European Journal of Information Systems

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjis20

Digital nudging for technical debt management at Credit Suisse

Kazem Haki, Annamina Rieder, Lorena Buchmann & Alexander W.Schneider

To cite this article: Kazem Haki, Annamina Rieder, Lorena Buchmann & Alexander W.Schneider (2022): Digital nudging for technical debt management at Credit Suisse, European Journal of Information Systems, DOI: 10.1080/0960085X.2022.2088413

To link to this article: https://doi.org/10.1080/0960085X.2022.2088413

Published online: 21 Jul 2022.

EUROPEAN JOURNAL OF INFORMATION SYSTEMS
https://doi.org/10.1080/0960085X.2022.2088413

THE OPERATIONAL RESEARCH SOCIETY

Taylor & Francis
Taylor & Francis Group

EMPIRICAL RESEARCH

OPEN ACCESS

Check for updates

Digital nudging for technical debt management at Credit Suisse

Kazem Haki[a], Annamina Rieder[b], Lorena Buchmann[c] and Alexander W.Schneider[c]

[a]Geneva School of Business Administration (HES-SO, HEG Genève), Switzerland; [b]University of St. Gallen, Switzerland; [c]Credit Suisse Group AG, Zurich, Switzerland

ABSTRACT

Technical debt (TD) is a technical compromise wherein the ability to maintain information technology (IT) applications over the long term is sacrificed for short-term goals. TD occurs when software development teams undergo constant pressure to release applications swiftly, on a tight schedule. The accumulation of TD, which often leads to a significant cost surplus, presents a ubiquitous challenge in technology-driven organisations. To keep TD levels under control, many organisations implement top-down mechanisms that impose enterprise-wide principles on software development teams. This clinical research presents a complementary but distinct approach to managing TD. A digital nudge was introduced at Credit Suisse, a global financial services company, to help raise awareness and understanding, and stimulate actions related to TD decision-making in software development teams. This paper reports on the nudge's clinical design, implementation, impact, and evaluation. As the nudge was effective in reducing TD in IT applications after one year of use, we demonstrate that digital nudges are viable means for guiding collective decisions in complex decision environments like that of TD management. Our findings have several implications for research and practice.

ARTICLE HISTORY
Received 14 September 2020
Accepted 6 June 2022

KEYWORDS
Technical debt; nudging; psychological effects; clinical research; design science; software development

1. Managing technical debt is crucial in technology-driven organisations

The financial services industry is on the front line of digital transformation (e.g., Chanias et al., 2019). As an integral part of economic systems, banks must ensure digital banking services are reliable to avoid business disruptions, but banks also pursue innovative operating models and services to keep up with constant changes in digital technologies. In this dynamic environment, software development teams are under pressure to release high-quality information technology (IT) applications on tight schedules (Austin, 2001). As pressure to bring business value and development complexity increase, software development teams may take shortcuts and workarounds that lead to *technical debt* (TD; Austin, 2001; Woodard et al., 2013). As such, TD management (TDM) decisions involve a trade-off between certain short-term outcomes vs. uncertain long-term outcomes, following a "now vs. later" logic (Ramasubbu & Kemerer, 2016; Siavvas et al., 2022). TD provides software development teams with the flexibility to release IT applications in a timely way by de-prioritising or postponing some actions to make room for more pressing ones (Rolland et al., 2018). However, excessive or uncontrolled TD or lack of awareness about its effects may affect an organisation's IT landscape negatively in the long run (Tom et al., 2013) by creating quality issues and increasing the cost of IT evolution (Ramasubbu & Kemerer, 2016; Woodard et al., 2013). Consequently, TD must be managed in such a way that it remains at a controllable level; the goal is not to eradicate TD but to manage it strategically.

Credit Suisse is a leading global financial services company founded and headquartered in Zurich, Switzerland. Supported by tens of thousands of IT employees and a budget that exceeds a billion-dollars, Credit Suisse's IT landscape contains thousands of IT applications. The size of the IT landscape and the large number of development projects leads to an enormous number of TDM-related decisions that are made daily. To guide these decisions, Credit Suisse has put in place top-down mechanisms to keep TD at an appropriate level. Given Credit Suisse's federated operating model and the short-term justifiability and case-by-case nature of TDM-related decisions, top-down mechanisms have been necessary but have also been insufficient means to manage TD.

This manuscript reports on the clinical design and evaluation of a *digital nudge* (Schneider et al., 2018; Weinmann et al., 2016) as a complementary mechanism for TDM. The nudge was developed and assessed by employees at Credit Suisse and, as of this writing, has been in use for more than a year, giving us sufficient time to track its impact.

CONTACT Kazem Haki kazem.haki@hesge.ch

2. Applying nudges to technical debt management

2.1. Challenges in managing technical debt

In technology-driven organisations, software development teams must meet short deadlines and high expectations for quality, which often necessitates a trade-off between business reality and software quality (Austin, 2001; Lim et al., 2012). These teams are under pressure to prioritise some tasks related to the development of an IT application (e.g., timely release) over others (e.g., code quality, maintainability) to meet business stakeholders' expectations. TD is the cost related to postponing such tasks. Cunningham (1992, p. 30) described TD and the associated risk 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. (Cunningham, 1992, p. 30)

TD arises from shortcuts and workarounds taken on the code level, as well as decisions made in other stages of the software development lifecycle. Examples include using outdated components in an IT application's development environment that can hinder development activities, lack of automated testing, and using manual build processes that make the build task time-consuming (Alves et al., 2016; Rios et al., 2018).

Inherent to TD is a trade-off between momentary benefits and future liabilities (Rolland et al., 2018). While incurring TD lends software development teams the leeway to meet business expectations (Kruchten et al., 2019; Siavvas et al., 2022), such as quickly and efficiently releasing novel IT applications (Li et al., 2015; Tom et al., 2013), excessive and uncontrolled levels of TD may be harmful to an organisation's IT landscape in the long run (Ramasubbu & Kemerer, 2016; Woodard et al., 2013). Incurring TD may result from carrying out tasks inadequately and making decisions that have short-term benefits but require significant maintenance later (Rios et al., 2018). Numerous small shortcuts that add up to TD can yield uncontrollable TD levels and negative consequences (Ramasubbu & Kemerer, 2016; Woodard et al., 2013), such as reduced productivity, quality issues, and premature loss of a system (Seaman et al., 2012; Tom et al., 2013; Yli-Huumo et al., 2016).

To address these challenges, TDM must consider activities that identify, measure, monitor, prevent, communicate, and reduce TD during software development (Fernández-Sánchez et al., 2017). Ways to measure TD include calculation models, code metrics, and human estimations (Li et al., 2015), but most approaches (e.g., Seaman et al., 2012) are highly sophisticated models that are too complex for use in practice (Fernández-Sánchez et al., 2017). Moreover, because of its invisibility to nontechnical stakeholders, communicating the business value of addressing TD remains a core challenge (Holvitie et al., 2018).

To manage TD, Credit Suisse applied such top-down mechanisms as dedicated funding for code-simplification activities, mandatory project reviews, and code quality gates. While these mechanisms offer high-level guidance for design decisions (Haki & Legner, 2021), the now-vs.-later trade-off persists, and timely delivery often remains the top priority for software development teams. Under certain circumstances, top-down mechanisms may even lead to conflicts with such competing objectives as tight delivery timelines and cost pressure.

Given the tactical value of TD, Credit Suisse's goal is not to eliminate TD from its IT landscape but to increase awareness and a shared understanding of TD among stakeholders. We expect that informed TD-related decisions will ensure that decisions to incur TD on the IT-application level are aligned with enterprise-wide strategic objectives. The formal procedures for TDM that Credit Suisse has in place require top-down enforcement mechanisms, but these mechanisms are sometimes circumvented by software development teams that are under tight deadlines. Hence, Credit Suisse lacked complementary mechanisms that address TD at its roots, that is, in the decisions software development teams make for individual IT applications, and mechanisms that software development teams would not consider as yet another constraint or formal procedure. Given these requirements, we sought to develop a *TDM nudge* that provides context-specific information about TD on the IT-application level and that supports software development teams in addressing TD in an unbinding fashion.

2.2. Nudges for managing technical debt

Cognitive psychology and behavioural economics suggest that individuals' decisions and choices are boundedly rational in that they may rely on heuristics and mental shortcuts that may lead to systematic errors and biased decisions (Kahneman, 2003). For example, individuals are inclined to stick to the current situation even when making a change might be more favourable, a tendency known as the status quo bias (Kahneman et al., 1991). Turning away from the status quo is often perceived as containing losses that exceed the status quo's obvious disadvantages.

Nudges are "any aspect of the choice architecture that alters people's behaviour in a predictable way without forbidding any options or significantly changing their economic incentives" (Thaler & Sunstein, 2009, p. 6). Thus, nudges present a subtle approach to guiding individuals towards better decisions while preserving their freedom of choice (Thaler & Sunstein, 2009). Nudges use or counteract the boundaries in individuals' decision-making in a targeted way and are often used in digital environments characterised by less costly implementation, evaluation, adjustment, and personalisation and high information loads (Weinmann et al., 2016).

Nudges have sparked interest because they offer a simple, impactful, and inexpensive alternative to conventional practices like laws and codes of conduct that are enforced in a top-down manner (Johnson et al., 2012). They have proven effective in situations in which decisions and their consequences are separated in time and when feedback on choices is delayed or infrequent (Ölander & Thøgersen, 2014; Thaler & Sunstein, 2009). As a way to target decisions, nudges fit particularly well with the TD-related decisions that must be made for individual IT applications at the level of the software development team.

Scholarship has brought forward myriad psychological effects (or heuristics and biases) that nudging may use. In our clinical design of the TDM nudge, we opted for the psychological effects that underlie nudging (cf., Mirsch et al., 2017) that fit well in the TD context (Table 1).

Table 1. Selected underlying psychological effects of nudging.

Effect	Description
Framing	The presentation of a decision problem can influence choice. Attribute framing highlights specific attributes or characteristics of a decision option (Levin et al., 1998). In contrast, goal framing highlights the relationship between specific actions and their respective outcomes (Tversky & Kahneman, 1981).
Social norms	Norms are beliefs and actions that are regarded as normal and acceptable by a specific social group. Descriptions of others' behaviours lead individuals to act in conformance to avoid standing out. Individuals also base their actions on what they anticipate others will think of them. In aiming to achieve conformance, they tend to choose behaviours that others favour (Cialdini & Trost, 1998).
Messenger effect	Information is evaluated according to the recipient's perception of the authority of its source. Information that is delivered by a credible messenger is likely to be perceived as true (Dolan et al., 2012).
Anchoring	When estimating unknown values, people are guided by starting points (i.e., anchors). The starting value is then adjusted to yield the final estimate, resulting in estimates that are biased towards the starting point (Tversky & Kahneman, 1974).
Availability heuristics	Estimates about the probability of an event are based on the ease of finding examples of the event, which may be contingent on factors like public visibility or individual experience (Tversky & Kahneman, 1974).
Loss aversion	Losses and disadvantages are given more weight than equivalent gains and advantages, so losses and disadvantages are usually avoided (Kahneman et al., 1991).

3. Developing a nudge for technical debt management at Credit Suisse

In our clinical design and evaluation of a TDM nudge at Credit Suisse, we adopted a design science research (DSR) paradigm (Peffers et al., 2007). The goal was to design a TDM nudge to direct software development teams' awareness of TD and induce them to make conscious decisions that take TD into account. We built the design elements of the TDM nudge on the psychological effects from the nudging literature (Table 1). We iteratively adjusted the nudge based on feedback we received from colleagues at Credit Suisse in multiple rounds. (See Appendix A for details about the research approach.) What follows provides an overview of the TD measures and the design elements of the TDM nudge at Credit Suisse.

3.1. Technical debt measures at Credit Suisse

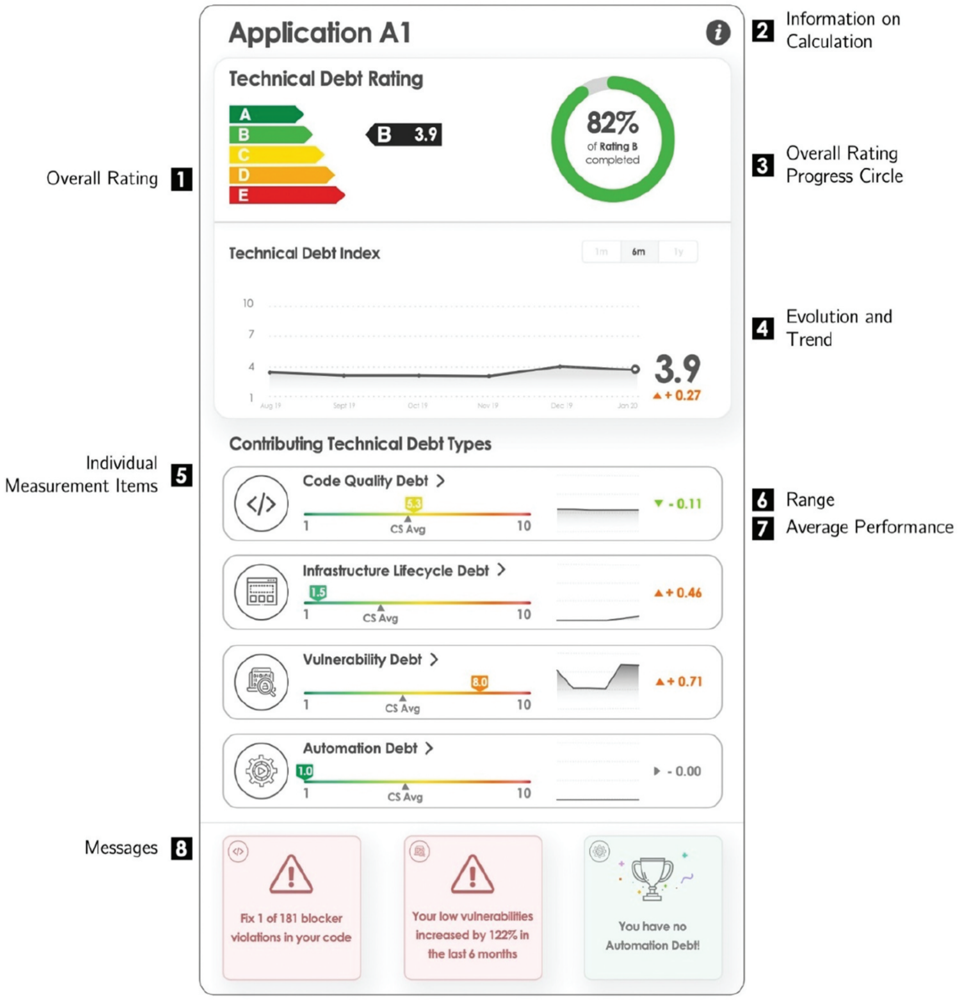
Adapted from existing discourse on TD measures (Fernández-Sánchez et al., 2017), the four types of TD that Credit Suisse manages across its IT landscape are code quality debt, infrastructure lifecycle debt, vulnerability debt, and automation debt. *Code quality debt*, such as code smells and defects, are identified by SonarQube, one of the most popular tools for measuring code-related debt (Avgeriou et al., 2020). *Infrastructure lifecycle debt* concerns hardware or software components that exceed the period in which they have vendor support. Infrastructure lifecycle debt is measured by counting the number of days without support for all infrastructure instances. *Vulnerability debt* is identified systematically by network scanners and code analysis tools and measured by counting the number of vulnerabilities and weighting them based on their criticality (i.e., high, medium, low). Finally, *automation debt* concerns the degree of build and deployment automation. It is measured on a maturity scale from 1 to 10 that considers the use of build pipelines, the degree of build processes' automation, and deployment processes for test and production environments.

While counting TD items and weighting them based on their impact is a common approach to measuring TD (Li et al., 2015), the TDM nudge we developed required comparability of individual IT applications. At Credit Suisse, IT applications differ in code size, infrastructure footprint, cost, and business functionality by up to four orders of magnitude, so, to facilitate simple comparisons among IT applications, we transform these metrics into a logarithmic scale that ranges from 1 to 10.

3.2. Design elements of the technical debt management nudge

The TDM nudge consists of eight design elements (DEs) that are geared to creating software development teams' awareness of TD and nudging them to manage TD in their IT applications. Figure 1 shows the TDM nudge for an exemplary IT application, A1. The DEs and their underlying psychological effects are summarised in Table 2.

We implemented the TDM nudge in a digital form building on a visualisation component and a data-processing component. Because of the clinical setting, we followed the principle that existing solutions, processes, and data at Credit Suisse should be leveraged whenever possible. Therefore, we used the *Tableau* analytics tool, which was already in use at Credit Suisse, to visualise the TDM nudge. We had to collect, integrate, and aggregate data from various tools to calculate several variables needed for the DEs (Table A7 in the Appendix). We employed an existing data platform at Credit Suisse and included two variables: those for individual IT applications and those that allowed for comparisons between multiple IT applications. First, we calculated variables for each IT application separately using the index values for each type of TD (DE05; Table A8 in the Appendix), the overall TD index value (DE01) that is the equal-weighted average of the four TD type indexes, and the colouring of the range for each type of TD (DE06). Notably, the calculation results had to be stored in each calculation run to generate evolution graphs (DE04). This step is indispensable if a source system does not archive the data collected by the TDM nudge. Warning and achievement messages (DE08) were also derived from the calculation results. Second, we calculated variables that require comparisons between IT applications, including the overall TD rating of the given IT application (DE01; Table A6 in Appendix), the overall TD rating's progress (DE03), and the average index values of all IT applications for each type of TD (DE07). More details on the implementation of the TDM nudge appear in Appendix D.



Fix 1 of 181 blocker violations in your code

Your low vulnerabilities increased by 122% in the last 6 months

You have no Automation Debt!

Figure 1. TDM nudge for a sample IT application, A1.

Table 2. Design elements and their underlying psychological effects.

No.	Design Element	Description	Instantiation of Psychological Effect in the Design Element
01	Overall rating	The overall TD rating of an IT application is communicated with a ranking (A to E) and colour code (best = green, worst = red).[1] Building on a TD index that ranges from 1 to 10 (1 = low debt; 10 = high debt) for each IT application, the overall rating is calculated based on pre-defined thresholds (Table A6 in the Appendix). The TD index is the equal-weighted average of the four TD measurement items. (See DE05.)	**Framing**: The ranking and colour code make an IT application's TD level salient, thus simplifying and drawing attention to the attribute. **Social norm**: The rating provokes comparison to the TD levels of other IT applications at Credit Suisse and indicates the social undesirability of high TD.
02	Information on calculation and data sources	Users can click the "i" to see details on the underlying calculations and data sources.	**Messenger effect**: The "i" sign and information about the computation signal reliability and increase the credibility of the TDM nudge.
03	Overall rating progress circle	Provides detailed information on the completion of the overall rating.	**Framing**: The progress circle emphasises the percentage of a rating's completion. The goal of achieving a higher rating is highlighted by displaying the effort left to achieve it. **Loss aversion**: The progress circle plays on the tendency to avoid a drop in the rating because of loss aversion.
04	Evolution and trend	Provides a visualisation of the evolution and trend of the overall TD index *and* for each type of TD. (See DE05.) The evolution graph can be adjusted to specific time frames (e.g., one month, six months, one year)	**Loss aversion**: The trend data encourages actions that support a positive evolution vs. a downward trend that will be considered a "loss".
05	Individual measurement items	The four measurement items (i.e., types of TD) are presented as an index normalised on a 1–10 scale (1 = low debt; 10 = high debt).	**Framing**: The accumulated TD for each measurement item is translated to a simplified form to reduce the cognitive effort. This DE also guides users in their TDM-related decisions regarding on which type of TD to focus first (considering the rating of each item).
06	Range	The rating for each type of TD is shown not only as a number from 1 to 10 but also on a colour-coded range and uses an arrow box (an anchor point) to indicate the rating for each type of TD.	**Anchoring**: The representation of the TD level by type serves as an anchor based on which TDM-related decisions will be made, adjusting from the salient reference point.
07	Average performance of other teams	Shows the average performance of all IT applications for every type of TD. The nudge includes a marker for each type of TD (formulated as "CS Average") that shows the average amount of TD for all of Credit Suisse's IT applications for the given type of TD.	**Social norm**: Stimulates comparison across software development teams and signals the norm for TD level in each item.
08	Messages	When the metrics reach a certain threshold, alert messages with TD focal points and/or achievements can be configured for the four measurement items (DE05).	**Availability heuristic**: Cues, alerts, and visual highlights help individuals to keep important TD-related topics top of mind.

Thanks to the nudge's digital form, all collection, integration, aggregation, and calculation activities are fully automated and conducted daily, so the TDM nudge presents an accurate and up-to-date TD status of all IT applications. The TDM nudge is available for more than 3,000 IT applications––almost the entire IT landscape of Credit Suisse.

4. Evaluating the impact of the technical debt management nudge

We evaluated the impact of the TDM nudge considering both its usefulness and its effects on TD levels at several points in time. To assess its *usefulness*, we conducted semi-structured interviews, focus group workshops, and surveys before and after the nudge was implemented. Almost a year after implementation, we assessed the *effects* of the TDM nudge on the IT applications' TD levels with selected software development teams. To assess the effects, we quantitatively tracked the evolution of TD levels and enriched the tracking data with survey and interview data. (Evaluation details appear in appendices B and C.)

4.1. Usefulness of the technical debt management nudge

The first evaluation step assessed the usefulness of the nudge and its reception by software development teams, who were the target audience for the nudge. We conducted semi-structured interviews with senior IT architects and IT managers, focus group workshops with software development teams, and surveys with software developers before and one week after the nudge's implementation to facilitate within-subjects comparison. We focused our evaluation on TD awareness, understanding, and action (Schilling et al., 2019).

We found that the TDM nudge was effective in raising TD awareness at the individual and team levels, as demonstrated through the survey (Appendix B) and the interviews. The interviews also revealed that the TDM nudge was helpful in directing attention to TD, making different types of TDs visible, and providing information about the teams' TD status. One divisional CIO summarised the nudge's affordances as: *"With this TDM nudge we are able to track the most important aspects of TD"*. (Divisional CIO)

We also found that the TDM nudge helped software development teams improve their understanding of TD-related issues and facilitated communication within teams and between internal stakeholders. While TD had been a vague term with little shared understanding across teams or functions, defining, standardising, and visualising types of TDs generated a global syntax of TD at Credit Suisse. Thus, the simplified visualisation facilitated communicating TD-related topics with other stakeholders by making TD comprehensible. The interviewees indicated that monitoring the evolution of TD levels can help the software development teams with their internal discussions and goal-setting regarding TD repayment and remediation. As one software developer explained: *"In terms of discussions, I think it especially supports because of the graph. That's the big plus. For instance, I want to see three weeks ago and today whether the technical debt has increased or decreased so that will help support the discussions in retrospective meetings of the development teams"*. (Software Developer)

Finally, the TDM nudge motivated software development teams to manage TD actively. Besides mentioning their greater understanding of the types of TDs and the need to manage and maintain TD, the interviewees frequently mentioned the alternative motivation mechanisms that the nudge provided. Allowing for cross-comparison and indicating the social desirability of certain behaviours (i.e., social norms) turned out to be important in shaping the teams' TDM motivation: *"If you bring people to see their own rating, then obviously people want to do things to improve their rating"*. (IT Architect)

Moreover, the TDM nudge served as a benchmark and management tool for the owners of IT applications: *"As an application owner you should have the motivation that your application is at least C or better. You don't want to be orange or red unless you know that the application will be decommissioned next year, then it's okay"*. (Chief Architect)

4.2. Effects on technical debt levels

After the TDM nudge was in use for about a year, we conducted a follow-up evaluation of effects on TD levels. We analysed the change in TD levels of the IT applications of the three software development teams that participated in the first evaluation step. Table 3 provides an overview of the changes in TD levels of the individual IT applications of the three teams. While quantitative tracking of TD levels was our primary data source, we enriched our data using another survey of the software development team members and semi-structured interviews with IT managers and architects. The survey asked questions regarding the frequency of and purposes for using the TDM nudge and about respondents' perceptions of the impact on TDM-related decision-making. The purpose of the interviews was to collect the managers' views of the use and impact of the nudge.

Table 3. Changes in TD levels by team and application.

	Δ Team 1	Δ Team 2	Δ Team 3	
TD item	Application 1	Application 2	Application 3.1	Application 3.2
Overall TD	− 0.2	− 1.2	+1.4	+0.7
Code quality debt	− 0.7	±0.0	±0.0	+1.0
Lifecycle debt	±0.0*	±0.0*	+2.6	±0.0*
Vulnerability debt	±0.0	− 4.6	+3.1	+1.9
Automation debt	±0.0*	±0.0*	±0.0*	±0.0*

TD was measured on a scale from 1 (=lowest) to 10 (=highest).
Values in the table show the change from implementation of the nudge to one year post-implementation.
±0 = there was no change in TD level; ±0* = the TD was 0 and no TD was added

The TDM nudge had an impact on teams 1 and 2, which reduced both overall TD and individual types of TDs. The reduction of TD was corroborated by respondents' statements indicating that they used the TDM nudge to check the progress of their remediation activities: *"The right place to use it is in sprint retrospectives when the development teams are anyway reflecting on what they did during the last two or three weeks and then they can look at it and see what has changed. Did the technical debt grow or did they take something back? If nothing changes, then you do not need to look at it [the TDM nudge] on a weekly basis"*. (Enterprise Architect)

For teams 1 and 2, the TDM nudge served as a reminder of the importance of keeping track of TD in the software development process: *"Maybe there are applications where it is perfectly fine to increase technical debt for a couple of weeks or months because there is high market pressure to get some capabilities out the market as quickly as possible. So it is not good but acceptable if you increase your debt. So you should be aware of it so that you can pay it back later on"*. (Enterprise Architect)

In addition, the survey and the interviews revealed that teams 1 and 2 followed different approaches and set different priorities in reducing TD in their IT applications. Team 1 decided to tackle the different types of TDs sequentially, starting with code quality debt: *"IT owners are going to say, 'guys code quality we have to improve it' but the consequences are not so obvious yet. Lifecycle is obvious, vulnerability is obvious, automation is fairly obvious right? These are things which you can look at and change your behaviour"*. (Enterprise Architect)

In contrast, team 2 eradicated vulnerability debt from its IT application by consolidating some legacy servers and shutting down others. Team 2 confirmed that code quality debt would be next and that they were already laying out a plan for how to address it over the next couple of months. Hence, while the TDM nudge accomplished the goal of guiding software development teams towards tackling TD, the specific trade-off decisions and approaches regarding how to do so remained at the team level.

Unlike those of teams 1 and 2, team 3's two IT applications' TD levels considerably increased both overall and across the types of TDs. The follow-up survey revealed that, unlike the other teams, maintenance or remediation of IT applications' TD was not among the priorities of team 3 or its managers, so they did not use the TDM nudge. Despite their favourable reception of the TDM nudge in the first evaluation step, most developers in this team did not consider using the nudge after its roll-out and did not take any TD-related action.

5. Discussion

This clinical research designed and evaluated a nudge at Credit Suisse that was intended to create awareness and understanding among software development teams of the need to manage TD in IT applications. While Credit Suisse has had several top-down mechanisms for TDM in place, we designed a nudge to target software development teams' decision-making, which would complement the existing mechanisms. By employing various design elements informed by nudge theory, we developed a TDM nudge to gently cue software development teams towards desired TDM-related decisions while preserving their design freedom. We found that the nudge raised awareness, created a common understanding, and stimulated active TD-related decision-making at the team level. Contingent on the support by management and software development teams, the TDM nudge effectively reduced the amount of TD in IT applications within a year of implementation.

In the following subsections, we share the scholarly contributions of our research to the TDM and nudging literature. Specifically, we highlight the key contributions concerning the use of nudges for TDM and for complex and collective decision-making. Finally, we discuss the study's implications for research and practice.

5.1. Nudges for decisions related to technical debt management

The scholarly literature and TDM practice have produced a plethora of measures for TD and mechanisms for managing it. A central issue for these measures is their complexity and, thus, the challenge of transferring them to use in practice (Fernández-Sánchez et al., 2017; Tom et al., 2013). In turn, the mechanisms for managing TD (e.g., software architecture principles, coding standards) are typically administered from the top down (e.g., Besker et al., 2022; Green & Ledgard, 2011; Yli-Huumo et al., 2016) and, despite the anecdotal effectiveness of top-down mechanisms, they have been criticised for restricting design freedom and yielding unfavourable decisions and outcomes (Haki et al., 2020). Our study provides pioneering evidence that nudges can be an effective complementary approach to managing TD and bridging knowledge and communication boundaries between stakeholders. In particular, nudges are easy-to-apply tools in TDM practice that complement top-down mechanisms because of their *non-binding* and *simple* properties and their use as *boundary objects.*

In contrast to top-down mechanisms that govern and constrain software developers' design freedom (Haki & Legner, 2021), the TDM nudge preserves decision-makers' design freedom. We follow Thaler and Sunstein (2009), who postulate that nudges must be both easy to follow and easy to avoid. Hence, as a relatively soft and non-intrusive intervention intended to steer software development teams towards desirable TDM-related decisions, the TDM nudge relies on a libertarian paternalistic approach (Thaler & Sunstein, 2009).

Given the multifaceted nature, complexity, and dynamism of decisions in the TDM context (Rolland et al., 2018), paired with individuals' limited cognitive capacity (Jacoby, 1984), reducing complexity is essential. Existing approaches have been more concerned with developing a holistic and in-depth picture of the various types and sources of TD (e.g., Li et al., 2015; Ramasubbu & Kemerer, 2016). Using a TDM nudge simplifies the context of TD-related decision-making for software development teams. While our findings show that a TDM nudge makes TD-related information comprehensible and accessible, it also facilitates engaging in specific actions to reduce TD because of its intuitive and simple design for choice architecture. Thus, complementing the extant research that has been criticised for focusing primarily on identifying and measuring TD (e.g., Li et al., 2015; Yli-Huumo et al., 2016), our use of a TDM nudge addresses higher-level TDM activities (cf., Li et al., 2015), such as developing a shared understanding of TD, facilitating decision-making, and stimulating repayment of TD.

Knowledge and communication boundaries between different stakeholders who are involved in software development are a key challenge in TDM (Doolin & McLeod, 2012; Huber et al., 2020). Boundaries can take various forms (Carlile, 2002). For example, software development teams may lack a shared vocabulary to ensure accurate TD-related communication and understanding of the consequences of incurring TD. This insufficient shared syntax between different team members (e.g., product owner, scrum master, and software developer) may hinder the highly collaborative work in software development, particularly in an agile context (Schwaber & Sutherland, 2020). The TDM nudge we implemented was effective in bridging the communication boundaries among members of two software development teams. The pre-and post-implementation survey results at Credit Suisse demonstrated a noticeable difference in software developers' awareness and understanding of types of TDs and showed that the nudge facilitated their discussions about the existence and location of TD. Boundaries for communication and collaboration can also arise between technical and business stakeholders (Klinger et al., 2011) because of their inconsistent interpretations of TD and divergent concerns (Gal et al., 2008; Levina, 2005). Our study suggests that nudges for TDM can be employed as a communication artefact to bridge knowledge boundaries between business and technical stakeholders and to help them reach a common understanding of TD. The TDM nudge we employed helped in overcoming such contrasting perspectives. For example, business stakeholders may demand high-quality software in a timely manner, whereas software developers may focus on technical elegance (Siavvas et al., 2022). The pre- and post-implementation survey results at Credit Suisse demonstrated that the TDM nudge helped to make TD explainable to business stakeholders and created a shared understanding among diverse functions.

Finally, the TDM nudge stimulated competition among software development teams. Social norms are a potent mechanism by which to induce behaviour change, as individuals tend to orient towards the behaviour of successful individuals (DiMaggio & Powell, 1983; Sunstein, 2014). Presenting the average performance of other IT applications gently pushes software development teams to mimic successful teams' actions to improve their TD measures.

5.2. Nudges for complex and collective decisions in organisations

Nudges have been applied traditionally to guide individuals' choices in non-complex and ad-hoc decision situations. The few examples of nudges applied in more complex decision situations relate primarily to the financial context, such as retirement savings (Thaler & Benartzi, 2004) and investment decisions (Gajewski et al., 2022). In expanding the applications of nudges, our study substantiates the potential success of nudges that target complex organisational decisions that feature conflicting objectives, a multiplicity of interests (e.g., business vs. technical stakeholders), high uncertainty (i.e., trade-off between unforeseeable outcomes), and long-term consequences (e.g., code evolvability (Rolland et al., 2018)).

Moreover, given the foundations of nudging in cognitive psychology, nudges have been applied primarily to individuals' decision-making, typically that of consumers (e.g., Dennis et al., 2020; Thomas et al., 2021), employees (Wu & Levy Paluck, 2021), end-users (e.g., Choudhary et al., 2022; Pennycook et al., 2020), or citizens (e.g., Fonseca & Grimshaw, 2017). Applying nudges at the team level in an organisation is a novel approach. Yoon et al. (2019) suggested using nudges to guide collective decision-making in the context of infrastructure development, but while that study highlighted the challenges of involving actors from diverse areas of expertise and social and political dynamics, it did not implement and empirically test the proposed nudge scenarios. Similarly, Galpin (2022) established a connection between the elements of a choice architecture (e.g., values, goals, policies, rules) and an organisational innovation culture, but the study did not focus on testing the effects of individual nudges. Another study that tested nudges' ability to enhance collective intelligence did not find significant effects (Gupta et al., 2019). Therefore, our findings pioneer the successful application of nudges to guide collective behaviour in organisations.

However, extending the application area of nudges from simple, ad-hoc, and individual to complex organisational (respectively, collective) decisions poses additional requirements for their design and implementation. What we learned from the clinical design of our TDM nudge suggests three points to be considered when designing nudges for complex and collective decisions in organisations. First, to enable high-quality decisions, the quality and accuracy of the underlying data, assumptions, and calculations must be ensured (e.g., DE02 in Figure 1). Disclosing such information will allow decision-makers to account for the multifinality of a complex decision problem, to gain trust in the information provided by the nudge, and to increase the likelihood that the information will be incorporated into TD-related decision-making. Second, decision-makers must be able to retrieve the information that is relevant to their individual situations and contexts, so adaptive features that enable them to browse information, drill down, explore, and customise outputs must be offered (e.g., DE04 and DE05 in Figure 1). Third, for nudges to show effect, decision-makers, whether organisations or individuals, must be sufficiently motivated to pursue the target behaviours (Fogg, 2009). In organisational settings, not only must individual decision-makers be motivated but management must support and encourage the organisation's members to follow the direction of the nudge (Besker et al., 2022). Our results show a significant divide in the TDM nudge's impact, contingent on the team's and the management's intent to address TD. While teams 1 and 2 decided to address TD in their IT applications with the help of the TDM nudge, team 3 had no intention either to use the TDM nudge or to eliminate TD. Motivation and intent were reflected in the nudge's impact on TDM levels: While teams 1 and 2 reduced TD in their IT applications, the TD levels in team 3's applications increased.

5.3. Implications for research and practice

Our study of the use of a TDM nudge at Credit Suisse has several practical and research implications. The first set of implications relates to the level at which the TDM nudge is directed. In the case of Credit Suisse, the goal was to complement existing mechanisms for managing TD by means of non-binding interventions that also preserve enterprise-wide and holistic interests. To this end, we opted for a nudge that focused on *single* IT applications that were developed and maintained by a team of software developers. However, an enterprise IT landscape is a complex system that contains a multitude of interactions and interdependencies among a plethora of IT applications (Haki et al., 2020). Consequently, optimising each IT application's TD does not necessarily lead to optimisation of the entire IT landscape. Therefore, practitioners might consider taking a holistic perspective in managing TD as a portfolio and through an architectural standpoint, in addition to TDM nudges that are directed at individual IT applications. Such an architectural TDM approach would, for example, enable an IT application with a critical release date to incur TD if another IT application in the same portfolio had little TD. Similarly, following an architectural TDM approach would help to unify functionally redundant IT applications instead of decreasing the amount of TD in each application separately. Such a scaled approach may be particularly beneficial for large organisations with IT landscapes of several hundreds or thousands of IT applications.

To set in place an architectural TDM approach, prospective research is encouraged to focus on higher-level (beyond single IT applications) TDM nudges and to elaborate on procedures for aggregating multiple IT applications' TD measurement metrics into groups of inter-related IT applications (based on, for example, organisational units, customer segments, product lines, business capabilities, business domains, or business processes). These aggregation procedures should include additional criteria, such as IT application size and business criticality, to account for the considerable differences between IT applications. Finally, a higher-level TDM nudge targets different decision-makers and may require different design elements and use of different psychological effects than the TDM nudge developed in this study. Hence, future research is encouraged to develop and evaluate such higher-level interventions.

The second set of implications can be derived from the clinical research approach we used. Researchers may make extended use of the benefits of testing solutions and designing artefacts in the field, and in organisations specifically. Several IS scholars have emphasised the value of contextualisation: in relation to theory-building to enhance explanatory power (Avgerou, 2019), and to design artefacts that are purposeful and effective in solving underlying problems (Hevner et al., 2004). Thus, scholars can make use of organisations' internal skills, resources, and in-depth knowledge of their contexts to develop solutions for the organisations' own problems. Internal stakeholders who work on a solution for a problem tend to be familiar with their organisations' internal processes and cultural idiosyncrasies, thus increasing the chances of success. Indeed, the results of our evaluation suggest that considering the contextual and situational factors at the team level is key to the successful deployment of a TDM nudge. Even though both teams 1 and 2 addressed the issue of TD in their IT applications using the TDM nudge, they took different routes and prioritised differently based on their contexts and knowledge.

Moreover, because of internal structures, organisational contexts may give rise to differences in behaviours and outcomes between organisational units (e.g., teams, business units, subsidiaries), which can be used for comparison purposes in a clinical setting. In our study, team 3 provided a useful basis for cross-comparison facilitating our exploration of why the TDM nudge had not been successful for their IT applications. This comparison not only allowed us to observe differences in effectiveness, but also to find reasons for these differences. The lack of intent to tackle TD through the TDM nudge and the lack of management support in team 3 critically influenced the effectiveness of the TDM nudge. Clinical research scholars could exploit such naturally occurring between-group-differences to obtain insight into the factors and processes that give rise to deviations in behaviours and outcomes.

However, conducting clinical research in an organisation comes with certain challenges, such as the interdependencies between the levels of decision-making. Researchers must consider the level at which an intervention is supposed to have an effect and the level at which decisions and behaviours can be targeted. As our study demonstrates, even for an enterprise-wide topic like TDM, decision power resides at the team level. Therefore, interventions must be tailored accordingly, and the effects must be tracked on the level at which the intervention is applied.

6. Conclusion

We clinically designed, implemented, and evaluated the impact of a digital nudge for managing TD at Credit Suisse. Following the design, which was based on established nudging principles, and the literature on psychological effects, we implemented the nudge and tested its impact on TD-related decision-making at the level of software development teams shortly after implementation and one year later. We found that the nudge was effective in raising awareness and creating a shared understanding regarding TD and that it yielded a reduction in TD at the level of individual IT applications. Our results suggest that, mainly because of their simplicity, non-binding nature, and their function as boundary objects that bridge communication boundaries, nudges are helpful tools with which to guide TD-related decision-making. Apart from TDM, our findings contribute to the nudging literature by demonstrating that nudges are not only effective in influencing non-complex, ad-hoc decisions at the individual level, but can also guide complex decisions made in teams in organisations.

Note

1. This design feature is inspired by the successful framework for the European Union's energy labelling and its use in an enterprise context (Schilling et al., 2019).

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Kazem Haki http://orcid.org/0000-0002-8807-0147

References

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. https://doi.org/10.1016/j.infsof.2015.10.008

Austin, R. D. (2001). The effects of time pressure on quality in software development: An agency model. *Information Systems Research*, *12*(2), 195–207. https://doi.org/10.1287/isre.12.2.195.9699

Avgeriou, P. C., Taibi, D., Ampatzoglou, A., Fontana, F. A., Besker, T., Chatzigeorgiou, A., Lenarduzzi, V., Martini, A., Moschou, N., & Pigazzini, I. (2020). An overview and comparison of technical debt measurement tools. *IEEE Software*, *38*(3), 61–67. https://doi.org/10.1109/ms.2020.3029468

Avgerou, C. (2019). Contextual explanation: alternative approaches and persistent challenges. *MIS Quarterly*, *43* (3), 977–1006. https://doi.org/10.25300/MISQ/2019/13990

Besker, T., Martini, A., & Bosch, J. (2022). The use of incentives to promote technical debt management. *Information and Software Technology*, *142*, 106740. https://doi.org/10.1016/j.infsof.2021.106740

Brinkmann, S. (2013). *Qualitative interviewing*. Oxford University Press.

Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: boundary objects in new product development. *Organization Science*, *13*(4), 442–455. https://doi.org/10.1287/orsc.13.4.442.2953

Chanias, S., Myers, M. D., & Hess, T. (2019). Digital transformation strategy making in pre-digital organizations: The case of a financial services provider. *The Journal of Strategic Information Systems*, *28*(1), 17–33. https://doi.org/10.1016/j.jsis.2018.11.003

Choudhary, V., Shunko, M., Netessine, S., & Koo, S. (2022). Nudging drivers to safety: evidence from a field experiment. *Management Science* 68(6): 4196–4214. https://doi.org/10.1287/mnsc.2021.4063

Cialdini, R. B., & Trost, M. R. (1998). Social influence: social norms, conformity and compliance. In D. T. Gilbert, S. T. Fiske, & L. Gardner (Eds.), *The handbook of social psychology* (pp. 151–192). McGraw-Hill.

Cunningham, W. (1992). *The WyCash portfolio management system. Addendum to the proceedings on Object-oriented programming systems, languages, and applications*, Vancouver, British Columbia, Canada. Association for Computing Machinery (ACM).

Dennis, A. R., Yuan, L., Feng, X., Webb, E., & Hsieh, C. J. (2020). Digital nudging: numeric and semantic priming in e-commerce. *Journal of Management Information Systems*, *37*(1), 39–65. https://doi.org/10.1080/07421222.2019.1705505

DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, *48*(2), 147–160. https://doi.org/10.2307/2095101

Dolan, P., Hallsworth, M., Halpern, D., King, D., Metcalfe, R., & Vlaev, I. (2012). Influencing behaviour: the mindspace way. *Journal of Economic Psychology*, *33* (1), 264–277. https://doi.org/10.1016/j.joep.2011.10.009

Doolin, B., & McLeod, L. (2012). Sociomateriality and boundary objects in information systems development. *European Journal of Information Systems*, *21*(5), 570–586. https://doi.org/10.1057/ejis.2012.20

Fernández-Sánchez, C., Garbajosa, J., Yagüe, A., & Perez, J. (2017). Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. *Journal of Systems and Software*, *124*, 22–38. https://doi.org/10.1016/j.jss.2016.10.018

Fogg, B. J. (2009). *A behavior model for persuasive design. Proceedings of the 4th International Conference on Persuasive Technology*, Claremont, California, USA, 40, 1–7. Association for Computing Machinery (ACM). https://doi.org/10.1145/1541948.1541999

Fonseca, M. A., & Grimshaw, S. B. (2017). Do behavioral nudges in prepopulated tax forms affect compliance? Experimental evidence with real taxpayers. *Journal of Public Policy & Marketing*, *36*(2), 213–226. https://doi.org/10.1509/jppm.15.128

Gajewski, J. F., Heimann, M., & Meunier, L. (2022). Nudges in SRI: the power of the default option. *Journal of Business Ethics* 177:547–566. https://doi.org/10.1007/s10551-020-04731-x

Gal, U., Lyytinen, K., & Yoo, Y. (2008). The dynamics of IT boundary objects, information infrastructures, and organisational identities: The introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems*, *17*(3), 290–304. https://doi.org/10.1057/ejis.2008.13

Galpin, T. (2022). Nudging innovation across the firm – Aligning culture with strategy. *Journal of Business Strategy*, *34*(1), 44–55. https://doi.org/10.1108/JBS-07-2020-0147

Green, R., & Ledgard, H. (2011). Coding guidelines: finding the art in the science. *Communications of the ACM*, *54* (12), 57–63. https://doi.org/10.1145/2043174.2043191

Gupta, P., Kim, Y. J., Glikson, E., & Woolley, A. W. (2019). *Digitally nudging team processes to enhance collective intelligence. Collective Intelligence Conference.*

Haki, K., Beese, J., Aier, S., & Winter, R. (2020). The evolution of information systems architecture: an agent-based simulation model. *MIS Quarterly*, *44*(1), 155–184. doi:10.25300/MISQ/2020/14494

Haki, K., & Legner, C. (2021). The mechanics of enterprise architecture principles. *Journal of the Association for Information Systems*, *22*(5), 1334–1375. https://doi.org/10.17705/1jais.00696

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. doi:10.2307/25148625.

Holvitie, J., Licorish, S. A., Spínola, R. O., Hyrynsalmi, S., MacDonell, S. G., Mendes, T. S., Buchan, J., & Leppänen, V. (2018). Technical debt and agile software development practices and processes: An industry practitioner survey. *Information and Software Technology*, *96*, 141–160. https://doi.org/10.1016/j.infsof.2017.11.015

Huber, T. L., Winkler, M. A., Dibbern, J., & Brown, C. V. (2020). The use of prototypes to bridge knowledge boundaries in agile software development. *Information Systems Journal*, *30*(2), 270–294. https://doi.org/10.1111/isj.12261

Jacoby, J. (1984). Perspectives on information overload. *Journal of Consumer Research*, *10*(4), 432–435. https://doi.org/10.1086/208981

Johnson, E. J., Shu, S. B., Dellaert, B. G., Fox, C., Goldstein, D. G., Häubl, G., Larrick, R. P., Payne, J. W., Peters, E., Schkade, D., Wansink, B., & Weber, E. U. (2012). Beyond nudges: Tools of a choice architecture. *Marketing Letters*, *23*(2), 487–504. https://doi.org/10.1007/s11002-012-9186-1

Kahneman, D., Knetsch, J. L., & Thaler, R. H. (1991). Anomalies: The endowment effect, loss aversion, and status quo bias. *Journal of Economic Perspectives*, *5*(1), 193–206. https://doi.org/10.1257/jep.5.1.193

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *The American Economic Review*, *93*(5), 1449–1475. https://doi.org/10.1257/000282803322655392

Kasunic, M. (2005). *Designing an effective survey*. Software Engineering Institute, Carnegie Mellon University. https://resources.sei.cmu.edu/asset_files/Handbook/2005_002_001_14435.pdf

Klinger, T., Tarr, P., Wagstrom, P., & Williams, C., (2011). *An enterprise perspective on technical debt. Proceedings of the 2Nd Workshop on Managing Technical Debt, MTD '11. ACM*, New York, NY, USA. Association for Computing Machinery (ACM), 35–38. https://doi.org/10.1145/1985362.1985371

Kruchten, P., Nord, R., & Ozkaya, I. (2019). *Managing technical debt: reducing friction in software development*. Addison-Wesley Professional.

Levin, I. P., Schneider, S. L., & Gaeth, G. J. (1998). All frames are not created equal: A typology and critical analysis of framing effects. *Organizational Behavior and Human Decision Processes*, *76*(2), 149–188. https://doi.org/10.1006/obhd.1998.2804

Levina, N. (2005). Collaborating on multiparty information systems development projects: A collective reflection-in-action view. *Information Systems Research*, *16*(2), 109–130. https://doi.org/10.1287/isre.1050.0055

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. https://doi.org/10.1016/j.jss.2014.12.027

Lim, E., Taksande, N., & Seaman, C. (2012). A balancing act: what software practitioners have to say about technical debt. *IEEE Software*, *29*(6), 22–27. https://doi.org/10.1109/MS.2012.130

Mirsch, T., Lehrer, C., & Jung, R. (2017). *Digital nudging: Altering user behavior in digital environments. Proceedings of the 13th International Conference on Wirtschaftsinformatik (WI 2017)*, 634–648.

Ölander, F., & Thøgersen, J. (2014). Informing versus nudging in environmental policy. *Journal of Consumer Policy*, *37*(3), 341–356. https://doi.org/10.1007/s10603-014-9256-2

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pennycook, G., McPhetres, J., Zhang, Y., Lu, J. G., & Rand, D. G. (2020). Fighting COVID-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological Science*, *31*(7), 770–780. https://doi.org/10.1177/0956797620939054

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

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. https://doi.org/10.1016/j.infsof.2018.05.010

Rolland, K. H., Mathiassen, L., & Rai, A. (2018). Managing digital platforms in user organizations: the interactions between digital options and digital debt. *Information Systems Research*, *29*(2), 419–443. https://doi.org/10.1287/isre.2018.0788

Schilling, R., Aier, S., & Winter, R. (2019). *Designing an artifact for informal control in enterprise architecture management. Proceedings of the 40th International Conference on Information Systems (ICIS 2019)*, Munich, Germany. Association for Information Systems.

Schneider, C., Weinmann, M., & Vom Brocke, J. (2018). Digital nudging: Guiding online user choices through interface design. *Communications of the ACM*, *61*(7), 67–73. https://doi.org/10.1145/3213765

Schwaber, K., & Sutherland, J. (2020). *The scrum guide - the definitive guide to scrum: the rules of the game*. Scrum.org https://www.scrum.org/resources/scrum-guide

Seaman, C., Guo, Y., Zazworka, N., Shull, F., Izurieta, C., Cai, Y., & Vetrò, A. (2012). *Using technical debt data in decision making: potential decision approaches. Third International Workshop on Managing Technical Debt (MTD 2012)*, Zurich, Switzerland. IEEE.

Siavvas, M., Tsoukalas, D., Jankovic, M., Kehagias, D., & Tzovaras, D. (2022). Technical debt as an indicator of software security risk: A machine learning approach for software development enterprises. *Enterprise Information Systems* 16(5): 849–891. https://doi.org/10.1080/17517575.2020.1824017

Sonnenberg, C., & Vom Brocke, J. (2011). Evaluation patterns for design science research artefacts. *European Design Science Symposium*, Leixlip, Ireland. Springer-Verlag Berlin Heidelberg.

Sunstein, C. R. (2014). Nudging: A very short guide. *Journal of Consumer Policy*, *37*(4), 583–588. https://doi.org/10.1007/s10603-014-9273-1

Thaler, R. H., & Benartzi, S. (2004). Save more tomorrow^{TM}: using behavioral economics to increase employee saving. *Journal of Political Economy*, *112*(S1), 164–187. https://doi.org/10.1086/380085

Thaler, R. H., & Sunstein, C. R. (2009). *Nudge: improving decisions about health, wealth, and happiness*. Penguin.

Thomas, D., Seenivasan, S., & Wang, D. (2021). A nudge toward healthier food choices: The influence of health star ratings on consumers' choices of packaged foods. *European Journal of Marketing*, *55*(10), 2735–2768. https://doi.org/10.1108/EJM-11-2019-0851

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

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). Focus groups for artifact refinement and evaluation in design research. *Communications of the Association for Information Systems*, *26*, 599–618. https://doi.org/10.17705/1CAIS.02627

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

Tversky, A., & Kahneman, D. (1981). The framing of decisions and the psychology of choice. *Science*, *211* (4481), 453–458. https://doi.org/10.1126/science.7455683

Weinmann, M., Schneider, C., & Vom Brocke, J. (2016). Digital nudging. *Business & Information Systems Engineering*, *58*(6), 433–436. https://doi.org/10.1007/s12599-016-0453-1

Woodard, C. J., Ramasubbu, N., Tschang, F. T., & Sambamurthy, V. (2013). Design capital and design moves: the logic of digital business strategy. *MIS Quarterly*, *37*(2), 537–564. https://doi.org/10.25300/MISQ/2013/37.2.10

Wu, S. J., & Levy Paluck, E. (2021). Designing nudges for the context: golden coin decals nudge workplace behavior in China. *Organizational Behavior and Human Decision Processes*, *163*, 43–50. https://doi.org/10.1016/j.obhdp.2018.10.002

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. https://doi.org/10.1016/j.jss.2016.05.018

Yoon, S., Naderpajouh, N., & Hastak, M. (2019). Decision model to integrate community preferences and nudges into the selection of alternatives in infrastructure development. *Journal of Cleaner Production*, *228*, 1413–1424. https://doi.org/10.1016/j.jclepro.2019.04.243

Appendix

Appendix A. Research approach

Our project for clinical design and evaluation of a TDM nudge at Credit Suisse adopted a design science research (DSR) approach, as Peffers et al. (2007) proposed. The DSR process that guides our project consists of six steps: (1) problem identification and motivation, (2) definition of the objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication. The project team started by elaborating on the problem space (step 1). With several years of experience in TDM, Credit Suisse already had several formal procedures related to TDM. However, all of these procedures required the kind of top-down enforcement that development teams may not appreciate or follow when developing software under tight schedules. Therefore, we lacked complementary mechanisms that address TD at its roots – that is, in individuals' decisions – and that development teams do not consider as yet another constraint or formal procedure. In step 2, we defined the solution's objective as to develop and evaluate a TDM nudge at Credit Suisse that supports software development teams at the individual decision level to address TD at its roots in a nonbinding fashion. Based on this objective, we inferred the necessary actions for designing and developing the nudge in step 3. Based on the current TD practices and premises for nudging, the TDM nudge included design elements that combined the underlying psychological effects of nudging with measures from Credit Suisse for the type of TD. Then, in step 4, we demonstrated the developed nudge to IS experts and agile software development teams at Credit Suisse.

The evaluation step, step 5, consisted of two main phases. In the first phase (Appendix B), we assessed the TDM nudge's usefulness for the target audience using three activities. First, we evaluated the nudge qualitatively and collected feedback on the artefact's initial version through nine semi-structured interviews with senior architects and senior IT managers at Credit Suisse. Second, to evaluate how well the target users (i.e., agile software development teams) understood the nudge, we conducted three focus group workshops, each with eight to nine participants. Third, we assessed the artefact's applicability and usefulness through pre- and post-implementation surveys of the artefact's target users at Credit Suisse. The data was obtained twice from the same audience (within-subject comparison): The pre-implementation survey was distributed before we demonstrated the TDM nudge to software development teams. Then, a week later, the second post-implementation survey was distributed to the same participants. Both questionnaires measured the constructs of TD awareness, TD understanding, and TDM-related action (Schilling et al., 2019), which we derived from TDM literature. The questionnaire included statements about identifying, measuring, prioritising, monitoring, repaying, representing, and communicating TD (Li et al., 2015). Data from 16 respondents (67% response rate) was collected. In the second phase of the evaluation (Appendix C), we assessed the TDM nudge's effects on individual IT applications' TD levels by tracking the quantitative changes in TD levels for the IT applications after one year of the TDM nudge's deployment and use. To gain insight into the development of and the reasons behind the changes in TD levels, we distributed another survey among software development teams and conducted follow-up interviews with senior management. Both evaluation phases used team-level evaluations of TD because the TDM nudge targeted TD decisions that are made at the level of the software development teams and because team-level analysis allowed us to circumvent several uncontrollable, enterprise-level variables (e.g., strategic turnarounds, changes in personnel, adaptations in TD monitoring and measurement tools). In step 6, the final step, we communicate our project through the manuscript at hand.

Appendix B. Evaluation Phase 1: Usefulness

B1. Semi-structured Expert Interviews

Following Brinkmann's (2013) and Sonnenberg and Vom Brocke (2011) guidelines, we presented a brief introduction to the project and then asked experts to evaluate the design of the nudge based on three criteria (Table A1):

- Understandability: the degree to which they understand the nudge
- Simplicity: the degree to which the nudge contained the minimum number of elements

Table A1. Interview guideline.

Topic	Question
Understandability	1. Are the digital nudge and its elements easy to understand?
	2. Which elements of the nudge are particularly difficult to understand?
Simplicity	3. Is the nudge simple enough (without limiting its value contribution)?
Effectiveness	4. Do you think the nudge can support Technical Debt Management-related decisions?
	5. Do you find the nudge useful for communicating Technical Debt towards relevant stakeholders?
Improvement	6. Do you have any suggestions for improving the nudge in terms of information or design without limiting its value contribution?

Table A2. Overview of expert interviews.

Interview Session	Part. ID	Role/Background	Date/ Time	Location	Duration
1	P1	Global Chief Architect	24.02.20, 16:30	Zurich	30 min
2	P2	Chief Architect of Business Department A	25.02.20, 13:30	Skype	52 min
3	P3	Head of Application Support	03.03.20, 14:00	Skype	33 min
4	P4	Chief Architect of Department B	10.03.20, 15:30	Skype	37 min
	P5	Architect of Department B		Skype	
5	P6	Performance Manager	12.03.20, 14:00	Skype	32 min
6	P7	Head of Development Practices	12.03.20, 15:00	Skype	14 min
7	P8	IT Platform Architect	23.03.20, 14:00	Skype	28 min
8	P9	CIO of Business Department C	07.04.20, 10:00	Skype	30 min
	P10	IT COO of Business Department C		Skype	
	P11	Chief Architect of Business Department C		Skype	
9	P12	Head of Security Services	15.04.20, 11:30	Skype	20 min

- Effectiveness: the degree to which the nudge can support TDM-related decision-making

Table A3. Overview of focus group workshops.

Workshop Session	Part. ID	Role/Background	Date/ Time	Location	Duration
1	S1	Developer/Architect	03.03.20, 14:00	Zurich	60min
	S2	Developer/Architect			
	S3	Developer/Architect			
	S4	Developer/Architect			
	S5	Developer/Architect			
	S6	Developer/Architect			
	S7	IT Owner/Product Owner			
	S8	Developer/Architect			
2	S9	Developer/Architect	30.03.20, 13:00	Skype	60min
	S10	Developer/Architect			
	S11	IT Owner/Product Owner			
	S12	Developer/Architect			
	S13	IT Owner/Product Owner			
	S14	Developer/Architect			
	S15	Developer/Architect			
	S16	Developer/Architect			
3	S17	IT Owner/Product Owner	16.04.20, 16:00	Skype	60min
	S18	Developer/Architect			
	S19	Developer/Architect			
	S20	Developer/Architect			
	S21	Developer/Architect			
	S22	Developer/Architect			
	S23	Developer/Architect			
	S24	Developer/Architect			

Table A4. Pre-implementation questionnaire.

Construct		Measurement Item
Awareness	Awa1	Our team perceives the existence of Technical Debt in our application.
	Awa2	Our team is aware of the amount of overall Technical Debt in our application.
	Awa3	Our team is aware of the level (e.g., low, high) of the overall Technical Debt in our application compared to other similar applications.
	Awa4	Our team is aware of the evolution of the Technical Debt over time in our application.
	Awa5	Our team is aware of the different types of Technical Debt.
	Awa6	Our team is aware of other teams that successfully manage Technical Debt.
Understanding	Und1	Our team has a common understanding of Technical Debt.
	Und2	Our team understands how to consider Technical Debt in software design and development.
	Und3	Our team understands the consequences of incurring Technical Debt.
	Und4	Our team understands how to prioritise remediation of Technical Debt items.
Action	Act1	Our team actively considers Technical Debt in software design and development.
	Act2	Our team discusses Technical Debt topics, e.g., during daily stand-up, sprint planning or sprint retrospective.
	Act3	Our team tracks different types of Technical Debt.
	Act4	Our team monitors the evolution of Technical Debt in our application over time.
	Act5	Our team actively repaid Technical Debt in our application in the past, e.g., through refactoring or reengineering.
	Act6	Our team uses visualisation tools to communicate Technical Debt to relevant stakeholders.
	Act7	Our team actively prioritises remediation of Technical Debt items or asks the Product Owner to prioritise remediation of Technical Debt items.

Suggestions for improving the nudge were also collected (Table A1), for which we analysed our notes from the interviews. We conducted nine interviews (Table A2) from February 2020 to April 2020 with senior architects and senior IT managers of Credit Suisse who had extensive knowledge of TD. The interviews took place right after each demonstration session.

B2. Focus Group Workshops

We demonstrated the TDM nudge to the target users – three agile software development teams at Credit Suisse with eight or nine participants each – during focus group workshops between March and April 2020 (Table A3). Each software development team was presented with the TDM nudge and TD data for its IT application(s). We also used the workshops to evaluate how well the software development teams understood the nudge. Following the guidelines on using focus groups in DSR that Tremblay et al. (2010) proposed, the workshops began with an explanation of the motivation behind the design of the TDM nudge by one co-author who took the role of a moderator. Then the workshops proceeded with a description of how the artefact could be used, along with details of the design, and concluded with the participants' evaluating the nudge using the interview guideline (shown in Table A1). Notes taken during the focus group workshops were subsequently analysed.

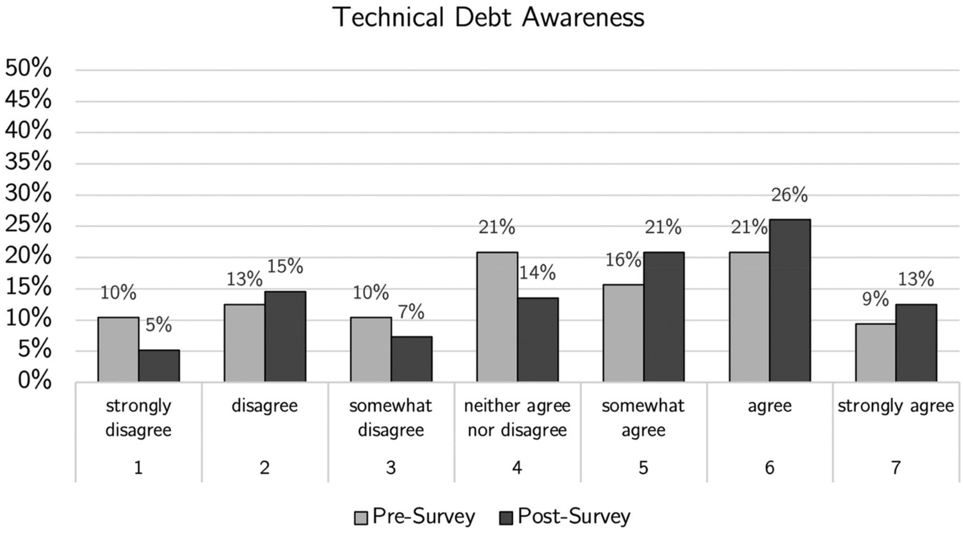


Figure B2. Frequency of answer choices (in %) on the TD awareness construct.

B3. Pre- and Post-Implementation Survey

We followed Kasunic (2005) in designing the questionnaire. The questionnaire was tested with six developers at Credit Suisse, after which we made adjustments to phrasing to ensure clarity (Table A4) and distributed it to participants via email.

The first part of the pre-implementation questionnaire solicited demographic information about the participants' experience in software development, their project roles, the team's size, and whether the team is geographically distributed across various locations. The second part of the questionnaire measured the constructs of TD awareness, TD understanding, and TDM action (Schilling et al., 2019), according to which the statements in the questionnaire were categorised. The measurement items were derived from the TDM literature and included statements about the TDM-related activities (Li et al., 2015) of identification, measurement, prioritisation, monitoring, repayment, representation, and communication of TD. The statements included answer choices on a seven-point Likert scale (1 = strongly disagree to 7 = strongly agree).

Only in the pre-implementation questionnaire did participants provide demographic information, as the same participants filled out the post-implementation questionnaire. After the nudge demonstration, the post-implementation questionnaire (Table A5) was distributed to the same participants, and data from 16 respondents (67% response rate) was collected and analysed (Figure B2, B3 and B4).

Appendix C. Evaluation Phase 2: Effects on Technical Debt Levels

One year after the TDM nudge's deployment and use, we quantitatively tracked the development of TD in the three software development teams' IT applications. Then we conducted a follow-up survey and interviews to substantiate the quantitative tracking results and determine why we obtained these results. At Credit Suisse and in all surveys and interviews, the TDM nudge is referred to as a "TDM label" to avoid revealing more information than necessary about the intervention's intent and its underlying mechanisms. At this stage, while the follow-up survey targeted members of the

Table A5. Post-implementation questionnaire.

Construct		Measurement Item
Awareness	Awa1	Our team perceives the existence of Technical Debt in our application.
	Awa2	Our team is aware of the amount of overall Technical Debt in our application.
	Awa3	Our team is aware of the level (e.g., low, high) of the overall Technical Debt in our application compared to other similar applications.
	Awa4	Our team is aware of the evolution of the Technical Debt over time in our application.
	Awa5	Our team is aware of the different types of Technical Debt.
	Awa6	Our team is aware of other teams that successfully manage Technical Debt.
Understanding	Und1	Our team has a common understanding of Technical Debt.
	Und2	Our team understands how to consider Technical Debt in software design and development.
	Und3	Our team understands the consequences of incurring Technical Debt.
	Und4	Our team understands how to prioritise remediation of Technical Debt items.
Action	Act1	I would like my team to actively consider Technical Debt in software design and development.
	Act2	I would like my team to discuss Technical Debt topics, e.g., during daily stand-up, sprint planning or sprint retrospective.
	Act3	I would like my team to track different types of Technical Debt.
	Act4	I would like my team to monitor the evolution of Technical Debt in our application over time.
	Act5	I would like my team to actively repay Technical Debt in our application, e.g., by refactoring or reengineering.
	Act6	I would like my team to use visualisation tools to communicate Technical Debt towards relevant stakeholders.
	Act7	I would like my team to actively prioritise remediation of Technical Debt items or ask the Product Owner to prioritise remediation of Technical Debt items.

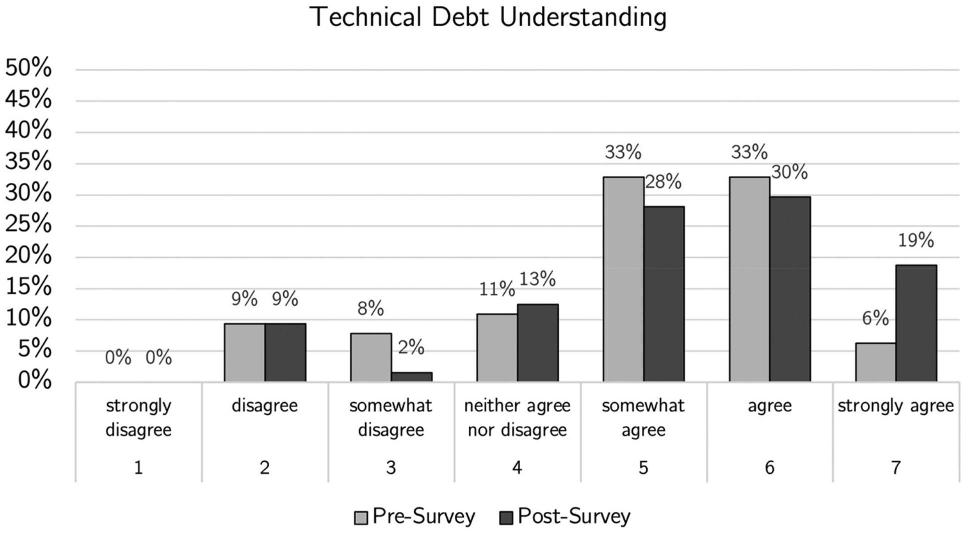


Figure B3. Frequency of answer choices (in %) on the TD understanding construct.

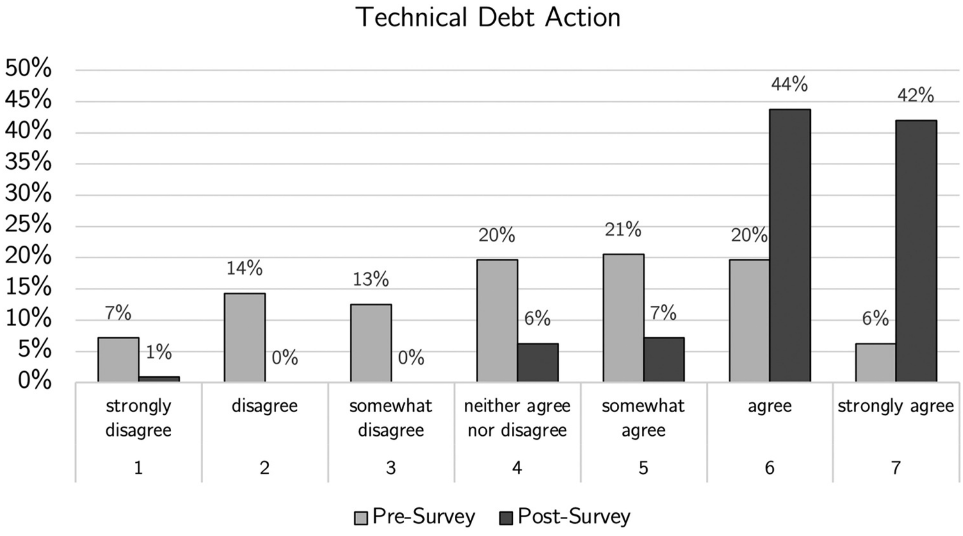


Figure B4. Frequency of answer choices (in %) on the TDM action construct.

software development teams, we held semi-structured interviews with IT managers to gain insights from both perspectives.

Both modes of inquiry – the survey and the interviews – served the goal of explaining the use patterns of the TDM nudge over the course of the year and the perceptions of its impact on decision-making. Mainly containing open-ended questions, the survey and the interviews revolved around three themes. The first theme was that of *use patterns*, where we determined such factors as frequencies of the TDM nudge's use, situations and occasions in which the TDM nudge was used, and practices related to using the TDM nudge, such as checking the team's TD levels and comparing them with other teams' TD levels. The second theme was that of *communication*, where we determine, for example, how the TDM nudge impacted communication with internal and external stakeholders and when and how the TDM nudge was referenced to discuss TD-related issues. The final theme, that of *action*, concerned such factors as whether the TDM nudge impacted software development teams' daily decisions, whether the nudge triggered managers' strategic TD-related actions, and whether the nudge would continue to be used in the future.

Appendix D. Implementation of the Technical Debt Management Nudge

The overall rating (DE01) threshold was initially calibrated based on all IT applications' distribution in the TD index, as provided in Table A6. For example, the threshold of the A rating was initially calibrated as the highest-rated 15 percent of IT applications at Credit Suisse. The TD index thresholds for the ratings stayed stable after the initial calibration.

Table A7 describes the variables used to evaluate each type of TD.

To calculate the indexes for each type of TD and the overall TD index for each IT application, Credit Suisse developed calculation formulas as depicted in Table A8 using the variables described in Table A7. The formulas use a logarithmic scale to make nuances at the lower end of the scale visible; therefore, base values were selected so a reasonable start for the scale could be defined. Automation debt is calculated based on a step function. The calculation considers build and deployment execution for test and production environments, whereby use of an automated, code-based build and deployment pipeline defines the no-TD state.

Table A6. Overall TD rating calculation scheme.

Rating Colour	🟩	🟩	🟨	🟧	🟥
Rating	A	B	C	D	E
Distribution	<15%	<35%	<65%	<85%	>85%
TD Index Threshold	<2.3	<3.7	<4.6	<5.9	>5.9

Table A7. Variables calculated for the TDM nudge.

Type of TD	Variable	Description
Code Quality Debt	*lines_of_code*	Number of lines of code in the application's source code
	major_violations	Number of major violations identified in the source code
	critical_violations	Number of critical violations identified in the source code
	blocker_violations	Number of blocker violations identified in the source code
	total_violations	Total number of violations identified in the source code
	code_debt_rating	Code quality debt rating
Infrastructure Lifecycle Debt	*eosl_comp*	Number of components that are End-of-Service-Life
	eosl_comp_months_cum	End-of-Service-Life months accumulated
	infr_debt_ratg	Infrastructure lifecycle debt rating
Vulnerability Debt	*low_vuln*	Number of identified vulnerabilities rated as low
	med_vuln	Number of identified vulnerabilities rated as medium
	high_vuln	Number of identified vulnerabilities rated as high
	total_vuln	Number of total vulnerabilities identified
	vuln_debt_rating	Vulnerability debt rating
Automation Debt	*automation_debt_rating*	Automation debt rating
Overall Technical Debt	*tech_debt_rating*	Overall Technical Debt Index

Table A8. Calculation formula for TD type indexes.

Type of TD	Calculation Formula
Code Quality Debt	$Rating_{Code\_Quality} = 4.5 + \log_2\left(\frac{100 \cdot total_{violations}}{lines_{of_{code}}}\right)$
Infrastructure Lifecycle Debt	$Rating_{Component\_Lifecycle} = \log_2\left(\frac{eosl_{comp_{months_{cum}}}}{4}\right)$
Vulnerability Debt	$Rating_{Vulnerability} = 4.6 + \log_2\left(\frac{low_{vuln}}{25} + \frac{med_{vuln}}{5} + high_{vuln}\right)$