



Designing a Technical Debt Visualization Tool to Improve Stakeholder Communication in the Decision-Making Process: A Case Study

Alexia Pacheco^{1,2(✉)}, Gabriela Marín-Raventós^{1,2},
and Gustavo López¹

¹ Research Center for Communication and Information Technologies (CITIC),
University of Costa Rica (UCR), San José, Costa Rica
{alexia.pacheco, gabriela.marin,
gustavo.lopez_h}@ucr.ac.cr

² Computer Science Graduate Program (PPCI), University of Costa Rica (UCR),
San José, Costa Rica

Abstract. Technical debt is the release of immature software to meet time to market. In large complex companies, technical debt drives IT decision, even when participants in the decision-making process do not realize it. In this paper, we propose technical debt visualizations that can serve as a communication platform between different stakeholders. We conducted a case study in a large multi-industry state-owned company that faces significant challenges due to global digital transformation, its rigid control structures, and external pressures for cost reduction and investment optimization. In this study, we designed and evaluated technical debt visualizations to get feedback concerning its usefulness and ease its future acceptance. The results from this case study show that technical debt visualizations were considered useful for decision-making processes associated with software life cycle, especially for executives, business managers, software product owners, architects and project managers.

Keywords: Design requirements · Technical debt visualization
Technical debt · Data visualization · Data-driven decision-making

1 Introduction

Deploying immature software in a company is like acquiring debt [1]. Technical debt means that even though participants in the software development process know that deploying a piece of software might cause problems, they decide to do it. The two main factors that create technical debt are software complexity (unintentional technical debt) and strategic decisions (intentional technical debt) [2]. In large complex organizations with constant changes, these factors are key because the debt acquired by someone's decisions will become apparent when someone else is in that position.

At Dagstuhl Seminar a definition for technical debt was proposed: “*In software-intensive systems, technical debt is a collection of design or implementation constructs that are expedient in the short term, but set up a technical context that can make future changes more costly or impossible. Technical debt presents an actual and contingent*

liability whose impact is limited to internal system qualities, primarily maintainability and evolvability” [3]. For this research, technical debt is the release of immature software artifacts to meet time to market [4].

In this paper, we describe a complex organizational context in which technical debt offers an opportunity to improve communications in the decision-making processes. Since the decision-making process is usually conducted at a high level of the organization, technical debt needs to be described at an architectural level (rather than at a technical level). Therefore, We considered the stakeholders and concerns identified by Li, Liang, and Aygeriou [5]. In their research, authors propose different viewpoints that provide a comprehensive understanding of technical debt at a managerial level.

Furthermore, since this paper presents visualizations as communication tools, we used the data structure collections spreadsheet described by Guo, Seaman, and da Silva [6].

During the first efforts of this research, we faced resistance due to the novelty of the concept and the complexity of the context. This phenomenon is independent of technical debt (i.e., significant efforts are required to gain acceptance of any new practices and artifacts). However, we believe that visualizing and measuring technical debt is provides substantial benefits for the company.

Following the design science approach [7], we conducted a case study in which visualizations (artifacts) were designed to show technical debt explicitly. A group of stakeholders contributed with their perspectives of the information provided through technical debt visualizations. We gathered these perspectives using a survey.

The primary goal of technical debt visualization is to promote a shared understanding of different company-wide perspectives and contribute to achieving general agreements in the decision-making process of the company.

In this paper, we aim to show an example how the design science approach made possible the introduction of novel concepts despite resistance in the organization and to share our findings of technical debt visualizations.

2 Context

The complex and accelerated world in which companies work, force decision-makers to decide between well-built software solutions and quick releases of software that sometimes carry defects [1]. This problem is evident in large companies that are accustomed to rigid processes but are affected by increasingly smaller business windows of opportunity.

In most companies, IT decisions are based on business criteria. However, in large companies, time to market pressure is usually distant in the corporate ladder from the IT decision makers. IT decision makers focus on information technologies that they believe allow business automation and operation of the corporation. Moreover, the decisions in a software project affect different contexts (e.g., hardware developments, marketing, production planning). Therefore, coordination between decision makers in different contexts and with different perspectives is required [2].

This case study was developed in the context of a multi-industry state-owned company with more than ten thousand employees. By state-owned, we mean that it is a company created by the government to develop commercial activities on the government's behalf. As a state-owned company, it must meet many regulations.

By multi-industry, we mean that the company is engaged in two entirely different businesses. Moreover, the company is geographically distributed around the country.

Inside the company, IT is managed in two ways. There is an IT Department for the whole company. However, each management (i.e., each industry) also has local IT areas. The IT Department (1) assures that IT activities and investment comply with regulations and standards, and (2) supports all organizational management levels in their technological requirements.

The company's IT investment is considerable. The company's IT budget is significant compared with total company budget. Most of the IT expenses go to third parties providing services or products (e.g., outsourcing of software development and maintenance, outsourcing of solution operation, solution and infrastructure providers). Rigid control structures are used to manage and monitor incomes and expenses in the organization. Furthermore, since it is a multi-industry company, information has been managed in silos (i.e., each industry segment managed information separately). Since it is a state-owned company, many technical regulations are applied for regulatory purposes (e.g., ITIL, COBIT, TOGAF and ISO 27000).

Efforts to comply with government regulations for IT started several years ago, and IT-related practices have different levels of implementation, most of them based on ITIL and COBIT. These efforts have driven financial and engineering practices looking to create a holistic vision of IT.

A significant disruption occurred when one of the industries in which the company performs moved from a monopolistic industry to a competitive one. This disruption forced several strategic competitive efforts and to investment in IT to optimize budgeting, and costs. Another challenge emerged because of technology disruptions in the other industry. As a consequence, the company faces significant challenges due to digital transformation caused by technology disruptions and converging technologies in both industries.

The company began a process to show the comprehensiveness of IT and to provide it with visibility across the organization. A data model was developed to define a common language and to support data mining across many data sources in the company.

3 Related Work

In 2009, Munzner [8] proposed the Nested Model for Visualization Design and Validation which provides a framework for thinking about the design and validation of visualization systems at four levels. According to [8], the four levels are: (1) characterize the tasks and data in the vocabulary of the problem domain, (2) abstract them into operations and data types, (3) design visual encoding and interaction techniques, and (4) create algorithms to execute these techniques efficiently. These levels are nested; the

output from an upstream level is inputted to the downstream level below. The challenge of this nesting is that an upstream error inevitably cascades to all downstream levels. If a poor choice was made in the abstraction stage, then even perfect visual encoding and algorithm design will not create a visualization system that solves the intended problem.

In [9] authors indicate that this model has provided guidance, motivation, framing and ammunition for a broad range of visualization proposals, including problem-driven design studies, technique-driven work, evaluation, models, and systems.

In 2014, Shahin, Liang and Babar [10] conducted a review to determine the best visualization techniques for software architecture. This paper was useful to determine some types of visualizations that also apply to technical debt.

In 2016, Fernández-Sánchez, Garbajosa, Yague and Perez [11], conducted a systematic mapping to identify and analyze the elements required to manage technical debt. This work was useful to associate key elements associated with technical debt. However, the perspective of this research focused on the desires of stakeholders on the technical debt data.

Once the industrial context and critical concepts have been described, in the next section we will describe the methodological approach used to design a technical debt visualization tool to improve communication during decision-making processes at different levels of the organization to facilitate understanding of short-term benefits and long-term implications.

4 Methodology

This research is framed within the design science methodology which seeks the design and research of artifacts in a specific context [12]. Artifacts are designed to interact with a problem context to improve something in that context. A design science project iterates on two activities: (1) designing artifacts that enhance the context for stakeholders, and (2) empirically measuring their performance in the context to create knowledge.

The first step was a literature review both in academic repositories and industries best practices repositories. Also, practices in the company were observed, especially practices related to the software lifecycle decision-making processes. Moreover, the context of the company was studied.

Technical debt visualizations prototypes were designed to show the value and impact of improving communication between the different stakeholders in the decision-making process. Later, the designed artifacts were presented in detail to crucial actors involved with IT management in the company and were evaluated through a survey.

The main research question of this research is: How to improve communication and facilitate understanding of short-term benefits and long-term implications to support the business decision-making process at different levels of the organization through visualizations of technical debt?

To address this question, we tried to determine the decision-makers that would benefit from a technical debt visualization and their main concerns. Afterwards, we focused on identifying the main stages of the software lifecycle in which these visualizations could be useful.

4.1 Artifact Design Cycle

To design the technical debt visualizations, we conducted a series of steps:

1. We studied the company context, through document review and observation of decision-making processes.
2. We analyzed different actor's contexts, through the observation of their role and actions in the decision-making processes.
3. We determined best models to capture the reality of the company, such as existing technical debt models and good IT practices, and the industries where the company participates.
4. We tried to conceptualize technical debt, its management, and its representation model.
5. We designed possible visualizations (See Figs. 1 and 2).



Fig. 1. Visualizations built with the prototype. (a) Dynamic graphic of technical debt; (b) Which IT business services are affected by TD-Item 12?

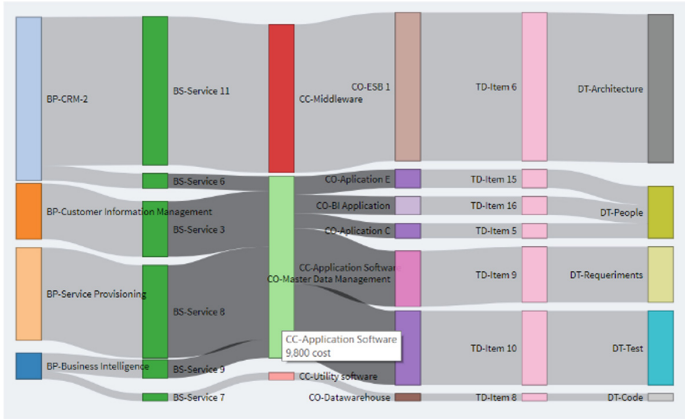


Fig. 2. Visualizations built with the prototype. How much technical debt do IT business service of Industry 2 have?

6. We built a prototype and generated visualizations examples using simulated data in the context of the company, which were used to present the concept of technical debt to stakeholders to obtain feedback from them.

4.2 Knowledge Generation Cycle

After the design cycle was completed, data was analyzed to determine the perceived usefulness of technical debt visualizations produced in the decision-making process. This evaluation was performed from the viewpoints of different stakeholders involved in the context of software development for the company. The evaluation process included:

1. Defining research questions to validate artifacts
2. Designing the instruments to obtain feedback from key stakeholders related to IT management
3. Present the technical debt concept and the visualization prototype to stakeholders
4. Collect feedback using the designed instrument and performing data analysis. To guarantee the future acceptance of the technical debt visualization tool, we were careful in the presentation of the concept and the visualizations provided to stakeholders.

We prepared a presentation about technical debt. This presentation is structured in three parts. In the first part, the central concepts about financial debt expressed in simple language are presented: What is debt? What is the purpose of the debt? What does the debt obligate? Problems generated by debt, what is bad debt? and What is good debt? The purpose of this part is to prepare the audience to facilitate its understanding of the technical debt metaphor. In the second part, we introduced the technical debt concept, challenges of technical debt, examples of technical debt incurred, associated concepts of technical debt as main and interest, why does technical debt matter?

Technical debt management framework and examples of artifacts to register technical debt were also explained. The last part included technical debt visualization examples for hierarchical data, relational data, spatial visualization and detail data presentation.

The presentation was attached to the invitation to complete the survey. We also presented it to a subgroup of the survey participants. During the presentation, they demonstrated an interest in the topic. Participants expressed that it could help very much during decision-making process in which people from the business are involved.

4.3 Survey

Participant opinions regarding the technical debt concept and the designed tool were collected through an online survey. The survey included a section to gather demographic data. The main questions were arranged in five groups to facilitate the answering process to participants:

1. Usefulness valuation of the technical debt concept in process/stage associated with software lifecycle.
2. Roles involved in decision-making processes of the software lifecycle and concerns in relation to technical debt.
3. Usefulness valuation of the visualizations structures.
4. Possible interactions and usefulness valuation of software for technical debt management.
5. Forms to quantify technical debt and feasibility of achieving it.

We invited 30 employees to complete the survey. They are involved in the software lifecycle and business IT solutions development and participate in IT governance. All the significant roles and positions were invited to participate.

5 Results and Findings

This section describes the results of the survey conducted in this research. Figure 3 shows the company's position of the 19 respondents and Fig. 4 shows the role of respondents; almost half of them have administrative roles and the rest are IT professionals. The member of the Board of Directors that answered the survey is an IT professional also, and his opinion is viewed as very valuable.

The roles for which we have the most participants are business management, architecture and development. More than 60% of respondents have worked in IT industry and in the company at least 15 years, and more than 60% of respondents have less than 15 years in their current position.

In this work we tried to determine which actors in the decision-making processes would require a visualization of technical debt. More than 50% of the respondents agreed that the following actors could be users of technical debt visualization during the decision-making processes: Business Management, Executive Level, Software Product Owner, Architect and Project Management.

Furthermore, we focused on the main concerns of these stakeholders. Figure 5 shows the level of concern of respondents of the relative importance of aspects about technical debt (stated as questions to be answered by a technical debt tool). For all the questions, except two, more than 50% of participants showed extremely or very important for them to be answered. The question with the highest accumulated answers of extremely and very important is: Which IT business services are affected by specific technical debt item?

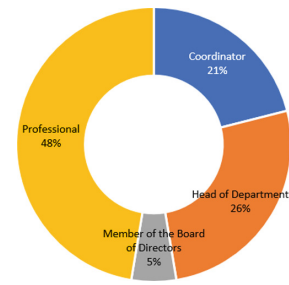


Fig. 3. Position of respondents (n = 19).

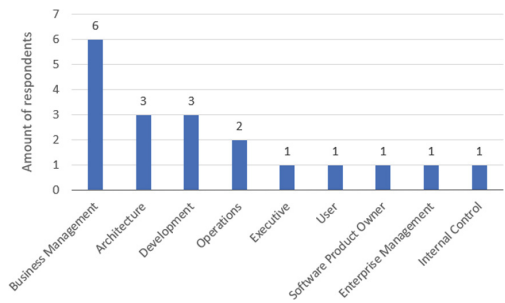


Fig. 4. Role of respondents

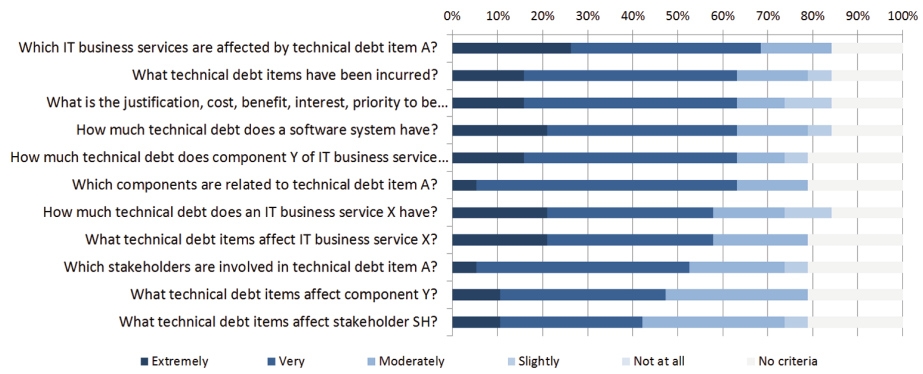


Fig. 5. Level of concern on each question

We also addressed the perceived usefulness of a tool like the one described in this paper. Figure 6 shows the results. The main finding is that the perceived usefulness of the visualizations is generalized. The question with the highest accumulated answers of extremely and very useful is: How much technical debt does an IT business service X have? It shows the perceived necessity to present the impact about IT decisions in business terms, to capture the attention from company's decision-makers. According with respondents, our technical debt visualizations should answer all stated questions.

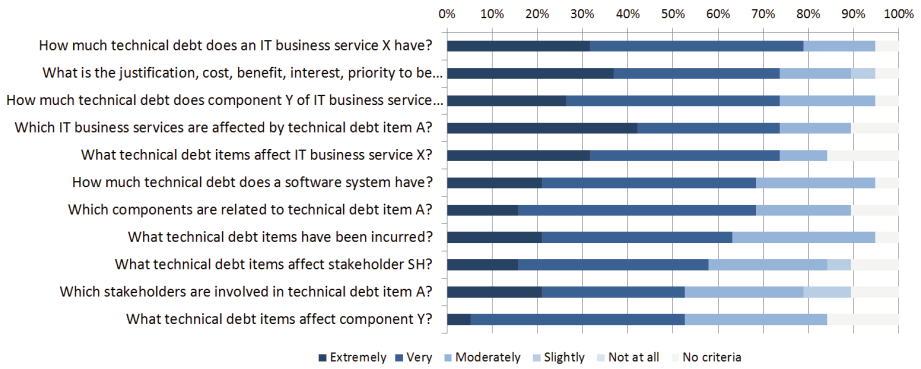


Fig. 6. Perceived usefulness of the visualization system that have the capacity to answer each question

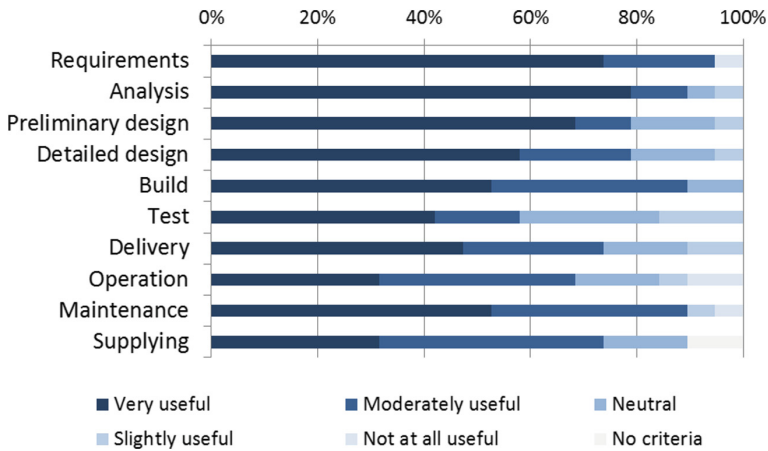


Fig. 7. Perceived usefulness of technical debt concept and its visualization for each stages/processes of the software life cycle

The perceived usefulness of the technical debt concept and visualization in decision-making processes for each stages/processes of software life cycle are shown in Fig. 7. For all the stages/process, more than 50% of participants consider very useful or moderately useful the technical debt concept and its visualization in decision making processes. Thus, we should consider all stages/processes in our designs.

After understanding the necessity and perceived usefulness of visualizations we focused on determining the “best” types of visualizations that allow decision-makers to understand technical debt. More than 80% of respondents consider diagrams/schemes and graphs as most useful presentation structures to understand concepts or problems. Figure 8 shows the perceived usefulness of visualization structures to demonstrate and

facilitate the understanding of technical debt in decision making. The top five (regarding accumulated extremely and very useful) are: tree, fishbone, hierarchical bars, Sankey and reference model.

More than 60% of participants consider feasible to quantify technical debt by amount, effort or qualitative with moderate, or a lot of effort, to gather the necessary information. Therefore, we should consider this issue in the process of developing the technical debt visualization tool. The preferred form to quantify technical debt was by debt amount, followed by effort. In relation with visualizations iterations, the options presented were: zoom-in, zoom-out, drill-in, drill-out, tooltip and selection. All interactions are deemed useful. The interaction less appreciated is zoom-out.

The main finding of our research is that all participants considered that the concept of technical debt is useful, except one that selected *No criteria* option. 89% of participants believed that having a software tool to manage technical debt is extremely or very useful. Thus, the development of technical debt visualizations for this company with a refined version of the proposed design seems to make sense.

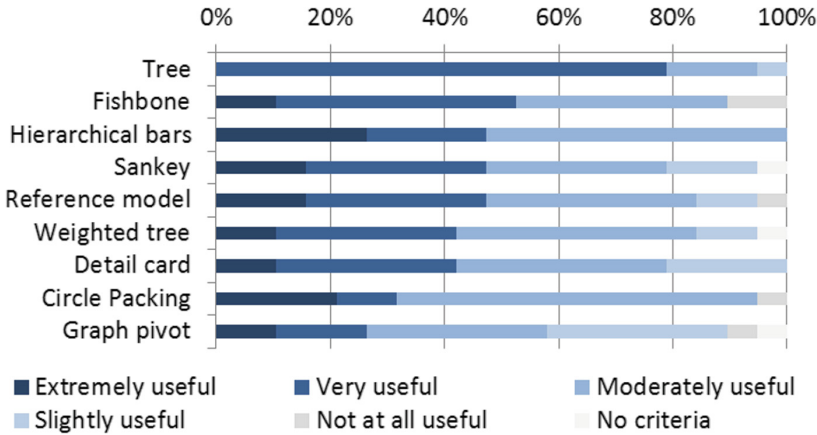


Fig. 8. Perceived usefulness of visualization structures

6 Conclusion and Future Work

To improve communication that supports the decision-making processes at different levels of the organization associated with the software life cycle is a challenge that companies must confront nowadays, especially because of digital transformation and converging technologies. In industrials sectors that have shorter and shorter time to market, many times products are being delivered too early, threatening a company's future wellbeing. The technical debt metaphor offers an opportunity to communicate trade-offs between the benefits in short terms and the impacts in long term. On the other hand, data visualization has demonstrated its ability to improve understanding of concepts and support decision making. Therefore, designing and building technical debt visualizations are crucial to support organizations in the digital transformation era.

To design technical debt visualization, we applied a design science approach. The process allowed us to make viable the acceptance of technical debt by the company's software development stakeholders. Technical debt metaphor has accepted as a valuable toll to reach a shared understanding with decision makers from other domains, on the long-term impacts of decisions.

Technical debt visualizations (artifacts) were designed to improve stakeholder communication to support the business decision-making process at different levels of the organization. The designed artifacts were evaluated empirically to get feedback from stakeholders. We used an on-line survey to gathered data which allowed us to improve the alignment of artifacts with the needs of company.

From this case study we conclude that design science approach helped us make viable the introduction of a novelty concept in this industrial context.

Furthermore, we concluded that the technical debt visualization contributes to improve the communication in the decision-making processes associated with the software lifecycle. Moreover, the main actors that would benefit from this type of visualizations of technical debt include executives, business manager, software product owner, architects, and project managers.

The main question that stakeholders want to answer is: how much technical debt does an IT business service have? This proves the necessity of describing technical debt in business terms rather than technical. On the other hand, visualization structures that are deemed to help facilitate the understanding of technical debt are: tree, fish-bone, Sankey and reference model.

As future work, first, we will improve visualization prototypes according the results obtained in this case study. Second, we will put into production the tool and evaluate technical debt visualizations usefulness in the real decision-making process at different level of organization.

Acknowledgments. This work was partially supported by Research Center for Communication and Information Technologies (CITIC) and the Doctoral Program at University of Costa Rica.

References

1. Cunningham, W.: The WyCash portfolio management system. In: Addendum to the Proceedings on Object-Oriented Programming Systems, Languages, and Applications, pp. 29–30. ACM Press, New York (1992)
2. Allman, E.: Managing technical debt. *Commun. ACM* **55**, 50 (2012)
3. Avgeriou, P., Kruchten, P., Ozkaya, I., Seaman, C.: Managing technical debt in software engineering. *Dagstuhl Rep.* **6**, 110–138 (2016)
4. Guo, Y., Spínola, R.O., Seaman, C.: Exploring the costs of technical debt management – a case study. *Empir. Softw. Eng.* **21**, 159–182 (2016)
5. Li, Z., Liang, P., Avgeriou, P.: Architecture viewpoints for documenting architectural technical debt. In: *Software Quality Assurance*, pp. 85–132. Elsevier (2016)
6. Guo, Y., Seaman, C., da Silva, F.Q.B.: Costs and obstacles encountered in technical debt management – a case study. *J. Syst. Softw.* **120**, 156–169 (2016)
7. Johannesson, P., Perjons, E.: *An Introduction to Design Science*. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-10632-8>

8. Munzner, T.: A nested model for visualization design and validation. *IEEE Trans. Vis. Comput. Graph.* **15**, 921–928 (2009)
9. Meyer, M., Sedlmair, M., Quinan, P.S., Munzner, T.: The nested blocks and guidelines model (2015)
10. Shahin, M., Liang, P., Babar, M.A.: A systematic review of software architecture visualization techniques. *J. Syst. Softw.* **94**, 161–185 (2014)
11. Fernández-Sánchez, C., Garbajosa, J., Yagüe, A., Perez, J.: Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. *J. Syst. Softw.* **124**, 22–38 (2017)
12. Wieringa, R.: *Design Science Methodology*. Springer, Berlin (2014). <https://doi.org/10.1007/978-3-662-43839-8>