ISSN: 2755-0214

Journal of Economics & Management Research

SCIENTIFIC Research and Community

Review Article

Open Access

Strategic Technical Debt Management: An In-depth Analysis of its Consequences and Mitigation Strategies in the Financial Industry

Sridhar Mooghala

Senior Advisor, Fiserv, United States

ABSTRACT

This research looks into the important area of "Strategic Technical Debt Management" in software programs and shows how it has big effects on the complicated financial industry. This paper also takes a close look at how technical debt affects the quality, maintainability, and speed of software, focusing on how it affects economic systems. Adding graphs, tables, charts, and pictures to a story makes it better by making complicated ideas easier to understand. Real-life case studies from the banking sector give us concrete information and show us more effective methods. This paper is meant to make a big contribution to the conversation about strategic technical debt. It is aimed at both practitioners and academics and will be published in reputable journals.

***Corresponding authors**

Sridhar Mooghala, Senior Advisor, Fiserv, United States.

Received: December 01, 2022; **Accepted:** December 07, 2022; **Published:** December 15, 2022

Keywords: Technical Debt, Financial Industry

Introduction

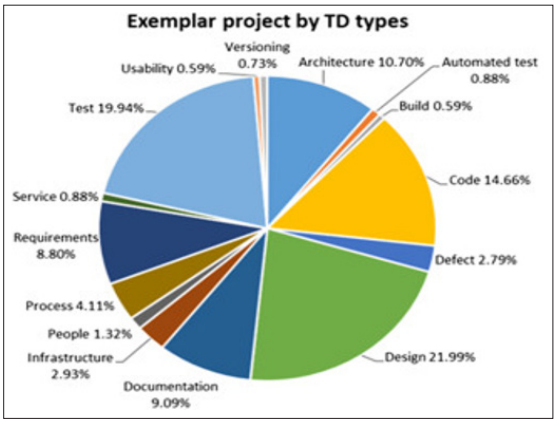
Because the financial world is always changing, it is now necessary to know how to handle strategic technical debt in software applications. As technology keeps getting better, businesses have to deal with the big problems that come with technical debt. This kind of debt has a big impact on software quality, system maintenance, and general performance. This study looks into the many complicated parts of strategic technical debt management, mainly how it subtly affects the financial industry. This piece tries to show how complicated the relationship is between financial services and technology by looking at the unique problems that financial institutions face.

This study aims to make a big difference in our understanding of strategic technical debt by using information from carefully picked sources. In this way, it hopes to make it easier to use better methods for making software, especially in the financial industry. Because it's hard to integrate technologies into financial systems, strategic technical debt management is becoming much more important. This shows how important it is to be proactive and well-informed at this important time in the history of both business and technology.

Literature Review

The term "technical debt" (TD) refers to a conceptual construct that is used in the dynamic field of software development to represent the compromises made throughout the development process that may result in a wide range of future challenges [1]. Increasing maintenance costs, decreased system responsiveness, and weakened reliability are some of these difficulties. They lead to a complex interaction between intentional compromises made to fulfill deadlines or overcome resource constraints and unintentional gains brought about by a lack of knowledge or comprehension.

There are several types of technical debt. Study identifies these as follows: design debt, which is characterized as poorly designed code that is difficult to read and alter; code debt, which is dirty code full of errors and inconsistencies; testing debt, which is caused by inadequate testing and leaves the software susceptible to flaws; documentation debt, which is the result of incomplete documentation, which impedes understanding and upkeep; and technical infrastructure debt, which relates to technological infrastructure [2].

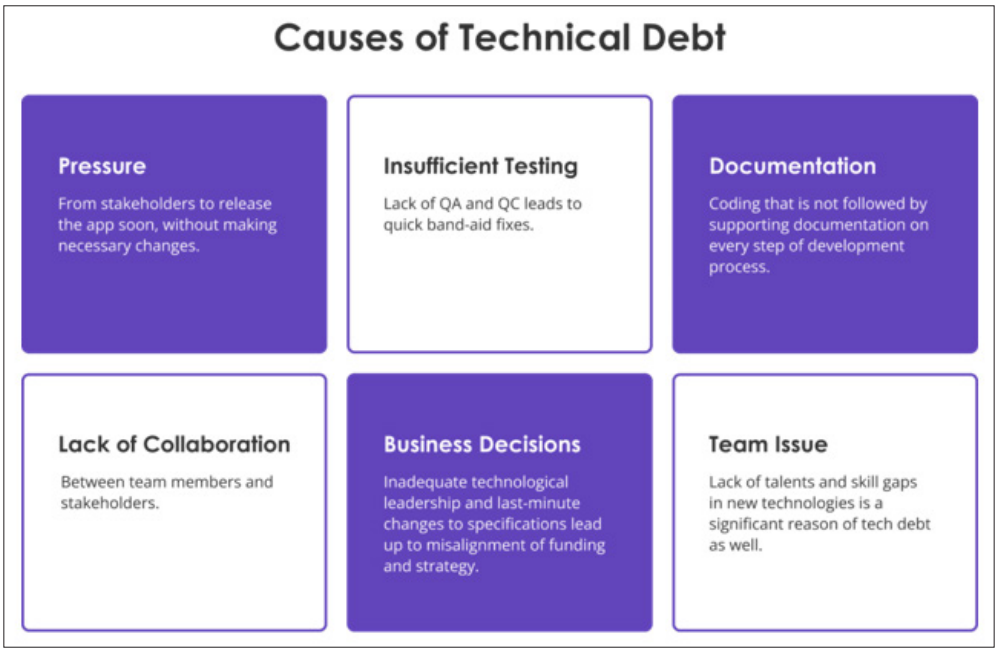


Customer satisfaction, company operations, and the financial health of businesses are all negatively impacted by technical debt in software programs [3]. Increased maintenance costs, lower software quality, less agility, a higher chance of failure, and lower customer satisfaction are among the anticipated outcomes. All of these results give organizations a complex web of challenges that they need to navigate successfully. However, in this intricate

setting, technical debt strategic management emerges as a useful tool for the financial sector. The procedure comprises identifying, prioritizing, and systematically resolving technical debt in order to maximize the value of the company. This offers a sophisticated way to weigh short-term sacrifices against long-term gains.

Several academic studies have looked into the financial industry's strategic management of technological debt, explaining potential benefits and outlining best practices. A thorough 2020 analysis conducted by McKinsey & Company found that TD is in charge of almost 40% of the IT asset value in the financial sector [4]. Furthermore, the study emphasized the idea that companies with the capacity to manage TD effectively could achieve increased profitability as well as substantial cost savings. It promoted the use of cloud technologies as a possible fix to enhance the management of TD. Apart from exploring its definition, causes, effects, and possible advantages, the article by provided a thorough introduction to TD by looking at useful strategies like code reviews and refactoring to guarantee efficient TD management [5].

Even though there is more and more academic writing on strategic TD management, there are still gaps in our knowledge, especially when it comes to how it is used in the financial sector. More research needs to be done on two very important areas: creating reliable metrics to measure the effects of TD and giving financial institutions better tools for making decisions about TD management actions. Additionally, organizations need clearer directions to help them find a balance between the short-term benefits of TD and the long-term risks that come with these concessions. Also, more focused study is needed right away to come up with the best ways to manage TD in different financial sectors, such as asset management, insurance, and banking. By recognizing and fixing these problems, we can improve our understanding of the strategic management of technical debt and give practical advice to organizations working in the complex finance field.



Research Design, Methods of Data Collection and Criteria for Source Selection

- **Research Design**

The research design for this study will be mixed methods, incorporating qualitative and quantitative data collection techniques to obtain a comprehensive understanding of strategic technical debt management within the financial industry. Exploration of current knowledge and practices will be facilitated through qualitative methodologies, which include a comprehensive literature review, interviews with industry experts, and case studies of financial institutions. The study will employ quantitative methods to examine financial data and survey financial institutions to quantify the operational repercussions of technical debt.

- **Methods of data collection**

The literature review phase will thoroughly analyze scholarly journals, industry reports, and other pertinent literature. This will establish a basis for identifying demonstrated knowledge and optimal approaches to strategic technical debt management within the financial industry. In addition, a restricted quantity of case studies will be undertaken to examine the management of technical debt by financial institutions, thereby illuminating particular obstacles and resolutions encountered by various organizations.

- **Criteria for Selecting Sources**

The selection of sources for the literature review will be conducted with great care, considering their pertinence to the research topic, credibility established through publication in reputable academic journals or industry organizations, and currentness to reflect the most recent developments and availability via academic databases or other dependable resources. The selection process for expert interviews and case study sources will adhere to specific criteria, including but not limited to proficiency in strategic technical debt management and the financial sector, accessibility for participants, and the inclusion of a wide array of organizations and individuals

representing diverse viewpoints.

- **Justification for Chosen Methodology**

A number of crucial factors support the rationale for choosing a mixed-methods research design. Primarily, integrating qualitative and quantitative methodologies provides supplementary viewpoints regarding the subject of investigation, guaranteeing a more all-encompassing and intricate comprehension. Qualitative research methods, including literature reviews and case studies, are highly suitable for investigating the present state of knowledge and implementations regarding technical debt management. In addition, using quantitative methods is crucial to quantify the ramifications of technical debt on financial institutions, thereby introducing a data-centric aspect to the investigation.

The justification for the methodology is additionally supported by the concept of triangulation, which posits that the amalgamation of numerous data sources and methods bolsters the credibility and dependability of the results. Using qualitative and quantitative methodologies, this research endeavors to authenticate and corroborate observations derived from various sources, guaranteeing a comprehensive and rigorous examination. Moreover, by employing a mixed-methods approach, a thorough examination of the research subject can be conducted, surpassing the individual limitations of either method. This results in a more holistic comprehension of strategic technical debt management within the financial sector.

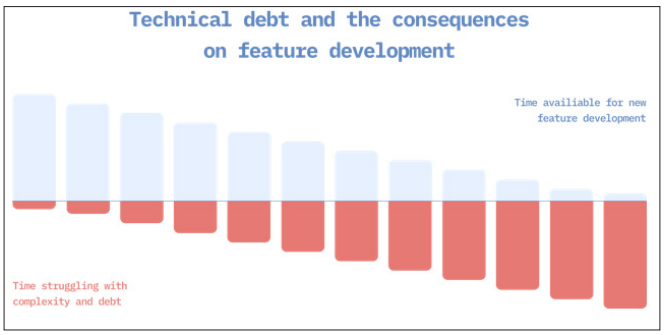
Implications of Strategic Technical Debt

The Effects of Technical Debt on the Quality, Maintainability, and Performance of Software:

- **Software Quality:** The consequences of Strategic Technical Debt (TD) have far-reaching effects on software quality, maintainability, and performance. Within software quality, TD substantially increases the likelihood of encountering flaws and defects, which has a detrimental impact on the user experience and, as a result, reduces the overall satisfaction of customers [6]. Furthermore, the negative consequences also impact functionality, as TD leads to the development of features that are either incomplete or inadequately implemented, thereby limiting the overall efficacy of the software. The concerns are further intensified by emerging security vulnerabilities, which increase the likelihood of catastrophic incidents like data intrusions.
- **Maintenance Considerations:** As attention transitions to maintainability, the influence of TD becomes notably apparent. TD significantly impacts the development process by increasing the complexity of the code, which consequently heightens the difficulties linked to comprehending and modifying it. Enhanced intricacy is a contributing factor to prolonged cycles of development and escalated expenses for maintenance. Furthermore, the impediments imposed by TD impede the reuse of code, resulting in redundant endeavors and increased costs associated with overall development. The cumulative consequence is a noticeable decrease in developer productivity, as the complexities introduced by TD pose a significant obstacle, resulting in project setbacks and failure to meet deadlines.
- **Performance:** In terms of performance, TD generates a software environment characterized by significantly reduced responsiveness and scalability, which harms the overall user experience. The adverse effects also encompass the utilization of resources, such as increased pressures on the central processing unit (CPU) and memory, which in turn escalate expenses and reduce operational effectiveness. In addition, the software failures that are practically certain to occur due to the compromised dependability caused by TD disrupt normal business operations. The significance of vigilant strategic technical debt management is highlighted by the interdependence of software quality, maintainability, and performance, underscored by this multifaceted impact.

Consequences Specific to the Financial Industry

The financial sector is significantly impacted by TD, which exacerbates the difficulties that institutions are already confronted with. TD complicates regulatory compliance for financial institutions, thereby increasing the probability of incurring fines and penalties for noncompliance with industry regulations, in addition to the general repercussions enumerated [7]. As TD increases the likelihood of system disruptions and data breaches, operational risk escalates, thereby threatening the reputation and profitability of the financial institution. Moreover, TD undermines consumer confidence and trust in the financial industry, ultimately leading to a negative customer experience. Consequently, this compromises the brand reputation and enduring relationships of the institution. The ramifications emphasize the critical necessity for proactive administration of strategic technical debt, particularly in sectors where accuracy, dependability, and adherence are paramount.



Mitigation Strategies

It is necessary to apply complex strategies at every stage of the development process to manage strategic technical debt effectively

- **Identification and Prioritization:** The initial phase involves conducting a thorough Technical Debt Inventory, which entails conducting frequent audits to identify and document any current technical debt methodically. The next phase, Prioritization, is a complex procedure that entails assessing the severity and impact of each debt item. This assessment enables the prioritization of tasks by considering potential risks and their effect on a company, allowing for the efficient deployment of resources for maximum effectiveness.

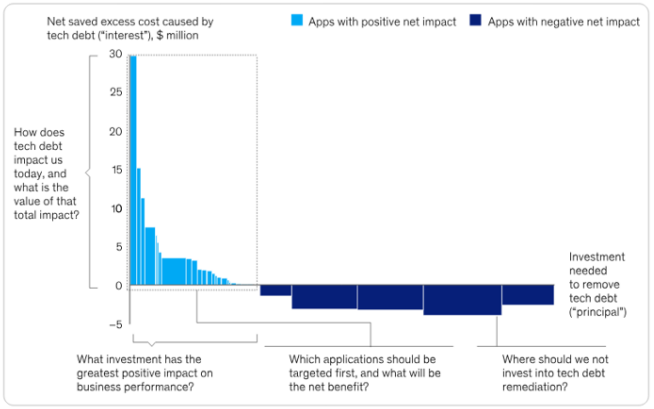
TABLE IV. PRIORITIZATION FACTORS – AS PART OF GUT FEELINGS

Current Factors Influencing the TD Prioritization		
Company	*Prioritization Aspect*	*Examples of Argumentation*
A	Risk Assessment	*"We want to fix issues that potentially could make us reach a crisis point. It is also important to focus on the potential risk of actually doing the refactoring."*
B	Product or Business Needs	*"When doing our prioritization, we assess "what is the most important* [TD item to refactor], *either from our* [developmemnt team] *point of view or a business point of view."*
B	Resource Utilization	*"Things gets prioritized because of available resources. We do not want anyone to sit and wait only."*
C	Software Quality	*"I focus a lot on different compromised quality 'ilities,' such as maintainability and flexibility during the prioritization of technical debt."*
C	Financial	*"Management is always interested in ROI. Thus, if you invest money, you should be able to get it back relatively quickly."*
D	Product or Business Needs	*"We do consider future perspectives. About the urgency, the request coming from the surroundings. So, it's really based on experience, on the discussion, on personal feelings."*

- **Refactoring and Modernization:** Repaying debt is a complex undertaking that requires systematic code reworking to improve code quality, maintainability, and performance. Concurrently, the strategy of Modernization entails transitioning to current technologies and frameworks. This resolves obsolete dependencies, promoting increased flexibility in development methodologies.
- **Continuous Integration and Delivery (CI/CD):** Automation plays a crucial role in Continuous Integration and Delivery (CI/CD). Automated Testing encompasses unit, integration, and performance elements, guaranteeing the prompt identification and resolution of problems. In addition, Continuous Integration enables the regular merging of code to avoid divergence and supports continuous reworking [7]. Simultaneously, Continuous Delivery automates the software release process, guaranteeing the prompt and dependable deployment of new features and updates.
- **Documentation and Knowledge Sharing:** The concept of Documentation and Knowledge Sharing is crucial. Ensuring concise and current documentation is not merely a procedural duty; it fosters comprehension among the development team. In addition, doing regular code reviews acts as a proactive strategy to detect possible technical debt and promote implementing optimal practices. Simultaneously, continuous education activities strive to educate developers about the hazards of technical debt, cultivating a culture that encourages the development of clean and easily maintainable code.
- **Enhancing the Process:** A comprehensive approach is implemented in the process improvement strategy. Agile techniques facilitate the process of iterative development and ongoing enhancement. Embracing a DevOps culture promotes collaboration between development and operations teams, eliminating barriers and improving overall efficiency. In addition, utilizing automated infrastructure management optimizes operations, minimizing the need for manual labor and promoting a proactive approach to managing strategic technological debt.

Maintain a tech balance sheet that tracks tech debt, allows economics-based prioritization, and triggers actions.

Illustration of tech debt abatement curve prioritizing application modernization



Source: McKinsey analysis

McKinsey & Company

Instances of Addressing Technical Debt in the Financial Sector

Bank of America (BoA) encountered a substantial obstacle with its old infrastructure, which was constructed using COBOL and obsolete technology [9]. This restricted its capacity to provide novel digital services and resulted in significant expenses for upkeep. BoA took proactive action by launching a multi-year modernization effort. This entailed transferring crucial programs to a contemporary cloud-based platform and embracing agile development practices. BoA achieved significant outcomes, including a decrease in IT expenses, a notable enhancement in system uptime to 99.9%, and expedited progress in creating novel digital banking offerings.

Regarding JP Morgan Chase, the issue at hand was a complicated and disjointed IT infrastructure consisting of more applications operating on several platforms. Due to this complexity, there were elevated expenses and operational inefficiencies. JP Morgan developed a strategic solution called the "Platform of the Future" program [10]. The goal was to streamline their IT infrastructure and establish uniformity by adopting cloud-based systems. Impressive results were achieved, as JP Morgan's server footprint was reduced. The consolidation optimized IT operations and allocated resources for strategic innovation, demonstrating the beneficial outcome of a well-implemented strategy to address technical debt.

Conclusion

To sum up, this study adds a great deal by exploring the complex field of "Strategic Technical Debt Management" in the financial industry. The investigation starts with a thorough literature study explaining technical debt's many manifestations and effects. Successful mitigation techniques are demonstrated by real-world case studies from Bank of America and JP Morgan Chase, highlighting the applicability of strategic management techniques in real-world settings. The study emphasizes how crucial it is to identify technical debt in financial systems, give it top priority, and deal with it strategically. Doing this promotes the incorporation of efficient mitigation techniques, such as process improvement, refactoring, continuous integration and delivery, identification and prioritizing, and documentation. The recommendations are meant to point practitioners in the direction of best practices as financial institutions work through the challenges of technological integration. This study fosters a proactive strategy to reduce technical debt and improve financial systems' resilience and innovation capacity, making it an invaluable resource for

researchers and industry experts.

These real-world examples from the banking industry demonstrate the efficacy of proactive strategies in eliminating technical debt. Institutions can overcome the challenges of outdated systems by embracing modernization, implementing agile approaches, and strategically integrating their IT infrastructure. This approach will result in better efficiency, reduced costs, and a firm platform for continuous innovation.

References

1. https://www.diva-portal.org/smash/get/diva2:1674968/FULLTEXT01.pdf.
2. https://www.sonarsource.com/learn/technicaldebt/#:~:text=Technical%20debt%2C%20also%20known%20as.
3. TERESE BESKER (2020) Technical Debt: An Empirical Investigation of Its Harmfulness and on Management Strategies in Industry. https://research.chalmers.se/publication/518318/file/518318_Fulltext.pdf.
4. Vishal Dalal, Krish Krishnakanthan, Björn Münstermann, Rob Patenge (2020) Tech debt: Reclaiming tech equity. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-debt-reclaiming-tech-equity.
5. Guo Y, Seaman C, QB da Silva F (2016) Costs and obstacles encountered in technical debt management – A case study. Journal of Systems and Software 120: 156-169.
6. Besker T, Ghanbar H, Martini A, Bosch J (2020) The influence of Technical Debt on software developer morale. Journal of Systems and Software 167: 110586.
7. The Nightmare Letter: A Subject Access Request Under GDPR (2018) https://news.ycombinator.com/item?id=16606629.
8. https://www.suse.com/suse-defines/definition/continuous-integration/.
9. World Economic Forum (2017) Beyond Fintech: A Pragmatic Assessment of Disruptive Potential In Financial Services Part of the Future of Financial Services series | Prepared in collaboration with Deloitte. https://www3.weforum.org/docs/Beyond_Fintech_-_A_Pragmatic_Assessment_of_Disruptive_Potential_in_Financial_Services.pdf
10. https://www.euromoney.com/article/2abgq0jmb56r29kd61yps/awards/awards-for-excellence/the-worlds-best-investment-bank-2022-goldman-sachs-all-for-one-and-one-for-all