



Contents lists available at ScienceDirect

## The Journal of Systems and Software

journal homepage: [www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss)

# Costs and obstacles encountered in technical debt management – A case study

Yuepu Guo<sup>a,\*</sup>, Carolyn Seaman<sup>a</sup>, Fabio Q.B. da Silva<sup>b</sup>

<sup>a</sup> Department of Information Systems, UMBC, Baltimore, MD, USA

<sup>b</sup> Center for Informatics, Federal University of Pernambuco, Recife, PE, Brazil

## ARTICLE INFO

### Article history:

Received 1 June 2015

Revised 29 June 2016

Accepted 6 July 2016

Available online xxx

### Keywords:

Software technical debt  
Technical debt management  
Case study  
Software maintenance  
Project management

## ABSTRACT

Technical debt (TD) is a metaphor that characterizes the effect of immature software artifacts. The costs and benefits of TD, along with the uncertainty of its interest repayment, provide leverage for software managers, but also could lead to problems such as increased costs and lower quality during maintenance if it is left unattended. Therefore, effective approaches to TD management are needed by software practitioners. As one of our series of studies on TD management, this study was originally designed to reveal the cost side of explicit TD management. The study design required applying a simple proposed TD management approach to the subject project and then collecting cost information. Not surprisingly, we observed some deviation of the actual management process from our proposed one, which provided us with an opportunity to investigate the obstacles to explicitly managing TD. We also identified some costs and cost patterns related to TD management. Based on the insights gained from this study, we further propose strategies to overcome the obstacles and improve the application of TD management in practice.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

Technical debt (TD) is a metaphor that characterizes the effect of software artifacts that are immature in some aspect. Such artifacts most commonly result from delaying or shortcutting development and maintenance tasks, either strategically to gain marketing advantage, or accidentally due to developer shortcomings. The effects of immature artifacts come in a variety of forms, from quicker software releases in the short run, to increased software complexity and stretched project schedule in the long run. The effects can be both positive and negative, and so there are both costs and benefits associated with TD. These costs and benefits, although varied, can all be monetized at least in theory (although often difficult in practice), thus allowing the debt metaphor to provide a common unit of analysis. The TD metaphor is useful in communicating a fundamental software development and maintenance dilemma, i.e. short-term benefit vs. long-term cost. TD refers to those short-term benefits (usually involving delaying development or maintenance tasks) that might lead to long-term costs (usually in the form of higher maintenance costs in the future).

Although this analogy comes from the finance domain, TD is not exactly the same as finance debt. A major difference between

TD and financial debt is the uncertainty of TD repayment. The interest associated with TD, i.e. the future cost of incurring TD, and even the TD principal, may or may not need to be paid off, which usually does not happen for financial debt. For example, a module with some code smells (due to a rushed modification) may not need to be refactored if no future change is required for this module (assuming, for the sake of simplicity, that the code smells do not directly affect other modules). This uncertainty further complicates the management of software projects, but it also provides an opportunity for project managers. TD does no harm to, or can even benefit, a software project as long as the interest does not need to be paid. However, it is rarely known when or if the interest will come due, which leads to the central dilemma of TD management – when and what TD should be paid in order to minimize the overall expected cost of a software project.

Currently, TD information (including known existing TD, its location, its severity, its potential impact, etc.) is often used implicitly in software project management, based on the software managers' past experience or even instinct. For large software projects, it's easy to lose track of TD or misunderstand its impact. The results could be budget overruns, project delays or even project failure. Therefore, we have developed a general framework, an approach that proposes managing TD in an explicit manner – identifying TD, measuring its cost, i.e. principal and interest, and making the decision to pay off TD through cost-benefit analysis (Seaman and Guo 2011).

\* Corresponding author.

E-mail addresses: [yuepu.guo@umbc.edu](mailto:yuepu.guo@umbc.edu) (Y. Guo), [cseaman@umbc.edu](mailto:cseaman@umbc.edu) (C. Seaman), [fabio@cin.ufpe.br](mailto:fabio@cin.ufpe.br) (F. Q.B. da Silva).

It should be noted though that explicitly managing TD incurs costs, beyond the principal and interest associated with the TD itself. Thus, it makes sense to examine the costs and benefits, not just of TD itself, but also of *managing* TD, which might be termed management overhead associated with TD. The cost-effectiveness of explicit TD management depends on whether the assumed benefits (better decision making about TD) can be achieved and outweigh the costs (management overhead) of using this approach. Therefore, cost and benefit are the two sides that should be investigated in order to formalize and evaluate TD management. Accordingly, we are conducting two types of studies, one of which investigates the benefits that can be gained from explicit TD management, while the other focuses on the cost side of TD management. The study described in this article is of the second type. In particular, this study was originally designed to address the following research question:

- (1) What are the costs of explicitly managing TD using the proposed approach?

In this study, we applied a simple but explicit TD management approach (described in our previous work (Seaman and Guo 2011) and summarized in Section 3) to an ongoing software project, then collected and analyzed data about various costs incurred in the course of TD management. Other information such as the TD items identified from the project and the decisions on carrying or paying off the TD, were also collected to help provide context.

As the study proceeded, especially in the data collection phase, we found that the project team we were studying encountered obstacles to applying the proposed approach to their project and the actual process deviated from what our approach prescribed. This was not surprising, as the proposed approach was designed to be simple and basic, allowing for future refinement as it was evaluated in practice. However, to some extent, this hindered us from fully addressing the original research question above, concerning the costs of TD management. We were able to collect and analyze some data addressing costs, and we interpret our results along with results from a previous similar study (Guo et al. 2016). On the other hand, the process deviation we observed also raised new and interesting questions on managing TD. We believe that revealing the causes behind the obstacles to applying explicit TD management in a software project could smooth technology transfer from academic research to industrial practice and thus improve adoption of TD management in industry. Moreover, the exploratory nature of this study design allowed us to direct the study to new targets. Therefore, we added the following research questions to the study:

- (1) What are the obstacles to explicit TD management?
- (2) What contributes to the deviation of the proposed TD management process?

The opportunity to address these two questions allowed us to expand our findings to further facilitate future efforts at explicit TD management.

In the following sections, we first review some important work from the related research areas as well as an earlier study of the same type we conducted on TD management. In Section 3, we present the TD management approach we proposed, followed by a detailed description of the study design, including subject selection, implementation process, data collection and analysis procedure, in Section 4. Then we give the results of this study in Section 5. Finally, we summarize the findings by research question, and discuss the implications and limitations of this study in the conclusion section.

## 2. Related work

The term “technical debt” was introduced by Cunningham in 1992, where he compared “not-quite-right-code” to TD and any work built on the code to interest on that debt (Cunningham 1992). He also addressed the two sides of TD as “a little debt” speeding up the development while accumulated debt bringing the project to a standstill. TD originally referred to poor system design or implementation, but this metaphor has been extended to include any immature artifacts in the software development lifecycle.

We have designed two types of studies on the TD management problem. We have previously carried out another case study similar to this one, i.e. focusing on the cost side of TD management (Guo et al. 2016). Like the present study design, the previous study required application of our proposed TD management approach to an ongoing software project and then collection of cost and decision making information. The study identified various costs associated with TD management – identification of TD, analysis and evaluation of the identified TD instances, communication between project members regarding TD management, documentation of changes to the TD list, and decisions on TD repayment. Among these costs, analysis and evaluation accounted for the largest portion of effort. Identification also incurred high cost. These two types of cost accounted for the major cost of explicit TD management and formed a significant initial overhead for TD management practices. The study also reported several important decision factors in TD management, including customer agreements, availability of development resource, the value of the debt, the current status of the debt-infected modules, and the impact of the debt on other features. In spite of these findings, the generalization power of the study was limited by the selected case – a relatively small and young software project, which may be different from a large and mature project in terms of cost patterns and management style.

The present study, then, is complementary to the one described above (Guo et al. 2016). The studied project has a very different profile in terms of project cost, development cycle, etc. In this sense the two studies together contribute to understanding the cost side of TD management.

Early TD research, including the work of Rothman (2006), Fowler (2009) and McConnell (2007) established the foundation of the area through conceptualization and classification. A variety of TD identification approaches and tools (Emden and Moonen 2002, Marinescu 2012, Bohnet and Doellner 2011, Letouzey and Ilkiewicz 2012, Brondum and Zhu 2012, Wang et al. 2013, Li et al. 2015), based on different theories and targeted to different types of TD, have been proposed, making TD identification one of the most fruitful areas of achievement in TD research. Likewise, various TD quantification approaches were proposed, including Nugroho et al.’s approach (2011), Brown et al.’s framework (2011), Letouzey’s approach (2012), and Nord et al.’s measurement approach (2012).

Although TD identification and quantification is not the focus of this study, they are essential for TD decision making and management. Strategies for paying off TD, which is closer to the focus of this study, were also proposed by software practitioners and researchers (Lester 2008, Zazworka et al. 2011, Nord et al. 2012, Bachmann et al. 2012, Schmid 2013). Among these management approaches and strategies, some are promising from a theoretical perspective, but most of them have limited power due to the stringent assumptions or application conditions. Moreover, none of these approaches in the literature have been tried in practice, which leads to the objective of our previous study (Guo et al. 2016) as well as this study – developing and empirically exploring a TD management approach in terms of its overhead costs. In a recent case study, Martini et al. (2015) explored TD management in a real

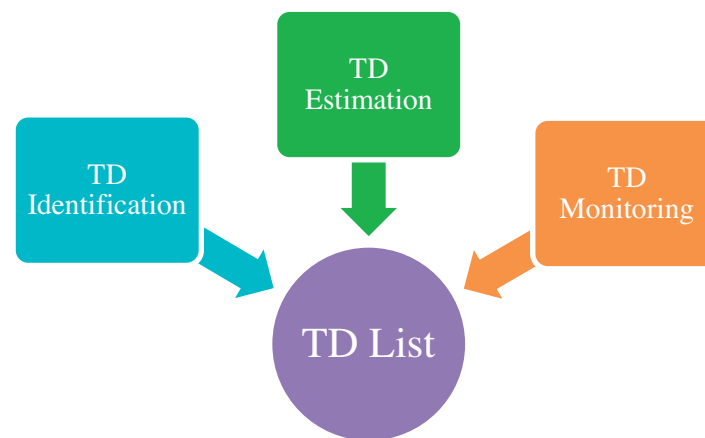


Fig. 1. TD management framework (Seaman and Guo 2011).

industrial environment by identifying causes of architecture debt, modeling debt accumulation, and evaluating different refactoring strategies in terms of the frequency of “crises” as defined by the authors. Thus, Martini et al.’s study comes closest to our study. However, the study didn’t focus on the cost aspect of TD management. Therefore, the management strategies they discussed in the study do not address such questions as when and what TD should be paid off. Decisions about paying off TD require information about the costs and benefits of TD, which our current and previous studies are targeted to answer.

Since one goal of this study is to reveal the causes behind the obstacles to applying the proposed TD management approach, it’s necessary to refer to fundamental theories of technology acceptance by users. According to the Technology Acceptance Model (TAM), two major factors that affect user acceptance of a new technology are perceived usefulness and ease of use (Davis 1989). The model was later expanded as the Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh et al. 2003). The theory was formulated with four core determinants of usage intention: (1) performance expectancy, (2) effort expectancy, (3) social influence, and (4) facilitating conditions. Determinants (1) and (2) correspond to the two major factors in TAM. Inspired by this theory, we incorporated these two factors into our study design, including the interview questionnaire and coding (data analysis). Also, one of our findings is related to determinant (4), showing it to be an obstacle to the application of TD management in the subject project. Closer to the area of TD management, a study on the causes of the difficulties in transferring research results for requirements engineering to the software industry (Kaindi et al. 2002) revealed a set of inhibitors:

- The provider of the technology lacking tools to support the proposed methods,
- The consumer of the technology facing frequently changing requirements,
- Insufficient funds and time for trying new methods under the pressure of business competition, and
- Concerns about lack of technical support.

These findings helped us understand the relationship among individual factors we found in our data, and identify some causes of the obstacles we discovered.

### 3. Proposed TD management approach

One goal of this study was to identify costs associated with explicit TD management. Although, in theory, we were interested in

such costs independent of the process or approach used to explicitly manage TD, in practice we needed to provide a management approach that our subject project could adopt. To this end, we describe in our previous work (Seaman and Guo 2011) a simple TD management framework. The framework is designed to be flexible and easily modified. It is a starting point from which to evolve based on the results of applying it. We summarize this framework here, but a fuller description can be found in our previous work (Seaman and Guo 2011).

As shown in Fig. 1, the proposed framework centers on a “TD list”, which contains a set of TD “items”, i.e. TD instances. These items represent delayed software development or maintenance tasks that run the risk of causing problems such as increased costs and lower quality during future maintenance. Each item is described by a set of properties, the most relevant of which are estimates of the principal and the interest. The principal refers to the cost of eliminating the debt or, seen another way, the amount of effort saved when the task was postponed. The interest is composed of two parts – interest probability and interest amount. Interest probability refers to the probability that a debt item will cause other work to be more expensive than it otherwise would be over a given period of time or a release. The “interest amount” quantifies the extra work that may be needed in the future if this debt item is not paid off.

In this approach, estimation of a TD item, i.e. principal, interest probability and interest amount, takes on high-medium-low value, a course-grained scale, in the initial stage. This is sufficient for tracking TD items and making preliminary decisions (Seaman and Guo 2011).

The identification and estimation of TD are not the subject of this article. The study we describe here was concerned with the monitoring step (in Fig. 1), which includes making decisions on when and what debt items should be paid or deferred. Due to the dynamic nature of TD, i.e., existing debt items are paid off and new debt items are incurred over the course of the software lifecycle, TD should be continuously monitored. The TD list should be updated to reflect these changes.

Given that the goal of identifying and measuring TD is to facilitate decision making, we chose to apply our proposed approach to release planning, a classic scenario in software project management, to help managers make optimal decisions using TD information. In release planning, a decision must be made regarding whether and what TD items should be paid off in the upcoming release (or iteration, or sprint, depending on the process terminology being used). Our approach relies on a straightforward cost-benefit analysis for each TD item, using the estimated principal to represent the cost of paying off that item, and the estimated

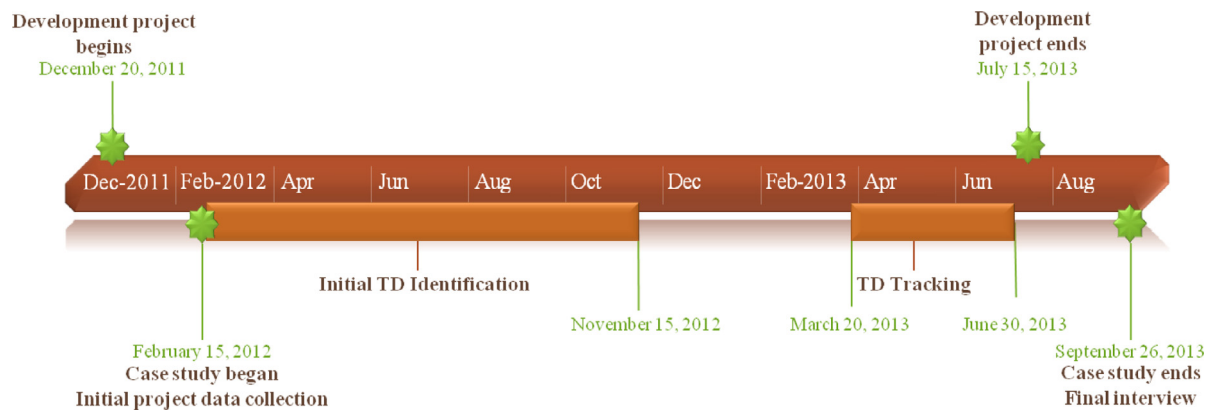


Fig. 2. Important project and case study dates.

expected interest (the product of the interest amount and probability) to represent the benefit, i.e. the cost avoided in the future by paying off the item now. Such a cost-benefit analysis requires numeric estimates of both principal and interest for all TD items analyzed, which is difficult and time-consuming. To reduce this burden, the approach advocates first filtering the TD list to isolate those items most likely to be found to be worth paying off. First, the list is filtered for items that relate to components already being touched in the current release, with the idea that combining planned maintenance work with TD repayment in the same component will be more efficient. The next filter is for items with high interest (both probability and amount), as these are the most likely to have greater impact downstream. The remaining TD items are then analyzed in more depth, first by numerically estimating principal and interest, and then comparing costs and benefits. The items for which the benefit outweighs the cost are then included in the release plan.

#### 4. Methodology

This study was designed primarily to understand the costs of explicit TD management (Research Question 1) by applying the approach (described in Section 3) to an ongoing software project through several release cycles in real time. Deviations in the application of the TD management process allowed us the opportunity to also focus on obstacles to incorporating the proposed TD management approach in the release planning process (Research Questions 2 and 3). To take advantage of this opportunity, we enhanced our study instruments to capture data on obstacles, e.g. our final interview with the project leader focused on this issue.

##### 4.1. Subject project

The subject project was selected by convenience by our industry collaborator in Brazil. It was a software application running on tablet computers on the Android operating system. The purpose of this software application was to provide an education platform for teachers and students in a school to share information. The platform integrates with other applications such as Moodle, EverNote and Dropbox. It provides call list, note list, messaging service, file sharing and access to libraries. Originally the project planned to carry out 17 sprints. When we started collecting data, the project consisted of 45 user stories with 18 features, implemented in 178,195 LOC (Lines of Code), but it grew over time. The entire application was written in Java.

The subject project was developed by a company (Developer) that developed software systems for a multinational organization of the mobile communication market (Client). Developer and Client

were located in different places in the same country and most interactions between the business team on the Client side and the development team occurred over distance, using conferencing tools. At the time of our study, the Developer had approximately 70 employees, with 60 working on software development. The organization of the software teams was flexible to fulfill the demands of the Client and several projects were conducted simultaneously. Two senior (over 15 years of experience) project managers supervised each project. The team developing the subject project was managed by the most senior project manager, who had over 20 years of experience in managing software projects in industry. Interaction with the Client team was conducted by the project manager and project leader, who addressed issues related to feature and release planning, together with the client.

At the beginning of the study, the development team consisted of 1 project manager, 1 project leader and 7 developers. This remained fairly constant, although the person occupying the role of project leader changed once, and some developers came and went from one sprint to the next. The project used Scrum with each sprint lasting about 4 weeks. The timeline of important project and case study dates is shown in Fig. 2. The project began on December 20, 2011. The case study began in the middle of February, 2012, with collection of basic project information. Afterwards, the project team started to identify TD items. For various reasons, the TD identification phase lasted until the middle of November, 2012. The project team started tracking TD on March 20, 2013. We observed and collected data on the TD management process until the end of June, 2013. The case study ended in September, 2013 with a follow-up interview.

##### 4.2. Case study process

From the beginning of the study until the end of TD identification, the principal researcher (first author) directly communicated with the Developer contact via email and teleconference (three persons were involved due to project personnel changes). When the project team started tracking TD for release planning, the second author was working on-site with the project team. So she was in charge of observing and supervising the project team in performing TD management. Then the first author took over responsibility and conducted a follow-up interview with the project leader at the end of the study. The third author's role was to facilitate participation of the subject project in the study, help with communication, and overall supervision of the study. The project personnel were all native Portuguese speakers, with varying levels of English proficiency. All communication between the first author and the company contact was in English. The majority of the communication between the second author and the project team was also



| ID                            | Responsible | Type          | Location   | Description   | Principal | Interest Amount | Interest Probability |
|-------------------------------|-------------|---------------|--|---|-----------|-----------------|----------------------|
| <b>Currently Active Items</b> |             |               |  |   |           |                 |                      |
| 10                            | Developer   | Design        | All of ListViews Adapters used in the project.<br>All classes that extend ThemedArrayAdapter | We could use a parent class for generic adapters. In the current project there are several classes with similar behavior. | Medium    | Medium          | Low                  |
| 15                            | Tech Leader | Documentation | Architecture document  | Better documentation of the internal architecture of EducationHub.  | Medium    | Low             | Low                  |
| 19                            | Developer   | Design        | Widget, WidgetsController  | Refactor the creation and control of widgets.   | High      | Medium          | High                 |
| 20                            | Designer    | Design        | All project screens  | The interface design is ugly and poorly designed. Serious usability problems.   | High      | High            | Low                  |
| 6                             | Developer   | Design        | Ui.messages  | Refactor the part wondering messaging to improve code readability.  | Medium    | Medium          | Medium               |

Fig. 3. TD list extract.

in English, with some Portuguese as necessary. The third author communicated with the project team primarily in Portuguese. All communication among the three authors was in English.

The case study was conducted with the following general steps:

- (1) The company contact collected basic project information such as project description, requirements and project schedule.
- (2) We designed a simple spreadsheet as the TD list to ease management of TD items and to collect data on its use. The list was kept under version control, along with other project documents.
- (3) The project team were trained on how to manage TD using the proposed approach and how to document and report information required by this study. Training materials were prepared by the two researchers in English, and then was translated to Portuguese and presented by the Developer contact.
- (4) The project team identified TD items in the subject system to prepare the initial TD list. This effort involved surveys of the development team, application of some source code analysis tools, and examination of the existing project backlog.
- (5) In the 13th release cycle, during which the project was tracking TD for the first time, the second author attended the sprint planning meeting, observed and took notes on the time spent by the project team on managing TD, the project members involved, and the decisions they made in the course of analyzing TD. After the meeting, the project team sent an updated TD list to the researchers.
- (6) After each of the planning meetings for sprints 14 and 15, we followed up with the project leader and other team members to collect data on the meeting content related to TD management. We also asked for information on the process followed during the planning meeting, as it deviated from the proposed TD management process. We also elicited the reasons for the deviation.
- (7) The principal researcher (first author) interviewed the project leader after the completion of the 15th sprint, which was not only the last sprint for our study, but also the last one of the project (the project was cancelled soon after).

#### 4.3. Data collection

The first type of data we collected was project documentation, which included a general project description, project requirements

and high-level design, and project schedule. This type of data was collected at the beginning of the study.

The second type of data was regarding the existing TD instances in the subject project. It was collected in the form of a TD list, as shown in Fig. 3. The TD list contained the description, responsible persons, location, principal, interest amount, and interest probability. The items were grouped by their current status (all the items that appear in Fig. 3 are “currently active” items). Over the course of this study, we collected 6 versions of the TD list.

Field notes are the largest category of data we collected. They contain the project team’s activities regarding TD management, e.g. estimating principal and interest, time spent on these activities, decisions made on the TD items and other situations related to the study such as project members’ opinions, obstacles they encountered and deviations we observed.

Interviews were another type of data we collected. We conducted two interviews. The first one was conducted on-site by the second author in the middle of the study with the project leader. This interview was semi-structured and followed an interview guide that included questions about the details of the previous sprint planning meeting, activities related to TD management outside the sprint planning meeting, and in general how TD management was going. This interview was brief (about 20 min) and was not recorded, but was immediately transcribed into English by the second author. Afterwards we expanded the interview guide with follow-up questions and sent it to the project leader via email (this version is in Appendix A), and the project leader responded via email. The focus of that interview was how the decisions on paying off TD were made (given that the proposed TD management process was not yet followed). The second interview was conducted with the same person, i.e. the project leader, at the end of the study (it is in the Appendix B). All of the questions and the response are in English. The questions were sent out and answered via email. It covered the interviewee’s profile, the project’s original TD management practices, obstacles of incorporating the proposed TD management approach and his opinion on the usefulness of the approach. The main goal of this interview was to identify obstacles and find out how to improve the adoption of TD management.

#### 4.4. Data analysis

Since most of the data we collected are qualitative in nature, e.g. open-ended questions, correspondence and field notes, coding was the primary approach for data analysis. The initial codes and categories were pre-formed based on the research questions. There are four main categories: cost, benefit, obstacle and process devia-

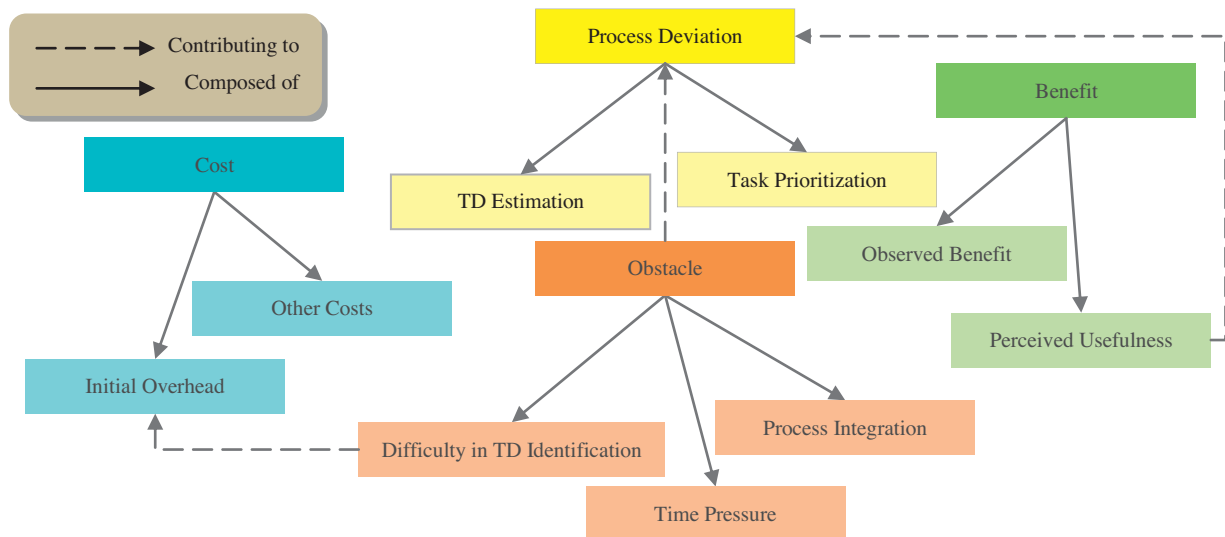


Fig. 4. Axial coding.

Table 1  
Coding scheme.

| Category          | Code  |
|-------------------|---|
| Cost              | Initial overhead<br>Other costs   |
| Benefit           | Observed benefit<br>Perceived usefulness                                |
| Obstacle          | Difficulty in TD identification<br>Time pressure<br>Process integration |
| Process deviation | TD estimation<br>Task prioritization                                    |

the project members, we found it useful to differentiate between observed benefit and perceived usefulness. The complete coding scheme is presented in Table 1.

The next step of the coding, axial coding, was to re-assemble these codes and categories and identify the relationships between them. Fig. 4 represents the results from axial coding. By analyzing the data, we found out there are different types of obstacles to explicit TD management, one of which is cost-related. The perceived usefulness (or lack thereof) of explicit TD management also hinders the application of the TD management approach. These obstacles eventually caused the process deviation. The detailed results are presented in the next section.

## 5. Results and findings

Through data analysis, we identified three major themes regarding TD management – costs of TD management, obstacles to applying explicit TD management to the project, and deviation of the actual TD management process from the proposed one. These three themes easily map to Research Questions 1, 2, and 3, respectively, and are discussed from that perspective in Section 6. By digging into the details of the data, we found various aspects within each theme. The axial coding of the data further revealed the relationships between the themes and the aspects, as shown in Fig. 4. The themes, aspects and relationships together helped us understand the rationales behind the TD management practice and hence address the research questions of this study. The validity of this study is also discussed at the end of this section.

### 5.1. Costs of managing TD

There are different costs tied to TD management. We identified these cost categories mainly through observation of how the project team dealt with the TD in the subject project. In particular, the field notes taken around the first sprint planning meeting record how many persons spent how much time in discussing TD related issues, estimating the values of TD items and updating the TD list before, during and after the meeting. The field notes taken around the subsequent sprint planning meetings also contain cost information, but it was not as detailed and exhaustive as the first one. In addition, the project leader, through an interview, provided some cost information such as the participants and the time they spent on TD management. Time and effort associated with the initial phase of TD identification, described below, was communicated

tion, as shown in Table 1. Cost refers to the costs incurred in the project that can be attributed to explicit TD management (Research Question 1). Benefit refers to benefits gained from explicit TD management, including the benefits observed in the project and the benefits perceived by the project team. Benefits of TD management were not a focus of this study or its research questions, but data relevant to benefits was collected and so was coded. In this study we observed that the project team encountered obstacles to using the proposed TD management approach. Therefore, one of the main codes is obstacle, which captures the factors that prevent the project from using explicit TD management (Research Question 2). Another code is process deviation, which refers to the differences between the proposed TD management approach and the actual TD management practice carried out by the project team (Research Question 3). Process deviations included deviations from the way we expected TD information to be used in decision making.

The data were then open coded using the categories defined in Table 1. In this step we identified and added new subcategories and codes from the data in vivo. For example, “initial overhead” was mentioned specifically by different subjects. So we used two specific codes, initial overhead and other costs, to code the cost category. Likewise, we created specific codes under the category “process deviation” to reflect what part of the process deviated, as we found specific references to these parts of the process in the data (“task prioritization” was primarily concerned with TD decision making). The data about obstacles was primarily concerned with three aspects – difficulty in TD identification, time pressure and process integration – and so these became the codes in this category. Since data about the benefits of applying the TD management approach were collected not only from observation of the implementation of the approach, but also from interviews with

**Table 2**  
Costs of TD management.

|   | Sprint 13 |                            | Sprint 14 |              | Sprint 15 |               |
|---|-----------|----------------------------|-----------|--------------|-----------|---------------|
|   | O/M       | I/M {11, 130} <sup>a</sup> | O/M       | I/M {8, 375} | O/M       | I/M {N/A, 90} |
| Updating TD list                          | {1, 30}   | 35 min                     | {2, 10}   | 0 min        | {0, 0}    | 17 min        |
| Sprint planning (percentage) <sup>b</sup> | {2, 5}    | 2 min (2%)                 | {2, 5}    | 0 min (0%)   | {0, 0}    | 3 min (3%)    |
| Communication                             | {2, 15}   |                            | {3, 10}   |              | {2, 10}   |               |

O/M: outside of meeting; I/M: during meeting

<sup>a</sup> The pair in the braces represents the number of involved persons and the minutes of time spent accordingly.

<sup>b</sup> The percentage of sprint planning time that was spent in managing TD.

to the research team via email from our Developer organization contact.

TD management started with identification of the TD instances and initial construction of the TD list. TD identification began in the middle of March, 2012 and continued, on and off, until November 2012 (see Fig. 2). This included periods in which the project was without a permanent project lead, or was suspended for business reasons, or the personnel were reassigned to other projects temporarily, and thus they did not engage in TD identification. TD identification actually took about 3 months of activity, but was spread over an 8-month period. Project members reported TD items in an ad hoc manner at first, but were aided by several other techniques. We analyzed the bug repository and change log of the project to help narrow down the areas of the source code where it was more likely to have TD than other areas. We also conducted a survey among the project members to elicit TD information from their past experience in a more systematic way.

In terms of time span and involved effort, TD identification was the most costly step in the TD management process. One reason is that the concept of TD involved a *learning curve* for the project team. The learning curve was evident when the first set of TD items were identified and reported to the research team, in April 2012. For example, one TD item they identified was regarding the lack of a feature. This was actually a missing requirement, not TD. Another example is that they reported a defect as a TD item. Based on our definition of TD as the effect of immature artifacts, delaying defect fixing could have effects on other development and maintenance activities, and these effects could be considered TD, but a defect in and of itself is not TD. From these and other TD items in the initial TD list, it appeared that the developers had difficulty in associating the abstract concept of TD to concrete forms of TD in reality. It took more than three months for the project team to finish identifying all TD items in the project. This learning curve was confirmed by the project leader in the interview we conducted with him. When we asked him about the obstacles to TD management in their project, he thought that "it was not always very clear what was a TD item and what was not. What I saw as a TD item, sometimes the team thought it was not, and vice versa. Also sometimes I had seen a TD as having been paid, but they thought it was not". Resolving these different interpretations, both within the project team as well as with the research team, took considerable time.

Besides the learning curve, another reason is that *identifying TD is still difficult*. Although various TD identification approaches have been proposed or implemented for software development practitioners, most of the approaches still need to be adapted to be effective for a project in a particular domain (Guo et al. 2010). During TD identification, our research team took various actions, such as analyzing change logs of the project and running a survey, to help the project team identify TD, but it still took more than three months for them to finish this step. When we asked the project leader what the most difficult/time-consuming part of the TD management approach was, he gave a very clear answer – identifying and recording TD. Because TD identification is the

first step of TD management, from the cost perspective we consider this difficult and time-consuming step a big initial overhead for TD management.

Once the TD items were identified, the project team constructed a TD list with information such as description, identification date, principal and interest. Then the project team used the technical list for sprint planning. The costs incurred in this process can be categorized into "updating TD list", "sprint planning" and "communication". "**Updating TD list**" represents the time spent in understanding TD items in the list and determining their current status. In Sprint 13, this occurred during the sprint planning meeting, but in other sprints it occurred outside of the sprint planning meeting. "**Sprint planning**" represents time during sprint planning meetings spent on using the TD list to plan a sprint, primarily deciding what to include in the next sprint. "**Communication**" captures other time, outside the sprint planning meeting, for discussing strategies to manage TD among the project team members.

As shown in Table 2, updating the TD list in Sprint 13 took much more time and involved many more people (highlighted with red color) than in the other two sprints. This was mainly because the TD list was used for the first time and the project team had to go over the entire list more than once to determine the status of each TD item.

There were three sprint planning meetings that we tracked for this study. They included all project team members and ranged from 90 to 375 minutes, but the TD information was barely used for sprint planning (3% or less of the total meeting time). This was where the major process deviation occurred, which we will discuss in more detail in the next section. Communication among project members regarding TD management happened occasionally outside the sprint planning meeting and involved 2–3 persons. Among these types of costs, updating the TD list for the first time accounted for the majority of the TD management cost. It could be viewed as part of the initial overhead we mentioned earlier. The other types of costs were relatively small and hence didn't have significant impact on the project, in comparison with the initial overhead. Therefore the initial overhead was the main cost category of TD management.

## 5.2. Process deviation

The original goal of this study was to explore the costs of explicit TD management. Therefore, the study design required that the project team apply the proposed TD management approach to the subject project, allowing us to track related costs and activities. However, the project team didn't strictly follow the proposed TD management approach, which gave us the opportunity to observe deviations as the project proceeded.

According to the proposed approach, the principal and interest of the items on the TD list should be estimated and the prioritization decision, i.e. which items have higher priority than others for repayment, should be based on cost/benefit analysis of these items. The project team, by contrast, didn't always do estimation

and when they did, the estimates were not used for their decisions. For example, when the first sprint planning meeting was held the principal and interest of the items on the TD list hadn't been estimated. They didn't do any estimation in the meeting, but a decision was made not to pay any TD items in next sprint. This may be because many of the TD items had already been paid off before the first sprint (during the project suspension), but time constraint could be another reason because they had a long list of features that needed to be implemented in the next sprint. A follow-up interview after the meeting confirmed this. The decision to not pay off any TD items in this sprint was actually made before the sprint planning meeting for the same reason. The project team didn't even have enough time to fulfill the requirements. Therefore, they would not pay off any TD items in this sprint unless it overlapped with the requirements.

Instead of estimating and using principal and interest in decisions about paying off TD, other criteria were used to drive these decisions. For example, after the second sprint planning meeting, a senior developer and the project leader discussed implementing more test automation in the next sprint, which was already on the TD list. They also considered restructuring the system architecture. So they updated the TD item about test automation and added restructuring the architecture as a new item to TD list with coarse estimates of its principal and interest. Then they negotiated with the Client to include test automation in the next sprint. For test automation, they did not do numeric estimation for the principal and interest as our approach requires. A follow-up interview with the team leader afterwards revealed that the decision to implement more automated tests had to do with the difficulty (the "pain") of testing, but had more to do with improving the quality of testing. In other words, at least in this case, the priority of a TD item has more to do with quality than cost. While test efficiency might have been fairly straightforward to represent as a type of "interest", the team found it hard to do this with test quality. Thus, our approach of using principal and interest fell by the wayside in making this decision. Similarly, there was no discussion of principal or interest on any TD item during the third sprint planning meeting. The team leader reviewed the entire TD list, added some new items, and made some decisions about paying off items in the meeting, but he didn't follow our approach to prioritize the items and make decisions accordingly. The decision to pay off one item, which was proposed by a developer, was made based on the availability of the development resource, not economic concerns.

In summary, the proposed TD management process was not strictly followed by the project team, which was not completely unexpected. The deviation was primarily in the form of the use of criteria other than a quantitative cost-benefit analysis involving principal and interest. Rather than estimating principal and interest on individual TD items, and then using those estimates to do a cost-benefit analysis for each one, the team chose TD items based on such criteria as current pain points in the project and availability of specific resources. In other words, information about existing TD (location, principal, interest, etc.) was barely used for decision making in release planning.

It was mentioned several times and on different occasions that the compromises the team took in sprint planning, including their approach to TD management, was a result of lacking time. To confirm this point as well as identify other causes of process deviation, we looked into the obstacles that the project team had in managing TD, which are presented in the next subsection.

### 5.3. Obstacles

In this study the project team encountered several obstacles to managing TD using the proposed approach. As we saw in Section 5.1, the difficulty of TD identification was the first major

obstacle for the project team. The project leader confirmed this in an interview, explaining that TD still seemed to be an abstract concept although he thought the training was sufficient. He suggested that "Maybe there should be harder rules for what should be classified as a TD and how to identify it". The difficulty surrounding TD identification not only created a big obstacle for the project team, but also incurred high cost at the beginning of their TD management experience. Meanwhile the project team had low perceived usefulness of TD management to their project, as indicated in an interview with the project leader. When we asked the project leader about his general impression on the TD management approach, his answer was "it's not very impressive". The difficulty and the low expectations of usefulness hindered application of TD management in their project. Three project members, including the leader, were asked about the usefulness of the proposed TD management approach. They all thought that the approach was generally useful, but the leader stressed that it's still hard to apply, especially when facing time pressure, such as "delivering many features in a short time frame". This leads to the second major obstacle – time pressure.

According to the project leader, requirements/features were always given the highest priority in their sprint planning. In the situation where the time was barely sufficient to deliver requirements, "other processes tend to be completely ignored" and "TD management is very vulnerable as a candidate to be cast away". Such situations did occur in this study when the second sprint planning meeting was about to start. The project leader described the situation as a "very long" meeting with "a lot of complicated features". Therefore they did not spend time going through the TD list or discussing any TD items during the sprint planning meeting. Under time pressure, it was natural for the project team to consider TD management an optional process on top of, rather than incorporated into, their original sprint planning process. It also then makes sense that the team would revert back to their original process for prioritizing tasks because the original sprint planning process ensures the requirements gain highest priority and are always treated first even if they do not have enough time. When the project leader was asked about the impact of TD management on the subject project, he explained this tendency in another way: "we had not used the recorded values to make the decisions and we usually paid the debts when the demand was low".

Because we have realized that our TD management approach was not strictly followed by the project team, we framed some interviews questions for the project leader as to what changes to the management approach would have made it easier for the project team to adopt. The answer by the project leader brought up another major obstacle – lack of process integration. The project leader suggested that the integration of our approach with their project management tool would have helped them in using the approach.

Actually the sprint planning process that the team was already following was very similar to our TD management approach – walking through tasks, estimating effort, prioritizing them accordingly, deferring some low priority tasks if the required effort is more than what is available for the next sprint. Therefore, our approach could have been organically integrated into their sprint planning process although it was not presented that way to the team originally. So the project team treated TD management as an addition onto their original process. It became another, separate task, with a separate document (the TD list) to deal with. They started trying to fit in discussion of TD items only after they finished all the items in their backlog, during the sprint planning meetings. For example, during the first sprint planning meeting, the project team first spent 1.5 h discussing each task in the backlog and assigning priorities. Then they moved to TD management and spent another 35 min on discussion of updating the TD list,



which might have been saved, at least partially, if TD management had been well integrated into their sprint planning process. The same approach was taken during the second sprint planning meeting, but because there was not enough time left over, TD management was only briefly covered at the very end of the meeting and no other actions or decisions on TD were made. In an interview with the team leader after the meeting, he stressed that they didn't have enough time to follow our approach to manage TD before and during the meeting. This indicates that TD management was very much viewed as an add-on, not an integrated part of the sprint planning process.

In summary, there were three major obstacles in this study for the project team. The first obstacle, i.e. the difficulty of TD identification, together with the low perceived usefulness by the project team, hindered motivation to apply the proposed TD management approach and thus caused the actual TD management process to deviate. Time pressure was an external factor that resulted in the intended prioritization mechanism of our approach (i.e. principal and interest) to be abandoned under some circumstances. Lack of process integration, another obstacle, also aggravated the process deviation.

#### 5.4. Evaluation of validity

The actual implementation deviated from the original study design in that the proposed TD management approach was not followed in their sprint planning process. This is a threat to the internal validity of this study as there might have been other types of costs that were not captured due to the process deviation. In addition, the cost pattern we identified, i.e. high cost of updating TD list for the first time and insignificance of other types of costs for TD management, may not accurately represent the cost pattern for other projects that more fully used TD information in the sprint planning process. Therefore, the reliability of our findings related to Research Question 1 is weak.

Another threat to both the internal and construct validity came from the shifted focus of this study. The original study design was to explore the costs of explicit TD management. Therefore the study instruments, including the observation/documentation guidelines and the interview questionnaire, didn't focus on obstacles and process deviation until the second sprint that we tracked. This means some details about the obstacles and process deviation that happened in the early stage of this study may be missed or captured by indirect means. The threat is also reflected in the data volume and level of detail, as most of the information regarding obstacles and process deviations was collected in the later stage of the study. However, because it was collected during the critical phase, i.e. sprint planning, we do not believe this has a significant impact on the validity of this study, especially on the conclusion regarding the obstacles and how they contributed to the process deviations.

There is a threat to reliability in that the data collection techniques were not as robust as they could have been. This is primarily due to the remote location of the researchers during some phases of the study and the mix of languages used in conducting the research. The cultural difference between some of the researchers and the project team may have had an effect as well. These factors may have led to some misunderstandings or to missing some context factors that might have been relevant. However, we believe these effects are at least partially mitigated by the fact that both the second and third authors were on-site during most of the data collection (the third author was on-site for the entire study), and there was a mix of cultural backgrounds represented in the research team.

## 6. Conclusions

The original goal of this study was to uncover the costs of explicit TD management. In the process of implementation of this study, we observed that the project team encountered some obstacles and the actual process of TD management they followed was different from the one we proposed. This deviation provided us with the opportunity to investigate the obstacles to application of TD management in software projects and reveal the reasons behind the process deviation. In the end, we investigated three research questions, presented in Section 1. We discuss the findings of our study in light of these three research questions in this section, followed by a discussion of the implications of this work, and its limitations.

### 6.1. Discussion of findings

*Research Question 1: What are the costs of explicitly managing TD using the proposed approach?*

The costs we have identified largely fall in four categories: cost of TD identification, cost of updating the TD list, cost of using the TD list for sprint planning, and cost of communication. Among these costs, TD identification accounted for the majority of the total TD management cost. The cost to update the TD list for the first time was also high, but it dropped significantly in the subsequent sprints. Because TD identification is the first step of TD management, and constructing and updating a TD list immediately follows TD identification, the high cost incurred by these first two steps created a large, almost prohibitive, overhead for TD management, as illustrated in Fig. 5. Other costs of TD management are insignificant compared to this initial overhead. It should be noted though that the low effort of updating the TD list in Sprint 14 and 15 compared to Sprint 13 may result from the project leader's decision of not managing TD for sprint planning in the two release cycles. Therefore, the cost pattern may have been different, or even the cost may have not formed a pattern, if the TD management had been practiced during the two Sprints.

However, we tend to believe this cost pattern is real because we found a similar one (high initial overhead and subsequent minimal costs) in our previous study (Guo et al. 2016). The cost categories found in the two studies are not exactly the same, but a clear mapping can be found between the two. For example, one cost category in our previous study is "analysis and evaluation of TD", which subsumes the cost of updating TD list in the current study. Therefore, the findings from this study about initial overhead and the cost pattern are consistent with the findings of our previous study.

Because the actual TD management process deviated from our proposed approach in this study, the cost pattern we identified may be different from a case in which the proposed approach is strictly followed. But given the consistency of this finding with that of our previous study (Guo et al. 2016), and the fact that this finding is further elaborated and explained through interview data, we are confident in concluding that the application of TD management has a significant initial overhead. The overhead is largely composed of two types of costs – TD identification and evaluation. Therefore, software managers who intend to introduce TD management to their projects should be fully aware and prepared for the impact of this overhead, while any effort in facilitating TD identification and analysis could reduce the negative impact of this initial overhead.

*Research Question 2: What are the obstacles to explicit TD management?*

One of the major findings of this study is that there are significant obstacles to the application of explicit TD management in software projects. Among the obstacles, those associated with

## TD Management Costs

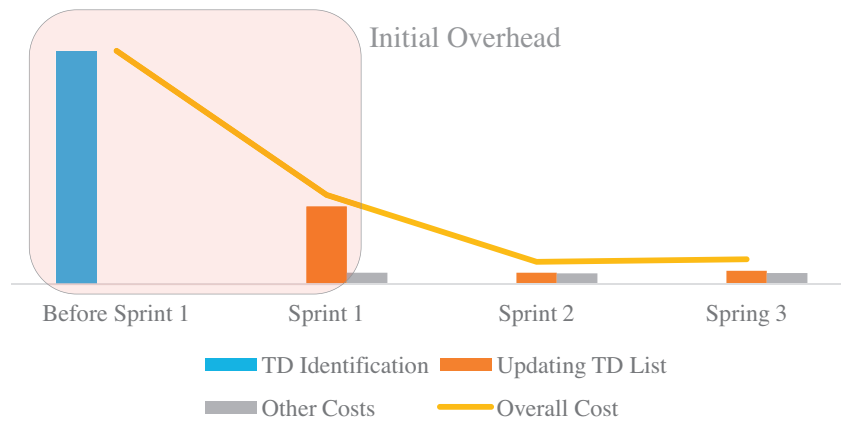


Fig. 5. Cost pattern of TD management.

TD identification, including the learning curve of understanding TD and the technical difficulty of TD identification, were major. Another obstacle was rooted in the fact that the project team had to implement a heavy load of requirements in an “unreasonably” short time frame. For software managers, a common strategy to deal with time constraints is to delay some non-critical tasks, which is usually the primary reason that TD is incurred. Ironically, in the studied case, the proposed TD management approach itself became a candidate to be abandoned. Therefore, time pressure was an obstacle to application of TD management in the project.

By looking deeper into the results, we also found that a lack of process integration was an obstacle, in that the project team regarded the TD management approach as an addition to their original sprint planning process rather than a part of it. In hindsight, there were clear ways that TD management could have been better integrated into the tools and processes used for sprint planning, as the two processes were not that different, but opportunities for integration were not obvious initially. As an add-on, it became easier for the project team to abandon TD management under time pressure, as discussed above. Therefore, lack of process integration became another obstacle.

*Research Question 3: What contributes to the deviation of the proposed TD management process?*

Due to the obstacles discussed above, the actual TD management process followed by the project team deviated from the one we proposed. The main deviation lay in the criteria used to decide if an instance of TD would be paid off in the next sprint. Rather than using TD principal and interest, in a cost-benefit analysis, to make this decision (as proposed in the TD management approach), the project team used a variety of other criteria, relying on their previous sprint planning process. In theory, any decision-making criteria could have been incorporated into the process of estimating principal and interest. For example, the availability of specific personnel is often considered in these decisions. If the personnel best suited to paying off a TD instance are not available, then this could be incorporated into that instance's estimate of principal, making the principal high enough that it would not be considered cost-effective to pay off. Similarly, if a TD instance is particularly important to a customer, that instance's interest would increase, reflecting the future cost (in terms of good customer relations) of not paying off the debt. However, incorporating such decision criteria into the notions of principal and interest is not intuitive, so the project team in this study preferred to use criteria other than principal and interest in their decision making. In our previous study (Guo et al. 2016), the project team also found it difficult to

incorporate other important decision criteria into the estimation of principal and interest, but they also found it possible to use principal and interest alongside the criteria they were used to using, in making decisions about paying off TD.

### 6.2. Implications

Our analysis of the obstacles we identified through this study, and the reasons behind them, allow us to reason about and recommend ways that the TD management process and strategy should be adapted. First, the initial overhead of TD management has to be reduced. Considering the learning curve and the effort of TD identification, an exhaustive search for TD items to construct an initial TD list may not be beneficial. Instead, the project could start with a small number of TD instances whose impact on the project is relatively obvious or serious enough to warrant immediate attention, for example, the most “painful” TD items in the project. It has been demonstrated that a careful analysis of just one TD item under certain conditions could pay off (Guo et al. 2011), while the results from this study show that managing a large TD list was not a cost-effective approach. Actually focusing on particular TD items rather than dealing with all TD issues at the same time not only reduces the management cost, but also helps the users overcome the learning curve and thus improve their expectations of the ease and usefulness of TD management. According to the Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh et al. 2003), perceived ease of use and usefulness are two determinants of user acceptance of a new technology. Therefore, increased expectations of usefulness of TD management on the part of the project team would eventually contribute to its acceptance positively.

Meanwhile, the difficulty surrounding TD identification calls for more automatic approaches. Most of the approaches and tools currently available tend to yield a comprehensive set of TD items, but they rarely provide accurate interpretation of the importance of individual items. Thus the users often face the situation that they have a large set of TD candidates, but barely know which ones are more important or critical in the context of their project. A strategy of narrowing the focus of initial TD identification is particularly important to large and mature software projects such as the one chosen in this study, because the size of the code base and the length of the maintenance phase provide more opportunities for TD of a larger variety to be incurred. Certainly it is not always true that a larger, more mature project will have more TD of more

types than a smaller and younger software project, but one can intuit that this is likely to be the case often.

One of the findings of this study, and to a slightly lesser extent our previous study (Guo et al. 2016), was that the criteria used for decision making about TD differed between the proposed TD management approach and the actual projects. In theory, principal and interest could incorporate notions of resource availability, customer preferences, and other decision criteria important to projects. But in reality, principal and interest were seen as representing only required effort. So during decision making, TD principal and interest needed to be combined (or, in the case of the current study, supplanted) with other factors to determine the priority of a debt item. This complicated the process of decision making, when the intent of the TD management approach was to simplify it. This experience leads us to recommend an approach to adopting explicit TD management that allows for a “transition” time, in which a project’s familiar decision making process and criteria are maintained. The notions of principal and interest are introduced, but initially represent simply required effort (e.g. to pay off the debt, in the case of principal). Gradually, ways to incorporate other decision criteria important to the project into the notions of principal and interest could be devised and evaluated. Eventually, guidelines could be developed to deal with any relevant decision criteria by folding it into the estimates of principal and interest, allowing a straightforward cost-benefit analysis to be used for prioritization of TD items. In this way the decision makers would be more likely to try TD management in their projects without worrying about its effect on their original decision making process. This would make it more likely to see the benefit of TD management earlier, which is another technology acceptance determinant (Venkatesh et al. 2003). This could eventually improve the acceptance of TD management in software practice.

In summary, the key take-aways of this study for practitioners beginning to manage the TD in their projects are:

- Narrow the focus of initial TD management efforts. Choose TD items that are obviously causing “pain” of some sort. This eases the upfront cost of developing a TD list, and also provides early successes.
- Integrate TD management into the existing project management process as much as possible, e.g. by incorporating TD items into the project backlog.
- Start with simple notions of principal and interest that focus on effort, and gradually incorporate other decision criteria by using them to adjust values for principal and interest.

From a research perspective, we also have some insights to offer those investigating TD phenomena in the future. The case we studied was chosen for convenience, as are the subjects of many (if not most) case studies in the software engineering literature. While we believe we have been able to glean interesting and valuable findings from this study, we were not able to address our original research question (1) as fully as we would have liked. We believe there were some aspects of the subject project that contributed to this inability, and that made the case non-optimal for this study. We also might have recognized some of these aspects earlier than we did, and so offer them as insights for future researchers. First, we underestimated the volatility of the business environment in which our studied project operated. Although we were aware that other projects in this organization (which we had previously considered for study) had been unexpectedly canceled, we were taken by surprise when the one we were studying was canceled. This cancellation cut short by one the number of sprints we had hoped to observe, and limited our ability to follow up with project personnel after the study was over. Second, while the project did not object to participating in our research, their participation did not stem from a recognized problem that they were seeking to solve.

Initial impressions of the usefulness of the TD concept were lukewarm from the start, which should have alerted us to potential problems with motivation that might have affected results later in the study. Initial difficulties with TD identification and the project’s steep learning curve might also have given us an early warning that TD management would not be a natural fit for this project. On the other hand, choosing a project with a clear and obvious need for exactly the TD management process we were proposing might have biased our results in the opposite direction. Third, the case was conducted at least partially remotely at all stages. The three authors and the studied project were never all co-located. We don’t believe the language barrier was significant (due to a sufficient level of bilingualism among both researchers and project personnel), but the geographical distance and the inability to communicate face to face much of the time made difficult concepts hard to convey.

For researchers, our results lend clear support for the continued need for better and more automated techniques for TD identification, particularly those that incorporate some notion of value or importance of the TD items identified.

### 6.3. Limitations

To overcome the limitations of our previous study of the same type (Guo et al. 2016), the subject project used for this study has a different profile. The software project described in this paper was developed in an organization belonging to a multinational company. It was a larger and more mature software project compared to the one used in the previous study. We expected that this study could complement the previous one and improve the generalizability of the conclusions drawn from them. However, the limitation of generalizability couldn’t be fully addressed by this single case. From the perspective of case selection, the project we used for this study was not the best choice as the project was not obviously feeling the “pain” of TD. The project personnel might have been more motivated, and thus offered less resistance, if the project had had some obvious TD issues and wanted very much a solution to the TD problem. This implies that our results, in particular the impact of the obstacles we encountered, might not apply in contexts where the need for TD management is recognized by all stakeholders.

So far all studies we have conducted on TD management, including this one, were hosted by commercial software companies from the private sector of the software industry. One area we haven’t covered is governmental systems, i.e. large contractors for government agencies, which differ from private software businesses in such aspects as project cost control, management style, and even enterprise culture. These aspects may influence TD and its management in their projects. Therefore, study of software projects in governmental systems could help us uncover more factors affecting costs or the cost pattern of TD management and thus improve the generalizability of the theory we are building for TD management. Similarly, we also propose to study non-Agile projects (all those we’ve studied have been Agile), as software development process could affect TD management and the related decision making as well.

The case used for this study was selected based on convenience and availability. Due to business reasons, the subject project was interrupted several times during the study, and was ultimately canceled right at the end of our study. As a result, we were not able to conduct a second-round follow-up interview with the project leader or other personnel. Thus, we were limited in the depth of insight we could elicit about the project’s experience with TD management. This raises the possibility that we are misunderstanding some of the data that we have collected. We have attempted to mitigate this risk through careful analysis, direct obser-

vation where possible, and peer checking within and outside our research team.

#### 6.4. Future work

We have focused, in this case study, on the process of managing TD in software projects. In contrast to much prior work on TD identification, we aim to shed light on the important question of what to do with TD information once it has been identified. Our analysis comes full circle, however, in that we have found that a major obstacle to successful adoption of explicit TD management is the difficulty of TD identification, in particular its high cost and technical difficulty. Thus better and more cost-effective ways of identifying TD are needed, and are an active area of research and practice. But better identification of TD will not have an impact on practice unless we know more about how TD information can be used in managing a software project, which is the subject of this study. We have identified several improvements to the basic TD management approach we originally proposed, based on a careful analysis of the case described here, as well as our previous study (Guo et al. 2016). More such cases, especially those in which new and innovative approaches to TD management are piloted, will continue to evolve our understanding of how to better manage the evolution of software systems.

#### Acknowledgments

We thank the members of the subject project team, and their management, who so generously offered their experience and wisdom to this research. The participation of Yuepu Guo in this work was supported by the US National Science Foundation (NSF), award #0916699. Carolyn Seaman was also supported by this grant, as well as by the Brazilian National Research Council (CNPq), project # 400849/2012-7. Fabio Silva's participation was supported by CNPq project #314523/2009-0.

#### Appendix A. Interview questionnaire I

##### Interview questions for the project leader on technical debt management activities in sprint 14

1. Date, time, length, and number of participants at sprint planning meeting.
2. Did you do anything with the TD list before the meeting?
3. Did you do anything with the TD list at the meeting?
4. If yes to either of the above:
  - a. What did you do? Who was involved?
  - b. Did you make any changes to the list?
  - c. Did you make any decisions about paying off or adding TD? How was the decision made? What were the major decision criteria?
    - i. Did you use/change any principal or interest estimates? How confident are you in these estimates?
  - d. How much time did it take? (broken down if possible by type)
  - e. Was anyone else involved?
  - f. How do you feel about TD management; was it useful?
5. If no to #2 and/or #3, please answer the following questions:
  - a. What are the obstacles to incorporating TD management into the sprint planning process?
  - b. When you do planning poker, what happens if the total amount of estimated effort is more than is available for the next sprint?

- c. Do you ever add things to the sprint plan that didn't come from SIDI, e.g. things you think are necessary to do but aren't part of some feature that they've requested? If so, do you have to justify it? How do you decide that it's important?

#### Appendix B. Interview questionnaire II

##### Interview questions for the project leader on technical debt management in the subject project

##### 1. Profile of the interviewee

- a. How long have you worked in software development?
- b. How long were you part of this product team?

##### 2. Original Management Practice

- a. Before we introduced our TD management approach to you, did you ever do anything or consider any ways to specifically manage TD? If so, could you briefly describe any ideas you had at that time about managing TD (even if you didn't call it TD at the time)? In other words, before you started learning about TD, did you have any ways, or consider any ways, to manage tasks aimed at improving maintainability, i.e. tasks other than fixing bugs and implementing new functionality?
- b. After the training of TD management, what's your expectation about the approach? Did you think it would work or did you have concerns?

##### 3. TD Management Approach

- a. What was the most important obstacle to incorporating TD management into your project management?
  - i. Besides <the first obstacle given>, what else presented an obstacle to incorporating TD management? (repeat as needed)
- b. Is the project a typical project in terms of size, development process, management style and resource availability? If not, what are the special characteristics of the project? Are they related to TD management?
- c. Was the training of the TD management helpful for understanding the TD concept and use the TD management approach? Was there any problem in the training?
- d. Was it easy or hard to identify TD items from your project? And why?
- e. In what ways could we have changed the management approach that would have made it easier to be incorporated into your project management than it actually was?
- f. Based on your experience of managing TD using the proposed approach, what's your general impression about it?
- g. What's the most difficult/hard/time-consuming part of the TD management approach, e.g., understanding the changes of TD items, reviewing and updating TD list, re-estimating TD items, etc.?
- h. Do you think it is necessary/worthwhile/feasible to numerically estimate the principal and interest of TD? And why?
  - i. What do you think were the impacts of managing TD on the project, if any?
  - j. Has your experience with managing TD on the project had any impact on your sprint planning on other projects? If so, what was it?

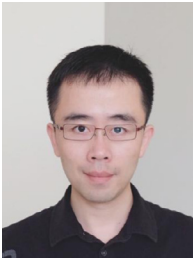
##### 4. Wrap-up

- a. Is there anything you want to add about the management process or the TD approach?



## References

- Bachmann F., Nord R.L., Ozkaya I. (2012) Architectural tactics to support rapid and agile stability. *CrossTalk*, May/June-2012:20–25.
- Bohnet, J., Döllner, J., 2011. Monitoring code quality and development activity by software maps. In: *Proceedings of the 2nd Workshop On Managing Technical Debt (MTD '11)*, pp. 9–16.
- Brondum, J., Zhu, L., 2012. Visualizing Architectural Dependencies. In: *Proceedings of the 3rd Workshop On Managing Technical Debt (MTD '12)*, pp. 7–14.
- Brown, N., Nord, R.L., Ozkaya, I., Pais, M., 2011. Analysis and management of architectural dependencies in iterative release planning. In: *Proceedings of the 9th Working IEEE/IFIP Conference On Software Architecture (WICSA)*, pp. 103–112.
- Cunningham, W., 1992. The wycash portfolio management system. In: *Addendum to the Proceedings On Object-Oriented Programming Systems, Languages, And Applications*, pp. 29–30.
- Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* 13, 319–339.
- Emden, E.V., Moonen, L., 2002. Java quality assurance by detecting code smells. In: *Proceedings of the 9th Working Conference On Reverse Engineering*, pp. 97–106.
- Fowler, M., (2009) Technical debt quadrant. Web. <http://martinfowler.com/bliki/TechnicalDebtQuadrant.html> Accessed 1 Dec 2013.
- Guo, Y., Seaman, C., Gomes, R., Cavalcanti, A., Tonin, G., Da Silva, F.Q., Santos, A.L., Siebra, C., 2011. Tracking technical debt — an exploratory case study. In: *The 27th IEEE International Conference on Software Maintenance*, pp. 528–531.
- Guo, Y., Seaman, C., Zazworka, N., Shull, F., 2010. Domain-specific tailoring of code smells: an empirical study. In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, 2, pp. 167–170.
- Guo, Y., Spinola, R.O., Seaman, C., 2016. Exploring the costs of technical debt management – a case study. *Empir. Softw. Eng.* 21 (1), 159–182.
- Kaindl, H., Brinkkemper, S., Bubenko Jr., J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., do Prado Leite, J.C.S., Mead, N.R., Mylopoulos, J., Siddiqi, J., 2002. Requirements engineering and technology transfer: obstacles, incentives and improvement agenda. *Requirements Eng.* 7 (3), 113–123.
- Lester A., (2008) Get out of technical debt. Web. <http://petdance.com/per/technical-debt/>. Accessed 1 May 2008.
- Letouzey, J.-L., 2012. The SQALE method for evaluating technical debt. In: *Proceedings of the 3rd Workshop On Managing Technical Debt (MTD'12)*, pp. 31–36.
- Letouzey, J.-L., Ilkiewicz, M., 2012. Managing technical debt with the SQALE method. *IEEE Softw.* 29 (6), 44–51.
- Li, Z., Liang, P., Avgeriou, P., 2015. Architectural technical debt identification based on architecture decisions and change scenarios. In: *Proceedings of the 12th Working IEEE/IFIP Conference On Software Architecture (WICSA)*, pp. 65–74.
- Marinescu, R., 2012. Assessing technical debt by identifying design flaws in software systems. *IBM J. Res. Dev.* 56 (5), 1–13.
- Martini, A., Bosch, J., Chaudron, M., 2015. Investigating architectural technical debt accumulation and refactoring over time: a multiple-case study. *Inf. Softw. Technol.* 67, 237–253.
- McConnell, S., (2007) 10x software development. Web. <http://forums.construx.com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx>. Accessed 1 May 2008.
- Nord, R.L., Ozkaya, I., Kruchten, P., Gonzalez-Rojas, M., 2012. In search of a metric for managing architectural technical debt. In: *Proceedings of the Joint 10th Working IEEE/IFIP Conference On Software Architecture (WICSA) and the 6th European Conference On Software Architecture (ECSA)*, pp. 91–100.
- Nugroho, A., Visser, J., Kuipers, T., 2011. An empirical model of technical debt and interest. In: *Proceedings of the 2nd Workshop On Managing Technical Debt (MTD '11)*, pp. 1–8.
- Rothman, J., (2006) An incremental technique to pay off testing technical debt. Web. <http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=COL&ObjectId=11011&tth=DYN&tt=siteemail&iDyn=2>. Accessed 1 May 2008.
- Schmid, K., 2013. A formal approach to technical debt decision making. In: *Proceedings of the 9th International ACM Sigsoft Conference On Quality Of Software Architectures*, pp. 153–162.
- Seaman, C., Guo, Y., 2011. Measuring and monitoring technical debt. *Adv. Comput.* 82, 25–46.
- Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D., 2003. User Acceptance of information technology: toward a unified view. *MIS Q.* 27 (3), 425–478.
- Wang, P., Yang, J., Tan, L., Kroeger, R., Morgenthaler, J.D., 2013. Generating precise dependencies for large software. In: *Proceedings of the 4th Workshop On Managing Technical Debt (MTD '13)*, pp. 47–50.
- Zazworka, N., Seaman, C., Shull, F., 2011. Prioritizing design debt investment opportunities. In: *Proceedings of the 2nd Workshop On Managing Technical Debt (MTD '11)*, pp. 39–42.



**Yuepu Guo** is a PhD candidate in the Department of Information Systems at University of Maryland Baltimore County (UMBC). He received the BS degree in industrial economics from Tianjin University, Tianjin, P. R. China, in 1996, the MS degree in information systems from University of Maryland Baltimore County, Baltimore, USA, in 2009. His research interests include software process, maintenance and evolution. In particular, he is interested in and currently investigating issues of technical debt in software lifecycle, part of which forms his doctoral dissertation. He can be reached at [yuepu.guo@umbc.edu](mailto:yuepu.guo@umbc.edu)



**Carolyn Seaman** is an Associate Professor of Information Systems at the University of Maryland Baltimore County (UMBC). She is also a Research Fellow at the Fraunhofer Center for Experimental Software Engineering, Maryland. She holds a PhD in Computer Science from the University of Maryland, College Park, a MS in Information and Computer Science from Georgia Tech, and a BA in Computer Science and Mathematics from the College of Wooster (Ohio). Contact her at [cseaman@umbc.edu](mailto:cseaman@umbc.edu).



**Fabio Q.B. da Silva** is coordinator of the Cin/Samsung Project Unit and Associate Professor at UFPE. He received his PhD in Computer Science at the University of Edinburgh His main research topics are TD and Human Factors in Software Engineering. Contact him at [fabio@cin.ufpe.br](mailto:fabio@cin.ufpe.br).