# Quantifying TD Interest: Are we Getting Closer, or Not Even That?

Elvira-Maria Arvanitou, Pigi Argyriadou, Georgia Koutsou, Apostolos Ampatzoglou, Alexander Chatzigeorgiou

Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
Department of Computer Science, Open Hellenic University, Patras, Greece
e.arvanitou@uom.edu.com, a.ampatzoglou@uom.edu.gr, achat@uom.gr

***Abstract*—Despite the attention that Technical Debt has attracted over the last years, the quantification of TD Interest still remains rather vague (and abstract). TD Interest quantification is hindered by various factors that introduce a lot of uncertainty, such as: identifying the parts of the system that will be maintained, quantifying the load of maintenance, as well as the size of the maintenance penalty, due to the existence of TD. In this study, we aim to shed light on the current approaches for quantifying TD Interest by exploring existing literature within the TD and Maintenance communities. To achieve this goal, we performed a systematic mapping study on Scopus and explored: (a) the existing approaches for quantifying TD Interest; (b) the existing approaches for estimating Maintenance Cost; and (c) the factors that must be taken into account for their quantification. The broad search process has returned more than 1,000 articles, out of which only 25 provide well-defined mathematical formulas / equations for the quantification of TD Interest or Maintenance Cost (only 6 of them are explicitly for TD Interest). The results suggest that despite their similarities, the quantification of TD Interest presents additional challenges compared to Maintenance Cost Estimation, constituting (at least for the time being) the accurate quantification of TD Interest an open and distant to solve research problem. Regarding the factors that need to be considered for such an endeavor, based on the literature: size, complexity, and business parameters are those that are more actively associated to TD Interest quantification.**

***Keywords*—*interest; maintenance; technical debt***

## I. Introduction

The Technical Debt (TD) metaphor this year celebrates 30 years from its introduction by Ward Cunningham [1]; whereas the TD research community is already active for more than a decade[1], having produced a vast number of research articles. However, the quantification of one of the two pillars of the metaphor (namely: TD Interest) still remains vague and abstract, in the sense that there is still no state-of-research and -practice approach for quantifying it [2]. Consequently, any study that attempts to quantitatively explore TD concepts, or propose technical debt management approaches, faces severe construct validity problems. This shortcoming also influences the business aspects of TD management, in the sense that any informed decision to repay or not TD, must be made based on numerical evidence on both the benefits of technical debt (saved effort or business-related gains), but also on the cost of technical debt (mostly associated with future Interest payments), being measured at the same unit. Finally, the lack of TD Interest quantification approaches, hinders the application of financial debt theories and management approaches, which heavily rely on the concepts of interest and interest rate to efficiently manage a debt, from an economics point of view.

Despite the fact that regarding TD Principal the situation is not (by far) better in terms of accuracy [3] and agreement among tools [4][5] is relatively low; a researcher that focuses on quantifying TD Principal has to choose among more than 20 tools (ranging from research prototypes to commercial tools) [2]; whereas the construction of benchmark datasets on the intersection of some tools have started to appear [4]. On the other hand, the quantification of TD Interest lies at such an infant level, that only very limited number of quantitative empirical studies have been performed using TD Interest, and the number of available tools is negligible [2]. TD interest is mostly discussed in the literature as a theoretical concept. At this point, we need to note that we differentiate quantification from assessment, in the sense that assessment includes proxies (such as maintainability), whereas quantification aims at calculating an amount of technical debt interest in a currency / monetary unit (as dictated by the metaphor itself), or at minimum using an effort unit, which can easily be transformed to currency, being aware of the payment units in a company.

The main reason for the lag of TD Interest quantification lies in the complexity and inherent uncertainty involved in the definition of TD Interest, which does not yet allow the quantification without any assumptions. According to a TD glossary, introduced through a former literature review [6], TD Interest is defined as "*The additional development effort required to modify the software (adding new features or fixing bugs), due to the presence of TD issues*". Based on this definition, the calculation involves many uncertain parameters: (a) an accurate calculation would require comparing the current version of a system with a zero-TD version of the same system, with respect to their difference in required effort for maintenance: of course, such a debt-free version does not exist and would be unrealistic to create in a real-world setting. To some extent it is not easy to define or even suggest how a debt-free system looks-like; and (b) an accurate calculation would require an a-priori knowledge of the future maintenance requests (features or bug fixes), as well as the maintenance load that they will impose. Despite the certainty that software will change, usually in practice we have "*no clue*" on how a software will evolve, since evolution drivers are numerous (customers, application domains, bugs, etc.).

As a first step to understand the current state-of-the-art on how the aforementioned challenges are approached by the TD community, in this paper we provide an overview of the literature on TD Interest quantification. Taking into account the lack of a comprehensive approach that solves the aforementioned problems, we expand our overview to models that aim at quantifying Maintenance Cost. To achieve this goal, we per-

[1] The 1st Managing Technical Debt (MTD) workshop was held in June 2010 at Pittsburgh, Pennsylvania.

formed a systematic mapping study on a broad-scoped indexing mechanism (namely: Scopus), attempting to answer three main research questions:

- What approaches exist for quantifying TD Interest?
- What approaches exist for maintenance cost estimation?
- What factors should be used for building a TD Interest quantification model?

The rest of the paper is organized as follows: in Section II we present related work, i.e., secondary studies on TD or state-of-practice studies that explore TD quantification. Next, in Section III we present our review protocol; the raw results are presented in Section IV (including a detailed presentation of existing TD Interest formulas). The results are discussed in Section V; the threats to validity are presented in Section VI; and finally, the paper is concluded in Section VII.

## II. Related Work

Tom et al. [7] performed a multivocal literature review, supplemented by interviews with software practitioners and academics, in order to establish the boundaries of the TD phenomenon. The search process is performed in Google's Web Search engine. After applying the selection criteria, 35 studies were identified. The results of their study suggested a theoretical framework that provides a holistic view of TD, comprising a set of TD dimensions, attributes, precedents, and outcomes. Li et al. [8] conducted a mapping study on TD management (TDM). More specifically, the study focused on the identification of a classification of TD concepts, and the current state of research on TDM. The search process was conducted between 1992 and 2013 in seven digital libraries (namely Scopus, ACM DL, Wiley, IEEE DL, INSPEC, Web of Science and SpringerLink), identifying 94 primary studies. The results of the study revealed 8 different TDM activities, 29 tools for TD management, and 10 types for the classification of technical debt concepts.

Ampatzoglou et al. [6] performed a literature review on the financial aspects of managing TD. More specifically, the goal of this paper focused on financial aspects underlying software engineering concepts. The search strategy identified papers until 2015 and was conducted on seven digital libraries (namely ACM, IEEE, ScienceDirect, SpringerLink, Scopus, Web of Science, and Google Scholar). At the end of the selection process, 69 primary studies were selected. The results of the study provide a glossary of terms and a classification scheme for financial approaches to be applied for managing technical debt. As a follow-up study on this work, the same authors re-explored the dataset, from a TD Interest perspective [9]. In this study the authors proposed a framework namely FITTED for managing TD interest, which takes into account technical debt literature and economic interest theories. The FITTED framework discusses in detail: (a) the types of TD Interest; (b) various characteristics on interest in economics that are applicable for the TD metaphor; and (c) present a novel TD Interest theory.

Alves et al. [10] conducted a mapping study on the identification of strategies that have been proposed to identify and manage TD in software projects. The search process was performed between 2010 and 2014 on eight digital libraries (namely ACM, IEEE, ScienceDirect, Engineering Village, SpringerLink, Scopus, CITESEER, and DBLP), and retrieved 100 primary studies. The results of the study suggest an initial taxonomy of TD types and a list of indicators to identify TD and management strategies. Fernández-Sánchez et al. [11] performed a mapping study on the identification of the elements needed to manage technical debt. The search strategy was conducted until 2017 and retrieved 69 primary studies. The results of the study suggest that the elements can be classified into three groups (basic decision-making factors, cost estimation techniques, practices and techniques for decision-making) and grouped based on stakeholders' points of view (engineering, engineering management, and business-organizational management). Behutiye et al. [12] conducted a literature review analyzing the causes, consequences, and management strategies for TD, in the context of agile software development. The search process identified until 2017 on six digital libraries (namely ACM DL, Google Scholar, IEEE DL, ProQuest, Scopus, and Web of Science) and selected 38 primaries studies. The results of the study suggest five research areas of interest related to agile software development and technical debt, highlighting "*managing TD for Agile Development*" as the most studied research area.

Besker et al. [13] performed a literature review on the management of architectural technical debt. The search strategy was conducted between 2005 and 2016 on 6 six digital libraries (namely ACM DL, IEEE DL, ScienceDirect, SpringerLink, Scopus, and Web of Science) retrieved 43 primary studies. The results showed a lack of guidelines on how to manage Architectural TD successfully in practice and of an overall process where these activities are fully integrated. Lenarduzzi et al. [14] conducted a literature review on technical debt items' prioritization. The search process identified papers until 2020, in which 44 primary studies were selected. The results of the study provide an impact map of the factors that the literature used to prioritize technical debt items.

Next, we present state-of-practice reports that focus on TD management tools, whose majority does not support TD Interest estimation. Avgeriou et al. [2] conducted a literature review on the identification of TD management tools. The authors compared existing tools for measuring TD, in terms of features, popularity, and empirical evidence. The search was performed on IEEE DL, ACM DL, Google, Bing, and Yahoo. The results suggest that all the TD management tools are applicable to quantify TD principal, but only few consider TD Interest. Saraiva et al. [15] performed a mapping study to identify and analyze available tools for managing TD. Most of the tools address technical debt related to code, design, and/or architecture artifacts. The search strategy identified papers on 6 digital libraries (namely ACM DL, IEEE DL, Science Direct, Research Gate and Scopus) and retrieved 47 primary studies. The results of the study suggest that from the 50 TD tools reported in the literature, 42 of them are new tools, and 8 tools extend an existing one. Pavlic and Hlis [16] provided a comparison between the available tools for managing technical debt. Based on their findings, the TD area is poorly supported by tools, for most TD activities. Finally, Fontana et al. [17] presented five tools that compute technical debt indexes. In particular, the authors provided the different TD Indices provided by the tools and described the differences focusing on the architecture-related issues. Additionally, the authors characterized which tools provide estimation either for TD principal either for TD interest.

Based on the above, it becomes evident that in the current secondary literature there is little evidence on how to quantify TD Interest, despite its importance for the success of the metaphor, as well as, the large number of secondary studies on the domain of TD Management.

## III. Review Protocol

This section describes the protocol of the performed systematic mapping study. A protocol is a pre-determined plan that describes research questions, and presents in detail how the mapping study was conducted, so as to be replicable. Our protocol has been developed, based on the guidelines of Petersen et al. [18].

### A. Research Goals and Questions

In this study, we plan to summarize existing methods, frameworks, approaches, or models that have been proposed for quantifying TD Interest or Maintenance Cost, as well as the factors that are considered during the quantification process. Driven by the aforementioned goal, we extracted two Research Questions (RQ), as follows:

**$RQ_1$**: *Which approaches exist for TD Interest quantification?*

**$RQ_2$**: *Which approaches exist for Maintenance cost estimation?*

Answering $RQ_1$ aims to identify all the existing formulas that directly quantify TD Interest. In $RQ_2$, we aim to find all the existing formulas that can prove useful for quantifying TD Interest, by estimating maintenance cost. Apart from noting and discussing the quantification formulas, we also categorize the parameters that the formula uses into HL categories. Additionally, we record the unit of measurement for each formula. The results on the two research questions are synthesized and compared, as part of discussion and not an additional research question.

### B. Search Procedure

The search procedure of a systematic mapping study aims to find as many primary studies, related to the research questions, as possible using an unbiased strategy. To achieve this goal, we have utilized a broad index of scientific literature, namely Scopus. In this study, we follow two different routes for primary study collection: (a) the direct TD Interest quantification route; and (b) the indirect TD Interest quantification route—through Maintenance Cost estimation. For the first route, we used the search term ("*technical debt*" OR "*TD*") in the title of the paper, and the term "*interest*" in the abstract of the paper. This process led to a set of primary studies—termed as ***PS1***. For the second route, we used the search term: ("*maintenance effort*" OR "*maintenance cost*" OR "*maintenance model*") in the title of the paper, since we were interested in studies that explicitly propose maintenance cost estimation approaches. This process led to a set of primary studies—termed as ***PS2***. As an extra step for enriching PS1, we have used forward and backward snowballing from all secondary studies in the field of TDM, as they are described in Section II. Both PS1 (after snowballing) and PS2 sets have been exported in a `bibtex` file, and were stored for further management and filtering in `JabRef`.

### C. Papers Screening

The papers that have been stored in PS1 and PS2 sets have been manually reviewed, so as to ensure that they are relevant to TD Interest quantification or Maintenance Cost estimation. In line with Dyba and Dingsoyr [19], we have performed filtering of the candidate article set at 3 stages. The search process, has returned a set of candidate primary studies ($1^{st}$ stage). Next, articles in PS1 and PS2 have undergone through a manual inspection of their titles and abstracts / conclusions ($2^{nd}$ stage), and their full text ($3^{rd}$ stage). During the $2^{nd}$ and the $3^{rd}$ stages, several inclusion / exclusion criteria were tested:

***Inclusion Criteria***

***IC1***: The study introduces or used a method / approach / model / framework for quantifying TD Interest.
***IC2***: The study introduces or used a method / approach / model / framework for Maintenance Cost estimation.
***IC3***: The output of the method / approach / model / framework is represented in a monetary or effort format.

***Exclusion Criteria***

***EC1***: The study is a previous version of a more complete paper about the same research
***EC2***: The study is not written in English
***EC3***: The study is an editorial, position paper, keynote, opinion, tutorial, poster or panel

As the final set of primary studies, we have retained articles that satisfied the following overall criterion:

| |
|---|
| ((IC1 \| IC2) ***&*** IC3) ***&*** (**NOT** (EC1 \| EC2 \| EC3)) |

At this stage we need to note that as an implication of IC2, we aim to exclude papers that belong to two domains with a vast literature volume: maintainability prediction (not cost, but ease of maintenance), as well as classic software cost estimation approaches, using COCOMO, ISBAG, etc., which is not related to maintenance costs, but development costs.

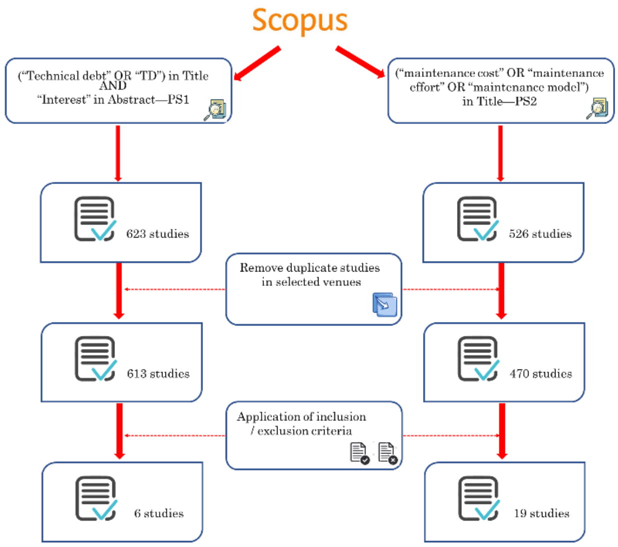


Fig. 1. Study Selection Process

Every article selection phase has been handled by the first four authors of this study using the voting approach as described by Farhoodi et al. [20]. The first four authors reviewed the publications and assigned a vote on a 4-point scale (4: strong inclusion, 1: strong exclusion)—leading to a maximum score of 12 points. Based on the threshold proposed by Farhoodi et al. [20], we retained studies with a score higher than 8 points. Studies that were marked with exactly 8 points (out of 12) were reviewed and discussed with the $5^{th}$ author of the study. As a first step, the authors discussed the process, so as to ensure that they hold a common understanding of the inclusion criteria. Next, the authors piloted the first 20 papers, which have been assessed in pairs by the four authors so as to have an open discussion on the voting scores. All authors explained their scores, until a consensus was reached. The high degree of a common understanding on the criteria is supported by the low disagreement rate in the inclusion / exclusion phase (i.e., 1.7%). We note that the exclusion criteria (language and type of paper) are straightforward and no validation or piloting was required. In Fig. 1 we present the overall process, accompanied by the number of articles characterizing each step.

### D. Keywording of Abstracts

As a next step for conducting systematic mapping studies, Petersen et al. [18] propose the keywording of abstracts, as a way to develop classification schemes. However, this step is not applicable in our study since it is not possible to extract neither the formulas nor the parameters that synthesize the formula from the abstract.

### E. Data Exatraction and Analysis

As a next step, we defined a set of variables that describe each primary study. Thus, for every study, we recorded the values of the following variables:

- [V1] **Publication Title**
- [V2] **Author**
- [V3] **Year**
- [V4] **Type of Paper** (Conference or Journal)
- [V5] **Publication Venue**
- [V6] **Formula** that quantifies TD Interest or Maintenance Cost
- [V7] **Parameters** that appear in the formula
- [V8] **Unit of Quantification**

The third and the fourth author independently extracted data. If there were inconsistencies in the extracted information, the involved authors discussed the inconsistencies between them. If they were not able to resolve the discrepancies, the first and the second author joined the discussion to resolve the disagreement. During the process 3 inconsistencies have been resolved. The dataset is available online[2].

Variables [V1] – [V5] have been used for documentation purposes. The rest of the variables have been used for answering the research questions and describing the context of the study. For reporting purposes, we used common visualization methods (i.e., bar charts), and frequency tables. For the consolidation of the values of the [V7] retrieved from the primary studies, we have performed the Open Card Sorting methodology [21], since the terminology used was quire diverse. As a first step we recorded themes from the parameters as identified in the primary studies, and then we reviewed them to find candidates for merging. As a final step, we defined the names of the final themes. The first two authors performed the process in order to identify the themes, and the fifth author validated the results. The level of disagreement was again quite low (approximately 4%) during the consolidation process of the extraction of themes.

## IV. Results

### A. TD Interest Quantification (RQ$_1$)

In this section we present the results of our mapping study related to TD Interest. More specifically, we have identified 6 studies that provide formulas in order to quantify the TD Interest [S1-S6]. The formulas, along with a brief explanation of their rationale is presented below.

Ampatzoglou et al. [S1] calculate TD Interest as the difference between the effort required to maintain the optimal system (e.g., *a hypothetical TD-free system*) and the actual system when performing a maintenance activity (e.g., the addition of a new feature). More specifically, Ampatzoglou et al. [S1] used the average number of lines of code maintained between sequential versions (k) as an estimate of future maintenance load. Additionally, they assumed that maintenance effort is proportional to the design quality, which is denoted as a fitness function value—for illustrating the fitness function they used a set of a well-known maintainability metrics; namely: DIT, NOCC, MPC, RFC, DAC, LCOM, CC, WMPC, SIZE1, and SIZE2. The ratio of optimum and actual fitness value is calculated as the average distance of the aforementioned metrics from a hypothetical (synthesized) optimal one (for more details see [S1]). The authors illustrate how their function can lead to currency assessments, by using values for the hourly rate of the developers, as well as the hourly maintenance productivity.

$$Effort = c * FitnessValue$$

$$Interest = \Delta Effort = Effort(actual) - Effort(optimum)$$

$$= Effort(actual) - Effort(actual) \times (MaintainabilityLevel)$$

$$= k * \left(\frac{FitnessValue(optimum)}{FitnessValue(actual)} - 1\right)$$

Falessi and Reichel [S2] approach a different approach for defining TD interest, since they consider interest as the cost of violating a rule or not fixing a violated rule (i.e., defect proneness), suggesting that such a cost is commonly measured or estimated as decreased productivity or extra defects [22]. Falessi and Reichel [S2] used the MIND tool to calculate TD Interest, which uses several metrics: Defect Proneness (DP), Maximum Defects per 100 LOC Touched (MaxDP), Extra Defect Proneness (EDP), Maximum Extra Defects per 100 LOC Touched (MaxEDP), Relative Extra Defect Proneness (REDP), Average Relative Extra Defect Proneness (AREDP), Violation Density (VD), and Linkage. The formulas for calculating these metrics are provided below—The EDP corresponds to the interest (measured as a percentage of extra defects). The key point in these calculations is the definition of $DP_{ideal}$, which corresponds to the number of defects that would be produced in an optimal *TD-free class*. This value is calculated, based on historical data of classes that do not have any violations (for more details see [S2]).

$$DP = \frac{Number\ of\ Defects}{LOC_{touched}}$$

$$MaxDP = 100 * MAX(DP)$$

$$EDP = DP_{actual} - DP_{ideal}$$

$$MaxEDP = 100 * MAX(EDP)$$

$$REDP = \frac{DP_{actual} - DP_{ideal}}{DP_{actual}}$$

$$AREDP = AVERAGE(REDP)$$

$$VD = \frac{Number\ of\ Violations}{LOC}$$

$$Linkage = \frac{NumberOfCommitsWithTickets}{NumberOfCommits}$$

Guo et al. [S3] approach TD Interest from the perspective of having additional cost, due to the delay of a task. In that sense

[2] https://users.uom.gr/~a.ampatzoglou/aux_material/SMS_TDInterest.xlsx

they calculate TD Interest as the difference of the effort required to perform a task (X), subtracted from the effort that would be need if there were no delays (P). As an extra parameter, they calculate interest probability as the chance of delay, as well as the need to perform the task after the delay. The formula for TD Interest Probability calculation is omitted, since it is beyond the scope of this study.

$$Interest\ Amount = X - P$$

Martini et al. [S4] estimate TD Interest in terms of extra-effort spent in development and maintenance, due to the existence of Architectural TD. In particular, the authors propose a technique to estimate the convenience of refactoring the ATD. The authors related the TD Interest of the ATD with the complexity (CC) and bug proneness (HAL_BUGS). In the calculation a higher weigh is assigned at files that are commonly changed (CHANGE_DIFF). For the calculation of the changes, they used the committed lines of code (LOC). Based on these metrics, the authors calculate a *Refactoring Index*, which prioritizes refactoring based on both the impact and the probability of a file of interest to produce TD Interest. The transformation of the base measurements (CC, HAL_BUGS) to POINTS is performed, based on metric thresholds (e.g., 15 for CC [23]) and a point system defined in the original article[S4]. As a next step, the method considers the refactoring index of the actual and the refactored system, and calculates the saved effort by taking into consideration business parameters, such as releases, Developers' Work Months (DVM), etc.

$$REFACTORING_{INDEX} = (CC_{POINTS} + HALSTEADBUGS_{POINTS}) * CHANGE\_DIFF$$

Nord et al. [S5] propose an approach that quantifies TD Interest. The approach calculates TD Interest, which is due to architectural rework. The approach computed the rework cost associated with each new architectural element in a specific number of releases. *Rework cost* is calculated as a product of: (a) the dependencies of the maintained architectural element to other elements of the system (D); (b) the cost of implementing the new architectural elements (C); and (c) the change propagation metric, introduced by MacCormack et al. [24] that captures the percentage of system elements that can be affected, on average, when a change is made to an element (Pc). The change propagation metric of a system is computed as the density of the visibility matrix that captures all the direct and indirect dependencies in the system architecture, or in other words, the transitive closure of the dependency relationship. The unit of measurement is the unit of the implementation cost.

$$Rework\ Cost = D * C * P_c$$

Nugroho et al. [S6] quantify TD Interest as the extra cost spent on maintenance due to technical quality issues. In particular the interest is the difference in maintenance effort between a particular quality level and the optimal level. *Maintenance Effort* (ME)—which is the measurement of TD Interest—is calculated (at various quality levels) as a function of Maintenance Fraction (MF), Rebuild Value (RV), and Quality Factor (QF). MF represents the amount of maintenance effort spent on a system in a yearly basis, measured as a percentage of lines of code that is estimated going to change (added, modified, or deleted) yearly due to maintenance. Essentially, MF can be measured based on historical maintenance data, which can be different from system to system. RV is an estimate of effort (in man-months) that needs to be spent to rebuild a system using a particular technology. This is determined as the product of system size and a technology factor. System Size (SS) represents the total size of a system measured in lines of code. Alternatively, SS can be measured using functional size (i.e., Function Points). Technology Factor (TF) represents language productivity factor, such as CBO, WMC, LCOM, and DIT. This factor provides a conversion from source code statement to effort (i.e., man-months per source statement) through 'backfiring'. Finally, QF is a factor that is used to account for the level of quality. It is assumed that the higher the level of quality, the smaller is the effort that needs to be spent on maintenance. This assumption is justified by previous research, which reveals that performing changes on systems with higher code quality is more efficient.

$$ME = \frac{MF * RV}{QF}$$

$$QF = 2^{(QualityLevel-3)/2)}$$

$$RV = SS * (1 + R)^{t} * \mathrm{TF}$$

Based on the aforementioned TD Interest quantification approaches, we can observe that 50% of the equations provide the *currency* as a unit of measurement, 33% the *time*, and 17% the *size*. Currency corresponds to a value measured in dollars or euros, representing the amount that need to be spent for TD Interest payments; Time corresponds to hours or months that are needed for the extra development or maintenance activities. Finally, Size corresponds to the extra effort in terms of lines of code that need to be maintained, due to the presence of TD inefficiencies.

In terms of factors that affect TD Interest, and can be considered as parameters for its quantification, we have in total identified 44 factors. Due to their diversity, we merged parameters in high-level (HL) categories—see Table I. For each HL category, we provide two values: (a) the number of distinct HL categories that appear in the PS1; and (b) the number of distinct parameters that have been merged under the HL category—sorted by (a). We note that the total number of the parameters that are used in the equations will be different with the sum of parameters that have merged into HL categories. For example, given the first row, we can observe that 14 distinct business parameters have been used for the quantification of TD Interest, and these parameters span in 4 out of 6 studies that provide a formula for TD Interest. We note that both views (i.e., columns) are useful: The first view denotes the importance of the HL categories, whereas the second view denotes the availability of parameters for each HL category. With respect to the first most important HL categories, as "***Business Factors***" the literature considers the *cost of work for the employees*, *months between releases*, and *number of releases*. Additionally, "***Size***" measures are mostly related to lines of code (LoC), appearing in three forms: (a) lines of code as an indicator of quality; (b) lines of code as indicators of maintenance load; and (c) lines of code as indicator of system growth. In addition, as a proxy of quality, the literature focuses on metrics assessing "***Coupling***", "***Cohesion***", "***Complexity***", and "***Inheritance***", which according to Riaz et al. [25] are the most common maintainability predictors. Finally, we note that in 3 studies "***TD Principal***" is considered as a factor that influences TD Interest; this decision complies with the financial view of the metaphor, suggesting that financial interest is calculated as the product of principal with interest rate.

TABLE I. FREQUENCY OF HL CATEGORIES FOR PS1

| HL Category | #Studies | #Metrics |
|---|---|---|
| Business Factor | 4 | 13 |
| Size | 4 | 6 |
| TD Principal | 3 | 4 |
| Coupling | 3 | 5 |
| Evolution Metric | 3 | 2 |
| Inheritance | 2 | 2 |
| Cohesion | 2 | 1 |
| Complexity | 2 | 1 |
| Correctness | 1 | 8 |
| Process Factor | 1 | 3 |

### B. *Maintenance Cost Estimation (RQ2)*

In this section we present the data extracted for maintenance cost estimation. We note that in this section we omit the formulas per se, since they cannot quantify TD Interest, but only be used as inspiration for future research endeavors[3]. In the literature, we have identified 19 studies that provide Maintenance Cost estimation, including in total 91 distinct parameters. Similarly to $RQ_1$, we present some basic demographics on the unit of measurement: 21% of the equations use *Currency*, 74% *Time*, and 21% *Size*. An interesting observation from this finding is that the TD community has a stronger emphasis on measuring interest in a *Currency* unit, whereas the Maintenance community on *Time*. This finding complies with the fact that the main selling point of the TD metaphor is the use of currency units to alert managerial stakeholders on possible financial losses.

Following the structure of reporting in $RQ_1$, in Table II, we present the HL categories of Maintenance Cost factors. The top three factors are: Size, Complexity, and Business factors. A comparison and the intuitive interpretation of the difference of TD Interest and Maintenance Cost factors are synthesized and discussed in Section V. Regarding specific metrics, we can observe that Lines of Code, and Number of Methods are the prominent "***Size***" metrics, Number of Internal (non-coupling) Method calls is the most used "***Complexity***" metric; whereas Cost Factors (such as salary) is the most used "***Business Factor***".

TABLE II. FREQUENCY OF HL CATEGORIES FOR PS2

| HL Category | #Studies | #Metrics |
|---|---|---|
| Size | 16 | 31 |
| Complexity | 10 | 23 |
| Business Factor | 8 | 15 |
| Evolution Metric | 7 | 13 |
| Process Factor | 7 | 8 |
| Correctness | 6 | 5 |
| Human Factor | 3 | 5 |
| Inheritance | 2 | 2 |
| Coupling | 2 | 1 |
| Polymorphism | 1 | 2 |
| Cohesion | 1 | 1 |
| RT Quality | 1 | 1 |
| Performance | 1 | 1 |

## V. DISCUSSION

In this section, we discuss the findings of our study, organized into three parts: (a) interpretation of main findings and comparison of the TD and Maintenance datasets; (b) implications for practitioners; and (c) implications for researchers.

***Interpretation of Findings***: First, based on our data extraction, we have validated that TD Interest quantification is supported by substantially less mathematical models / equations compared to Maintenance Cost estimation. This finding supports the fact that TD Interest is more difficult to quantify, but this is also supported by the early maturity of the maintenance compared to the TD community. Concerning quantification parameters, we synthesize the findings in Figure 2 (#studies) and Figure 3 (#of metrics). By observing the figures, three main findings can be drawn. First, the ***two aspects*** of quality ***are affected by many overlapping factors***: all TD Interest factors have already been studied by the maintenance community (except TD Principal); whereas on the other hand 4 (Performance, Polymorphism, Run-Time Qualities, and Human Factors) out of 13 Maintenance Estimation parameters have been explored by the TD community. This finding suggests that there is not a lot of room for transfer of knowledge from one community to the other. Second, despite the overlap of metrics, ***the importance of the factors*** in quantifying the two quality aspects ***are different***. TD Interest seems to be more related to Business Factors and Size; whereas Maintenance Cost are more related to Size and Complexity. This finding suggests that TD Interest is more related to the business parameters, since Size is also used to capture functional or evolution aspects. This difference in the nature of the two aspects is also evident from the focus of TD community to quantify Interest in the form of Currency. Finally, by comparing the ***number of studies against number of metrics***, we can observe only minor differences. The most prominent of which is Correctness for which only one study exists in the TD literature; suggesting however, 8 metrics. This observation makes the findings on correctness less generalizable.

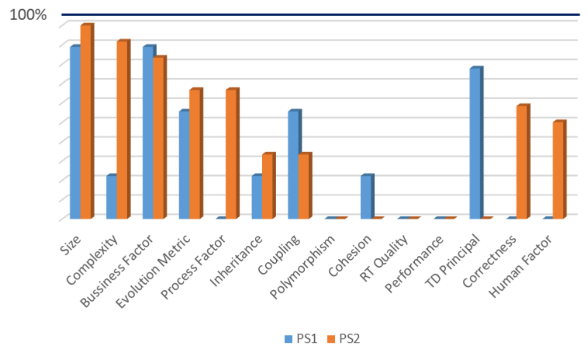


Fig. 2. DIFFERENCES BETWEEN PS1 & PS2 (#STUDIES)

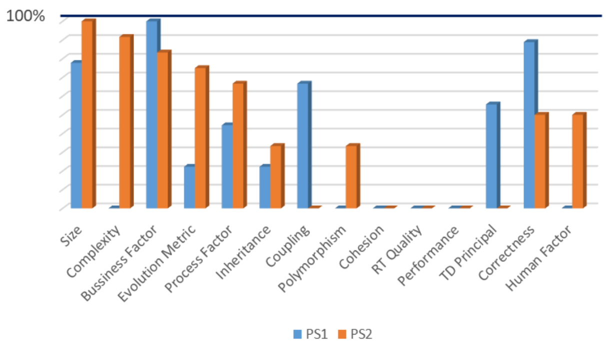


Fig. 3. DIFFERENCES BETWEEN PS1 & PS2 (#METRICS)

[3] Due to the page limitation, we provide all the formulas that quantify the Maintenance Cost in the online supplementary material of footnote 2.

***Implications to Researchers***: The findings of the study suggest that despite the existence of some mathematical models for quantifying TD Interest in the literature, these models are quite diverse in factors. Nevertheless, some important observations can be made to guide future endeavors: (a) models must consider both business and structural parameters. The business parameters are important to capture the financial perspective of TD, whereas the structural parameters are necessary for denoting the lag in terms of quality, due to the existence of TD; (b) models are closer to the metaphor, if they are calculating TD Interest in a monetary measurement unit; and (c) models must consider the evolution of the system in terms of identifying changing parts of the code, as well as the usual load of maintenance. We note that all future models must be accompanied by tools that will enable the automated application of the models in an industrial setting. As a concluding remark, we believe that additional models are needed, but also there is a need to validate existing models with industrial data, and perform empirical studies that use them to explore the TD phenomenon more comprehensively.

***Implications to Practitioners***: Based on the findings of our review, we encourage practitioners to pay attention on the factors that are crucial for generating TD Interest. Out of the major factors, the structural ones are more easily controllable by practitioners through refactorings. Therefore, we champion the adoption of maintenance community tools (such as refactoring opportunity detectors) that can improve structural quality and control TD, especially in parts of the system that change often. Upon the release of stable TD Interest calculation tools, we strongly suggest the combination of TD Principal and TD Interest quantification tools, so as to guide decision making in terms of TD management.

## VI. Threats To Validity

In this section we present the threats to validity which are organized based on the guidelines for identifying, mitigating, and reporting threats to validity for secondary studies in software engineering proposed by Ampatzoglou et al. [26].

***Study Selection Process***. To guarantee that our search process adequately identified all relevant studies, we use a protocol based on strict guidelines [27]. The search string was systematically constructed, in the sense that we have used the term "technical debt" in title combined with the term "interest" in the abstract for the first set of primary studies, whereas in the second set the search string focuses on the "*Maintenance Cost*". However, it could be possible to exclude studies that have used different terminology from the more established ones. Additionally, the inclusion and exclusion criteria have been extensively reviewed and piloted, by the authors to avoid misunderstandings and to ensure their clarity. Also, we have excluded grey literature, since we intent to focus only on published academic literature, guaranteeing some level of rigor and relevance. Our study is not suffering from missing non-English papers and we were able to access all publications because our institution provide access to digital libraries.

***Data Validity***. In terms of data validity, the main threat is related to data extraction bias and the selection of specific venues. Regarding the data extraction, all relevant data were extracted and recorded manually by the third and the fourth authors. In order to avoid the subjectivity in this process, the fist and the second authors reviewed and further refined the collected data, re-validating them. After this process, the results were discussed among all authors and they resolved any conflicts. Additionally, there is no publication bias in the selected studies, in the sense that the primary studies have been retrieved by various venues, covering both the TD and the Maintenance communities. Thus, the aforementioned studies are not affected by a closed and small circle of researchers. Our mapping study is not affected from the following threats: (a) small sample size, as it became possible to recover 25 articles; (b) lack of relationships, the study did not aim to identify relationships between data, but only to classify and compose; and (c) the selection of variables to be extracted, as the research questions of this study did not create disagreements in the discussions between authors based on the variables to be extracted. Furthermore, we did not find problems with the use of statistical analysis, since the nature of our research questions did not require hypothesis testing, but only basic descriptive analysis. Finally, to mitigate the researchers' bias in data interpretation and analysis, all the authors discussed the data categorization, based on the HL categories of the identified metrics.

***Research Validity***. In terms of research validity, to increase the reliability and replicability of the study, we involved more than one researcher to all steps of the process, and all data have been made available. Additionally, we ensured that the correct research method has been used, i.e., an SMS since only limited synthesis was required to achieve the high-level goals of this study. However, we acknowledge that the lack of direct related work has not allowed comparison of results; however, the experience of the authors on TDM research allowed interpretation of results, increasing generalizability.

## VII. Conclusions

One of the main pillars of the Technical Debt metaphor is the notion of interest, which expresses the additional effort to be spent in future maintenance, because of the inefficiencies in the current code, design or architecture. However, estimating TD interest is extremely challenging as it requires the anticipation of future software changes and the quantification of how much more maintenance effort these changes would require compared to a zero-TD system. In this paper we survey existing approaches for quantifying TD interest by means of a Systematic Mapping Study.

The results revealed that there is indeed a lack of mathematical/analytical approaches for estimating TD interest. Furthermore, the factors which are considered in TD interest calculation approaches are not novel; to a large extent they are common to the factors taken into account when estimating maintenance effort in general, and include mainly size, code/design quality measures and business parameters. We encourage both researchers and practitioners to systematically record data related to spent maintenance effort in hope that empirical models will be able to capture this highly valuable, yet hard-to-quantify aspect of Technical Debt.

## Acknowledgment

This work has received funding from the European Union's H2020 research and innovation programme, under grant agreement: 871177 (SmartCLIDE).

## REFERENCES

[1] W. Cunningham, "The WyCash portfolio management system", *7th International Conference on Object-oriented Programming Systems Languages and Applications (OOPSLA '92),* ACM, Vancuver, Canada, April 1992.

[2] P. Avgeriou et al., "An Overview and Comparison of Technical Debt Measurement Tools," *IEEE Software*, vol. 38, no. 3, pp. 61-71, May-June 2021.

[3] M. T. Baldassarre, V. Lenarduzzi, S. Romano, N. Saarimäki, "On the diffuseness of technical debt items and accuracy of remediation time when using SonarQube", *Information and Software Technology*, Elsevier, 128, 2020.

[4] T. Amanatidis, N. Mittas, A. Moschou, et al. "Evaluating the Agreement among Technical Debt Measurement Tools: Building an Empirical Benchmark of Technical Debt Liabilities", *Empirical Software Engineering*, Springer, 25, pp. 4161–4204, 2020.

[5] J. Lefever, Y. Cai, H. Cervantes, R. Kazman and H. Fang, "On the Lack of Consensus Among Technical Debt Detection Tools", 43rd International Conference on Software Engineering: *Software Engineering in Practice (ICSE-SEIP)*, pp. 121-130, 2021.

[6] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou, and P. Avgeriou, "The financial aspect of managing technical debt: A systematic literature review", *Information and Software Technology*, Elsevier, 64, pp. 52-73, 2015.

[7] Tom, E., Aurum, A., Vidgen, R., "An exploration of technical debt", *Journal of Sysems and Software*, 86 (6), pp. 1498–1516, 2013.

[8] Li, Z., Avgeriou, P., Liang, P., "A systematic mapping study on technical debt and its management", *Journal of Sysems and Software,* 101, pp. 193–220, 2015.

[9] Ampatzoglou, Ar., Ampatzoglou A., Avgeriou, P. and Chatzigeorgiou, A., "Establishing a Framework for Managing Interest in Technical Debt", *5th International Symposium on Business Modeling and Software Design (BMSD' 15)*, Milan, Italy, July 6–8, 2015

[10] Alves, N. S., Mendes, T.S., de Mendonça, M.G., Spínola, R.O., Shull, F., Seaman, C., "Identification and management of technical debt: A systematic mapping study", *Information and Software Technology*, 70, pp. 100–121, 2016.

[11] Fernández-Sánchez, C., Garbajosa, J., Yage, A., Perez, J., "Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study", *Journal of Sysems and Software,* 124, pp. 22–38, 2017.

[12] Behutiye, W.N., Rodríguez, P., Oivo, M., Tosun, A., "Analyzing the concept of technical debt in the context of agile software development: A systematic literature review", *Information and Software Technology,* 82, pp. 139–158. 2017.

[13] Besker, T., Martini, A., Bosch, J., "Managing architectural technical debt: A unified model and systematic literature review". *Journal of Sysems and Software,* 135, pp. 1–16, 2018.

[14] Lenarduzzi V., Besker, T., Taibi, D., Martini, A. and Fontana, F. A., "A systematic literature review on Technical Debt prioritization: Strategies, processes, factors, and tools", *Journal of Sysems and Software*, 171, 2021.

[15] Saraiva, D., Neto, J., Kulesza, U., Freitas, G., Reboucas, R. and Coelho, R., "Saraiva, D., Neto, J., Kulesza, U., Freitas, G., Reboucas, R. and Coelho, R.", *23rd International Conference on Enterprise Information Systems ('21),* Online Streaming.

[16] Pavlic, L. and Hlis, T., "The Technical Debt Management Tools Comparison", *Software Quality Analysis, Monitoring, Improvement, and Applications (SQAMIA'19)*, Ohrid, North Macedonia, 22-25 September 2019.

[17] Fontana F. A., Roveda, R. and Zanoni, M., "Technical Debt Indexes Provided by Tools: A Preliminary Discussion", *8th International Workshop on Managing Technical Debt (MTD'16)*, Raleigh, USA, 4 Octomber 2016.

[18] Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M. "Systematic mapping studies in software engineering", *12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08)*, British Computer Society Swinton, Bari, Italy, pp. 68-77, 26 - 27 June 2008.

[19] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: a systematic review", *Information and Software Technoly*, Elsevier, 50 (9), pp. 833–859, 2008.

[20] R. Farhoodi, V. Garousi, D. Pfahl, and J. Sillito, "Development of Scientific Software: A Systematic Mapping, a Bibliometrics Study, and a Paper Repository", *International Journal of Software Engineering and Knowledge Engineering*, 23 (4), 2013.

[21] D. Spencer, "Card Sorting: Designing Usable Categories", *Rosenfeld Media*, 1st Edition, April 2009.

[22] P. Kruchten, R. L. Nord, I. Ozkaya, and D. Falessi, "Technical debt: towards a crisper definition", *4th International Workshop on Managing Technical Debt*, ACM SIGSOFT Softw. Eng. Notes, vol. 38, no. 5, p. 51, Aug. 2013.

[23] V. Antinyan et al., "Identifying risky areas of software code in Agile/Lean software development: An industrial experience report", *18th International Working Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE '14)*, IEEE, pp. 154-163, 2014.

[24] A. MacCormack, J. Rusnak, and C. Baldwin, "Exploring the Duality between Product and Organizational Architectures: A Test of the Mirroring Hypothesis (Version 3.0)", *Harvard Business School*, 2008.

[25] M. Riaz, E. Mendes, E. Tempero "A systematic review on software maintainability prediction and metrics", *3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09)*, IEEE Computer Society, Florida, USA (2009), pp. 367-377.

[26] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, and A. Chatzigeorgiou, "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies", *Information and Software Technology*, Elsevier, 106 (2), pp. 201-230, 2019.

[27] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering", *Technical Report EBSE 2007-001*, Keele University and Durham University, 2007.


## APPENDIX – TD INTEREST PRIMARY STUDIES

[S1] Ampatzoglou, A., Michailidis, A., Sarikyriakidis, C., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P. "A framework for managing interest in technical debt: An industrial validation", *International Conference on Technical Debt (TechDebt'18) ,ICSE*, 27-28 May 2018, Gothenburg, Sweden.

[S2] Falessi, D., Reichel, A. "Towards an open-source tool for measuring and visualizing the interest of technical debt", *7th International Workshop on Managing Technical Debt (MTD'15)*, 2 Oct 2015, Bremen, Germany.

[S3] Guo, Y., Seaman, C., Gomes, R., Cavalcanti, A., Tonin, G., Da Silva, F.Q.B., Santos, A.L.M., Siebra, C. "Tracking technical debt - An exploratory case study", *27th International Conference on Software Maintenance (ICSM'11)*, 25-30 Sep 2011, Williamsburg, VA, United States.

[S4] Martini, A., Sikander, E., Madlani, N. "A semi-automated framework for the identification and estimation of Architectural Technical Debt: A comparative case-study on the modularization of a software component", *Information and Software Technology*, 93, pp.264-279, January 2018.

[S5] Nord, R.L., Ozkaya, I., Kruchten, P., Gonzalez-Rojas, M. "In search of a metric for managing architectural technical debt", *Joint 10th Working IEEE/IFIP Conference on Software Architecture (WICSA'12)*, 20-24 Aug 2012, Helsinki, Finland.

[S6] Nugroho, A., Visser, J., Kuipers, T. "An empirical model of technical debt and interest", *2nd International Workshop on Managing Technical Debt (MTD'11)*, 23 May 2011, Waikiki, Honolulu, HI, United States.