assessing systems with different IT development patterns would allow for comparison and discussion of the stakeholders' influence on TD. Such studies can contribute to the development of general management strategies for public organizations. Their results would help to reduce TD's potential negative impact, improve efficiency and stability, and reduce costs in the public sector. Finally, the long-term causes and consequences of TD would be an interesting topic for future studies.

Acknowledgements – Blinded for peer-review

References

Al-Ibraheem, N., & Ruël, H. (2011). In-House vs. Off-the-shelf e-HRM applications. In *Enterprise Information Systems: Concepts, Methodologies, Tools and Applications* (pp. 752–775). IGI Global.

Alves, N. S., 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.

Avgeriou, P., Ernst, N. A., Nord, R. L., & Kruchten, P. (2016). Technical Debt: Broadening Perspectives Report on the Seventh Workshop on Managing Technical Debt (MTD 2015). *ACM SIGSOFT Software Engineering Notes*, 41(2), 38–41.

Axelsson, K., Melin, U., & Lindgren, I. (2013). Public e-services for agency efficiency and citizen benefit—Findings from a stakeholder centered analysis. Government information quarterly, 30(1), 10-22.

Blair, J. D., & Whitehead, C. J. (1988). Too Many On The Seesaw: Stakeholder Diagnosis And Managemen. *Journal of Healthcare Management*, 33(2), 153.

Blomberg, J., Burrell, M.: An Ethnographic Approach to Design. In : Jacko, J.A. (ed.) *Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Boca Raton, FL (2012).

Brannan, M. J., & Oultram, T. (2012). Participant observation 17. *Qualitative organizational research: Core methods and current challenges*, 296.

Berenguer, C., Borges, A., Freire, S., Rios, N., Ramač, R., Taušan, N., ... & Spínola, R. (2021). *Technical Debt is not Only about Code and We Need to be Aware about It*.

Cunningham, W. (1992). The WyCash portfolio management system. ACM SIGPLAN OOPS Messenger, 4(2), 29–30.

Danish Ministry of Finance: Regeringens kasseeftersyn på it-området. Copenhagen, Denmark (2017).

Eom, S. J., & Lee, J. (2022). Digital government transformation in turbulent times: Responses, challenges, and future direction. Government Information Quarterly, 101690.

Fairbanks, G. (2020). Ur-technical debt. IEEE Software, 37(4), 95-98.

Flak, L. S., & Rose, J. (2005). Stakeholder governance: Adapting stakeholder theory to e-government. *Communications of the Association for Information Systems*, 16(1), 31.

Freeman, R. E. (1984). Strategic management: a stakeholder approach. Boston: Pitman

Freire, S., Rios, N., Mendonça, M., Falessi, D., Seaman, C., Izurieta, C., & Spínola, R. O. (2020, March). Actions and impediments for technical debt prevention: results from a global family of industrial surveys. In Proceedings of the 35th Annual ACM Symposium on Applied Computing (pp. 1548–1555).

Glaser B. & Strauss A.L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, New York.

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, 611–642.

Griffith, I., Taffahi, H., Izurieta, C., & Claudio, D. (2014, December). A simulation study of practical methods for technical debt management in agile software development. In Proceedings of the Winter Simulation Conference 2014 (pp. 1014–1025). IEEE.

Guenduez, A. A., Mettler, T., & Schedler, K. (2020). Technological frames in public administration: What do public managers think of big data? Government Information Quarterly, 37(1), 101406.

Janssen, M., & Helbig, N. (2018). Innovating and changing the policy-cycle: Policy-makers be prepared!. Government Information Quarterly, 35(4), S99-S105.

Janssen, M., & Janowski, T. (2015). Tribute to John Bertot and message from the incoming Editors-in-Chief. *Government Information Quarterly*, 32(2), 103–104.

Klinger, T., Tarr, P., Wagstrom, P., & Williams, C. (2011, May). An enterprise perspective on technical debt. In Proceedings of the 2nd Workshop on managing technical debt (pp. 35–38).

Kvale, S. (1997). Interview. 7. oplag. Hans Reitzels Forlag.

Lindgren, I., Madsen, C. Ø., Hofmann, S., & Melin, U. (2019). Close encounters of the digital kind: A research agenda for the digitalization of public services. *Government Information Quarterly*, *36*(3), 427-436.

Lindgren, I., Melin, U., & Sæbø, Ø. (2021). What is E-Government? Introducing a Work System Framework for Understanding E-Government. *Communications of the Association for Information Systems*, 48(1), 43.

Mikalef, P., Lemmer, K., Schaefer, C., Ylinen, M., Fjørtoft, S. O., Torvatn, H. Y., ... & Niehaves, B. (2021). Enabling AI capabilities in government agencies: A study of determinants for European municipalities. Government Information Quarterly, 101596.

Nielsen, M. E., Madsen, C. Ø., & Lungu, M. F. (2020, August). Technical Debt Management: A systematic literature review and research agenda for digital government. In International Conference on Electronic Government (pp. 121–137). Springer, Cham.

Nielsen, M., & Skaarup, S. (2021, October). IT Portfolio management as a framework for managing Technical Debt: Theoretical framework applied on a case study. In International Conference on Theory and Practice of Electronic Governance (pp. 89-96).

Omar, A., Weerakkody, V., & Daowd, A. (2020). Studying Transformational Government: A review of the existing methodological approaches and future outlook. Government Information Quarterly, 37(2), 101458.

Panagiotopoulos, P., Klievink, B., & Cordella, A. (2019). Public value creation in digital government. Government Information Quarterly, 36(4), 101421.

Pérez, B., Castellanos, C., Correal, D., Rios, N., Freire, S., Spínola, R., ... & Izurieta, C. (2021). Technical debt payment and prevention through the lenses of software architects. *Information and Software Technology*, 140, 106692.

Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., ... & Spinola, R. (2022). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. *Journal of Systems and Software*, 184, 111114.

Reinwald, A., & Kraemmergaard, P. (2012). Managing stakeholders in transformational government—A case study in a Danish local government. Government Information Quarterly, 29(2), 133-141.

Ribeiro, L. F., de Freitas Farias, M. A., Mendonça, M. G., & Spínola, R. O. (2016, April). Decision Criteria for the Payment of Technical Debt in Software Projects: A Systematic Mapping Study. In ICEIS (1) (pp. 572–579).

Rios, N., de Mendonça Neto, M. G., & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102, 117–145.

Rios, N., Spínola, R. O., Mendonça, M., & Seaman, C. (2020). The practitioners' point of view on the concept of technical debt and its causes and consequences: a design for a global family of industrial surveys and its first results from Brazil. *Empirical Software Engineering*, 25(5), 3216–3287.

Rose, J., Flak, L. S., & Sæbø, Ø. (2018). Stakeholder theory for the E-government context: Framing a value-oriented normative core. *Government Information Quarterly*, 35(3), 362–374.

Scholl, H. J. (2001). Applying stakeholder theory to e-government. In *Towards the E-society* (pp. 735–747). Springer, Boston, MA.

Scholl, H. J. J., Fidel, R., Liua, S. M., Paulsmeyer, M., & Unsworth, K. (2007, September). E-government field force automation: Promises, challenges, and stakeholders. In International Conference on Electronic Government (pp. 127–142). Springer, Berlin, Heidelberg.

The Swedish National Audit:Föråldrade it-system–Hinder för en effektiv digitalisering. Stockholm (2019).

Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt. Journal of Systems and Software, 86(6), 1498-1516.

VERBI Software. (2019). MAXQDA 2020 [computer software]. Berlin, Germany: VERBI Software. Available from maxqda.com.

Wilson, C., & Mergel, I. (2022). Overcoming barriers to digital government: mapping the strategies of digital champions. Government Information Quarterly, 39(2), 101681.

Yin, R. K. (2018). Case study research and applications.

Yli-Huumo, J., Maglyas, A., & Smolander, K. (2016). How do software development teams manage technical debt?–An empirical study. *Journal of Systems and Software*, 120, 195–218.

Ølnes, S., Ubacht, J., & Janssen, M. (2017). Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. Government Information Quarterly, 34(3), 355-364.

Appendix

We identified eight main stakeholder groups, which we describe in this section.

Board of directors

The board of directors decides which new IT projects the IT department should initiate. It determines the priority of the projects and systems, thus influencing the resources assigned to them. For example, when a project is delayed, the board of directors decides whether resources from other IT systems or projects should be moved to the delayed project.

IT department

We describe four categories within the IT department. (1) The IT operations team maintains the servers the IT systems are deployed to and are responsible for IT security. They also ensure that the deployment process runs smoothly by monitoring the IT systems and troubleshooting malfunctions. For example, during traffic peaks, they ensure that the IT systems do not become unavailable to citizens. (2) Application managers manage IT systems, including setting priorities for maintenance and minor IT development tasks. They ensure that the business's needs are met and that incidents are handled. An application manager links the developer, the relevant business unit, and the IT architect, as well as working with the business unit to plan and prioritize tasks. (3) IT architects plan the overall architecture of IT systems and ensure that future solutions fit into the architecture. When an IT developer proposes more than one solution, the IT architect determines which is the most suitable of the available options. (4) IT project teams develop new IT solutions for existing IT systems or develop new IT systems. When they integrate solutions into existing IT systems, they often need to adjust the IT system to improve functionality.

Legal administration

Agency X has a legal team in addition to having law graduates working in the different business units. They ensure that the IT systems support legislation and that systems remain in adherence to the law when new functionality is added.

Internal users

Agency X has caseworkers and experts using the systems that we classify as internal users. The expert users gather information from the other internal users and provide feedback to the IT team that manages the system. The expert users become involved when the IT system requires further development. They clarify requirements and test new features.

Citizens

The citizens use the agency's services and thus the IT systems that the agency provides. Furthermore, they provide feedback to the agency (e.g., by contacting support). Their feedback is included when setting priorities for IT development. We spoke with a caseworker who emphasized that one of the agency's responsibilities was to be responsive to the problems and suggestions flagged by citizens.

Software companies

Agency X hires software companies to sit on-site to collaborate on developing and maintaining the IT systems. The consultancy company gives feedback on the state of the IT systems, suggests solutions, and codes the individual tasks.

Other authorities

The authorities that serve as stakeholders (including the Digitization Agency, which sets quality requirements for IT systems in public organizations) fall into one or a combination of four categories. First, there are task setters and funding providers such as Authority Z. Agency X has, on several occasions, developed an IT system on behalf of other authorities and ministries. For example, Agency X built and maintains REPORT for Authority Z, which continues to present its needs to Agency X and provide funding. Second, authorities can be users because their employees may use the system, which is seen in REPORT. Third, authorities can, and often must be, collaborators because they exchange data. For example, someone might need to be reported to the police or tax office. Fourth, authorities can set quality requirements for IT systems. For instance, the Digitization Agency required the authorities to map their critical IT systems and create action plans for the problems they encountered.

External political entities

The EU Parliament creates and changes legislation. This influences the IT systems built to fulfill its requirements (e.g., the GDPR). Alternatively, IT systems support the legislation—for example, by implementing a new type of control. The Danish Parliament performs the same type of actions as well as determining the level of funding to be allocated to the authorities through the Finance Act. Changes in the Danish Parliament can affect Agency X's tasks and funding.