A Systematic Literature Review on Intertemporal Choice in Software Engineering – Protocol and Results

Christoph Becker, Dawn Walker, and Curtis McCord

Abstract—When making choices in software projects, engineers and other stakeholders engage in decision making that involves uncertain future outcomes. Research in psychology, behavioral economics and neuroscience has questioned many of the classical assumptions of how such decisions are made.

This literature review aims to characterize the assumptions that underpin the study of these decisions in Software Engineering. We identify empirical research on this subject and analyze how the role of time has been characterized in the study of decision making in SE.

The literature review aims to support the development of descriptive frameworks for empirical studies of intertemporal decision making in practice.

Index Terms—Software Engineering, Behavioral Software Engineering, Intertemporal Choice, Technical Debt, Sustainability Debt, Trade-off decisions, Decision Theory, Sustainability

I. INTRODUCTION

Complex software-intensive systems play critical roles in our societies; their ongoing development, innovation, and maintenance is intertwined with our everyday social and economic activities. As recognition of the key role software technology can play in society’s sustainability grows, the need for a paradigm shift in the mindset of the software industry has become clear. Sustainability is often defined within the domain of “sustainable development”, which “meets the needs of the present generation without compromising the ability of future generations to meet their own needs” [1]. At its core, sustainability is the capacity to endure, but sustainability of social systems is different than technical or natural systems. Originally equated with environmental concerns, it is now clear that sustainability requires equal consideration of five dimensions: environmental, societal, individual, economic, and technical [2].

Within and across these concerns, software engineering (SE) decisions are made about system scope, goals and objectives, features, functions, architectural designs, and many other areas throughout the development lifecycle. The effects of these choices are often delayed, and many critical decisions involve trade-offs between outcomes at different points in time. In such cases, longer-term consequences are not always sufficiently considered [3], [4].

Research in psychology and behavioral economics calls choices that involve trade-offs across time “intertemporal choices,” defining them as “decisions involving tradeoffs among costs and benefits occurring at different times” [5]. Researchers have developed a number of theories of such choices [5], [6] and have demonstrated that straightforward assumptions about how decision makers evaluate and discount the future are often misguided and wrong [5], [6].

Herein, an important distinction is made between normative and descriptive decision theories. Normative theories focus on the identification of the best decision, and model an ideal decision maker. Normative models of how decisions are made in SE commonly assume a rational agent (with reasonable cognitive boundaries) choosing between a set of options according to a value function.

From choosing a software development methodology to evaluating release planning, prioritizing requirements and choosing between architectural design options, SE literature commonly assumes that decision making operates in a predictable, rational way. For example, one author writes “In most problems, to make a decision, a situation is assessed against a set of characteristics or attributes, also called criteria. Decision making based on various criteria is supported by multi-criteria methodologies” [8]. The assumption is that a team of competent engineers evaluates the options to the best of their knowledge, and they choose the option with the highest expected value. Much of their discussion in theory and practice focuses on how to best estimate that value. The frameworks of Value Based Software Engineering aim to base SE decisions more explicitly on an understanding of value [7]. Most commonly, this value is expressed in economic terms, and the incommensurability of multiple aspects of value is often addressed through application of utility functions [8].

The theory of expected utility stems from game theory [9] and was developed from principles, not empirical study. By contrast, descriptive theories aim to characterize the behavior of actual decision making. As Tversky and Kahneman write, “The modern theory of decision making under risk emerged from a logical analysis of games of chance rather than from a psychological analysis of risk and value. The theory was conceived as a normative model of an idealized decision maker, not as a description of the behavior of real people... the logic of choice does not provide an adequate foundation for a descriptive theory of decision making” [10].

Criticism of the prevailing normative decision theories has come from numerous perspectives, and various alternative
conceptions have been proposed. For example, well-known experiments have shown that people do not discount the future linearly [5] and that risk aversion is higher for gains than for losses [10]. More substantively, Tversky and Kahneman showed that some of the foundational axioms of normative decision theory, and in particular expected utility theory, are inconsistent with observable behavior. More radically, Klein's study of expert decision making showed convincingly that experienced decision makers do not actually weigh a set of alternatives against criteria to maximize expected utility when making critical choices [11]. It is this divergence between prevalent normative models and observed behavior that motivated this review. This corresponds to the recent emergence of Behavioral Software Engineering, a field that aims to draw in behavioural frameworks and concepts for a better understanding of software engineering [12], [13], [14].

In SE, choices that are expedient in the short-term but create unwanted longer-term consequences have been conceptualized most prominently as ‘technical debt’, which focuses on engineering choices that create hidden costs. The metaphor of debt aims to make these hidden costs visible and manageable. Interpreted more broadly, the notion of ‘sustainability debt’ expands the metaphor to direct and indirect effects across all dimensions of sustainability [15].

A. Objective

This review is motivated by the need to better understand how and why software practitioners incur sustainability debt in practice. In order to develop a descriptive framework for intertemporal choices in SE, we review the literature to identify whether the intersection of these concepts has been acknowledged and addressed; describe which perspectives and assumptions about decision makers underpin existing research; and analyze how the role of time has been characterized in the study of decision making in SE.

Because of our interest in distinguishing normative models theorizing about decision-making in SE from descriptive, empirical accounts of how trade-off decisions relating to time are made in software design projects, we will first map empirical and other types of research of decision making, and then analyze empirical research in detail in order to understand the assumptions of decision making models that underpin this research.

B. Contribution

We aim to reveal how trade-off choices have been conceptualized within SE so far, identify gaps in how decision making is reviewed and investigated, and map how SE literature approaches making trade-offs over time.

II. LITERATURE REVIEW STUDY DESIGN

A. Research Questions

We characterize perspectives on decision making within SE research through the following questions:

RQ1 Which empirical research in SE has studied trade-off decisions involving time?

Fig. 1. Overlap between SE and Intertemporal Choice SCOPUS query

TABLE II

| OVERVIEW OF SCOPUS SEARCH RESULTS FOR INTERTEMPORAL CHOICE |
| PRELIMINARY SEARCH |
| --- |
| Total number of search results | 0 |

RQ2 Which dimensions are considered in these studies?
RQ3 How has the role of time been conceptualized in these studies?
RQ4 Which assumptions on decision making underpin these studies?

While we are interested in the assumptions on decision making that underpin the perspective of the non-empirical studies, we focus our in-depth analysis on empirical work due to time restrictions.

B. Roles and Responsibilities

The roles and responsibilities for this project are defined in Table 1. We have one principal researcher, Christoph Becker, and two supporting researchers, Curtis McCord and Dawn Walker. External reviews were conducted by Stefanie Betz and Ruzanna Chitchyan.

C. Search Strategy

In order to produce a systematic overview of this area, the overall search process for this literature review is based on guidelines established by Kitchenham [16].

1) Information Sources: We performed automated searches on the following indexing systems and digital libraries: Scopus, IEEE Xplore, and ACM Digital Library.

2) Preliminary search: The term ‘intertemporal choice’ has come to describe precisely our area of interest. At an early stage, we conducted searches to identify whether there has been explicit attention to this concept in the literature.

“intertemporal choice”
AND “software engineering”

Fig. 1 and Table II show the resulting search numbers for SCOPUS. While the exact numbers differ for the other databases, the trend is mirrored and the intersection remained empty for all searches.

The search revealed that intertemporal choice is not explicitly treated in the literature at all, and the phrasing was not present in any papers. This does not necessarily indicate
that SE does not deal with intertemporal choices, but the absence of explicit mention of the term “intertemporal choice” suggests that the concepts arising from the field of behavioral economics have not been congruently linked to SE, i.e. that no direct conceptual mapping has been established between the two disciplines yet.

3) Preliminary concept review: Before conducting further searches, we aimed to establish a candidate set of concepts that would scaffold our understanding of decision making vocabulary. To do so, we reviewed textbooks and standards in Software Engineering [17], Value Based Software Engineering [18], Decision Analysis, Behavioural Economics [19] and Management Theory [19] to compose a working vocabulary of terms related to intertemporal choice. From these texts we developed a series of prototypical concept maps that decomposed key components of decision making into potential search terms. Terms such as “cost”, “value”, “benefit”, “risk”, “decision-making”, for example, were widely used across disciplines, and helped to structure our understanding of decisions and provide terminology for coding and analysis later on.

4) Search String: The goal for the search string was to capture results that dealt with intertemporal decision-making in SE, to examine how SE projects saw time as a factor in their decision processes, how they make decisions about the future of their projects, and how they might weigh future and present goods against each other. We included the clause “software engineering” to limit the disciplinary scope of our research—other disciplinary scopes such as “requirements engineering” could conceivably lead to different perspectives.

To capture the temporal aspect of decision making, we settled on the general term of “time”, with the intention of using more specific coding during analysis. Preliminary search queries (See Appendix A-B) were more complex and used more discipline-specific jargon (“life cycle”, “endurance”) whose specificity would occlude relevant papers that could be captured by a more general query.

While these searches included relevant results that connected to the concepts that emerged from initial review, the results were mixed and widely spread across disciplines. It became clear that introducing divergent and specific terms from multiple disciplines would increase the amount of papers captured, but not necessarily make the literature review more effective or representative. This, and the possible bias introduced through these more complex queries, led us to choose a simplified more generic query string and move some of the detailed aspects of intertemporal choice to the coding and analysis stages.

The same reasoning process governed our decisions on the second clause of our query; we were interested in papers discussing trade-offs, but recognized that while the term is widely used, it might not be used by all authors describing these types of decisions. What we really wanted to capture through coding was choices that required parties to weigh decision dimensions against each other.

Search queries were piloted twice (See Appendix A for pretest queries) prior to establishing the final search string: time AND “decision making” AND “software engineering”

5) Ancillary Search: Using the same search strategy, one ancillary search was performed as part of the literature review: “technical debt”

The concept of Technical Debt (TD) is prominent in software engineering and closely related to the dimensions of our main query. Technical debt can be defined as: “a design or construction approach that is expedient in the short term but that creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time)” (Ernst [20], borrowing from McConnell). In this framing, TD always includes an explicitly temporal dimension, built into the concept of debt. Decisions that are made about TD would presumably include a temporal dimension and the commensuration of future and present goods. As such, the literature on TD could be complementary to other areas of intertemporal choice and shed light on specific assumptions.

The results of the ancillary query were documented, but the only analysis performed within this review was an identification of the overlap with the primary search, as described further below (see Section III). The resulting corpus of publi-
D. Selection Criteria

1) Inclusion Criteria: We established the following criteria to identify relevant publications that would answer research questions:

- **Publication Year**: All years were included.
- **Publication Type**: We included peer-reviewed papers published in journals, conference proceedings, and workshop proceedings.
- **Content**: The paper had to contain a discussion of decision-making in software engineering projects.
- **Coverage**: The paper had to cover development of a software system rather than only hardware.

2) Exclusion Criteria:

- **Publication Language**: We excluded papers in languages other than English.
- **Publication Quality**: We excluded papers retracted by the publisher.
- **Publication Type**: We excluded non-paper results including: posters, abstract-only submissions, book reviews, books, entire volumes of proceedings, panels, presentations, tutorials, opinion pieces.
- **Technical**: We excluded papers where the PDF was unavailable (behind a paywall or not locatable).

E. Selection Procedures

After downloading, removing duplicates, and applying our exclusion criteria, the remaining papers were screened for relevancy using the following procedure:

1) The secondary researchers voted on relevance: They read identified paper titles and abstracts in order to decide on inclusion using the criteria above. A yes or no decision (“Y/N”) was assigned as well as a certainty value from 1-3 (where 3=certain).

2) Voters reviewed 10 of the 307 papers as a pilot, and then conducted a larger pilot of 49 papers including the original 10, and discussed the results together.

3) Following this quality assurance step, the remaining 258 papers were split and reviewed by one voter each, following the voting process established above.

4) In cases where papers were reviewed by more than one voter, disagreements were resolved through discussion and consensus.

5) All decisions on papers reviewed by only one voter were compiled. Those with a certainty value below 2 (191) were reviewed and discussed by both voters, and a randomly selected sample of 65 was evaluated for consistency. In case of remaining doubt, the papers were included.

6) 155 papers marked for further coding were looked over by an internal reviewer to determine whether inclusion and exclusion criteria were appropriate before analysis. Because of the rule to include papers in case of doubts, the focus was on verifying included papers at this stage.

F. Analysis

Papers from the relevancy review were analyzed in order to address the research questions established above. Application of a checklist as well as review ensured the quality of analysis and coding. Researchers extracted data using a form to capture fields relevant to our research questions.

1) Assessment: In order to ensure quality of analysis and findings, researchers conducted multiple internal reviews throughout many stages of the Literature Review: protocol, relevancy voting procedure, relevancy voting results, technical report. Additional external review of the technical report led to suggestions for improvement.

2) Data Extraction: Researchers classified studies according to the type or domain of the decision-making studied, the methods of investigation and research, whether there was a trade-off decision, and if so, the dimensions of the trade-off. Free annotation was also used to capture additional information the coders deemed relevant.

A form was also designed to capture these fields as well as metadata from the studies, including author, title, year of publication, DOI, and unique document key (generated from author, title, publication).

Coding Field 1: Scope of the Decision

- **(pm)** Project Management
- **(dev)** Software Requirements, Design, Architecture, Development
- **(mait)** Software Maintenance
- **(other)** Including business and business strategy

Within SE, requirements decisions, design, architecture, and development cover a wide range of tasks, fields, etc. We intentionally grouped these together to cover all engineering decisions as one, in part because these decisions often span multiple areas.

Coding Field 2: Research Methodology

- **(emp)** Empirical methods were used and the object of empirical study was a decision
- **(emp comp)** Empirical methods were used and the object of empirical study was NOT a decision
- **(lit)** The paper was (exclusively) a literature review or a systematic mapping study
- **(other)** The research was not empirical, i.e. it was theoretical or attempted to develop a model

Coding Field 3: Does the article discuss trade-off decisions?

- **(Y)** Yes
- **(N)** No

Coding Field 4: Dimensions of Trade-Off

- **(cost)** Cost (Often in monetary terms)
- **(func)** Functionality
- **(mait)** Software Maintainability
- **(qual)** Software Quality
- **(risk)** Risk
- **(time)** Time (Includes scheduling, delivery and release)
- **(value)** Value (As in terms of monetary value, “business value” or in some cases, in terms of benefit)
- **(other)** See Appendix
TABLE III
OVERVIEW OF SEARCH RESULTS FOR MAIN SEARCH

| Description                                                                 | Number   |
|-----------------------------------------------------------------------------|----------|
| Total Number of Search Results                                              | 889      |
| Total number of results after duplicate removal and exclusion criteria applied | 652      |
| Number selected after preliminary relevancy review                         | 307      |
| Number selected after voting                                                | 155      |

3) Analysis of Extracted Data: The secondary researchers extracted data and analyzed the results included below. From this they synthesized findings on the current research. Feedback was provided through an internal review by the principal researcher. In order to analyze the extracted data, the researchers:

- derived statistics of coded categories for mapping extracted data
- mapped out areas of existing work
- created visualizations with groups of dimensions

III. RESULTS

Search result statistics are provided in Table III. First the search results from the indexing systems and digital libraries were compiled, then results were de-duplicated and exclusion criteria applied. From those 652 papers, and initial assessment to determine whether they were relevant led to 307 papers selected. Based on voting and discussion to reach consensus, that number was reduced to 155 for final coding.

Statistics of the ancillary “Technical Debt” query are summarized in Table IV. As expected, there was some overlap between the papers returned in our Technical Debt query and those returned in our main query. As technical debt has become a more prominent term in software engineering discourse, it also becomes a phenomenon which can be analyzed and accounted for. In this way it becomes manageable:

TABLE IV
OVERVIEW OF SEARCH RESULTS FOR TECHNICAL DEBT ANCILLARY SEARCH

| Description                                                                 | Number   |
|-----------------------------------------------------------------------------|----------|
| Total number of search results                                              | 620      |
| Total number of results after duplicate removal and exclusion criteria applied | 246      |

the object of decision-making. For these reasons we were not surprised to find several recent papers that focus on decisions about monitoring, reporting and managing technical debt. Before relevancy voting, 7 papers were in both the technical debt and main query corpus. After relevancy voting, the intersection of the two corpuses was 2 papers, as illustrated in Fig. 2:

- Martini and Bosch, 2016, *An Empirically Developed Method to Aid Decisions on Architectural Technical Debt Refactoring: AnaConDebt* [A3]
- Oliveira, Goldman, and Santos, 2015, *Managing Technical Debt in Software Projects Using Scrum: An Action Research* [A11]

IV. FINDINGS

To explore assumptions that underpin the existing empirical work on trade-off decisions in SE, we first extracted statistics from the coded categories of each paper in order to map extracted data. Subsequently, areas of existing empirical work were further identified, visualized, and data on the groups of dimensions was collated. From this, final analysis was performed in depth on the subset of empirical papers that discussed decision making in SE. We will discuss the main research questions in separate sections below.

A. Which empirical research in SE has studied trade-off decisions involving time?

As described above, the relevant publications were coded for an empirical focus on studying decisions, and for including in particular decisions involving ‘trade-offs’.

We found 93/155 papers had some degree of empirical component, and 88/155 discussed trade-offs in such capacities. The resulting Venn diagram shown in Fig. 3 shows a quite
TABLE V
ARRANGEMENT OF SELECTED PAPERS BY TYPE (SEE APPENDIX C)

| Segment       | Count | Description                                                                 |
|---------------|-------|-----------------------------------------------------------------------------|
| [A1-13]       | 13    | Research has an Empirical Component which Studies Trade-Offs or Like Decisions |
| [B1-46]       | 46    | Research has an Empirical Component and Discusses Trade-Offs or Like Decisions |
| [C1-34]       | 34    | Research has an Empirical Component and Does Not Discuss Trade-Offs or Like Decisions |
| [D1-29]       | 29    | Research Does Not have an Empirical Component and Discusses Trade-Offs or Like Decisions |
| [E1-33]       | 33    | Research Does Not have Empirical Component and Does Not Discuss Trade-Offs or Like Decisions |

Fig. 4. Number of studies with \( n \) trade-off dimensions

even distribution across the emerging subsegments, but indicates that only 13 studies were identified that explicitly used empirical methods to study trade-off decisions where time was a relevant element. This set represents papers that attempt to examine decision-making in software engineering in real or experimental situations.

Table VII summarizes key characteristics of the 13 identified papers, including the research method(s) and citation counts. The most prominent method is case study research.

We will focus later on this set of 13 papers in detail.

B. Which dimensions are considered in these studies?

Of the 155 papers coded, 88 identified at least one dimension of trade-off. The majority of those, 54, are choices within one or two dimensions, for example within differing stakeholder goals or between costs and time.

The most discussed single trade-off dimension is cost (39), the next highest mention is time (38), then quality (33), which includes nonfunctional requirements discussed as a group or specifically as “usability,” “security,” etc... The “other” category (38) had a diversity of dimensions seen in Table VIII those that occurred more than 5 times are indicated with an asterisk. In a few cases we categorized terms as the same when the language had some variation (e.g. value, business value, and business benefits into “value”). A record of these decisions was not recorded.

Of the 13 papers that empirically discussed a trade-off decision, the emphasis within dimensions was on cost (10), then time (7). Table IX shows the number of papers in which each dimension occurs within this set.

Further analysis of these dimensions could identify which sets of dimensions frequently co-occur. The data set has been prepared to support this analysis.

C. How has the role of time been conceptualized in these studies?

Time is the most popular dimension across all papers, 38 papers with time as a dimension; this is unsurprising considering the search term. Within this, time is addressed in the various mapping groups: empirical (7), empirical component (19), non-empirical (12), and literature review (0).

However, the role of time is of course not always intertemporal. Time surfaces

- As the object of effort estimation: How much time will each of these options take?
- As a factor in project management: How much time is available to the team?

TABLE VII
COUNT OF TRADE-OFF DIMENSIONS IN ALL STUDIES

| Dimension | Count |
|-----------|-------|
| cost      | 39    |
| time      | 38    |
| other     | 38    |
| quality   | 33    |
| functionality | 16 |
| risk      | 15    |
| value     | 13    |
| maintenance | 5    |

TABLE VIII
“OTHER” TRADE-OFF DIMENSIONS

| Dimension            | Count |
|----------------------|-------|
| benefit*             |       |
| competition          |       |
| complexity           |       |
| labour*              |       |
| methodology*         |       |
| opportunities        |       |
| return on investment |       |
| goals*               |       |
| technical debt       |       |
| vendor               |       |
| other                |       |

*Those with an asterisk appeared more than 5 times

TABLE IX
COUNT OF TRADE-OFF DIMENSIONS IN 13 EMPirical STUDIES

| Dimension        | Count |
|------------------|-------|
| cost             | 10    |
| time             | 7     |
| quality          | 6     |
| value            | 4     |
| other (benefit)  | 3     |
| other (goals)    | 3     |
| functionality    | 1     |
| other            | 1     |
| other (competition) | 1   |
| other (methodology) | 1   |
As an attribute of decision making - how much time does it take to make a choice?
As a factor in scheduling, finally, the closest to intertemporal choice: Should we release now or later?

In order to characterize the role of time in the 13 empirical studies, the primary researcher performed a detailed analysis of the 13 papers that were coded as empirical studies of trade-off decisions involving time. Reading through the papers, the researcher performed iterative qualitative coding. He identified all mentioning of ‘time’, reviewed each of their contexts, and iteratively developed a set of codes that described each new occurrence while continuing to apply to the existing occurrences. The resulting set of categories is summarized in Table X. Time in this set of papers is discussed most importantly as:

- A constrained resource in software project,
- The time it takes to apply a method (e.g. designed by the researcher) in a project, and often a measure of that time,
- An axis of discrete units ‘of time’ over which a sequence of events take place,
- The time to market or the time to delivery, and
- An axis of change on which to pick suitable moments for action.

Additionally, a number of unique attributes and aspects surfaced once that were interpreted as tangential, since the concept of time was not central to the focus or nature of decision making. For example, this included a case that discussed technical consideration of real-time systems in project decisions or a discussion of how fixed-time release cycles provided consistent structure and rhythm to an organization’s processes.

It is the last conception listed above, the axis of change, where intertemporal decisions arise explicitly. In this set, they arose in particular in two specific forms, each represented by
two distinct papers:

1) **Technical Debt** management raises questions around when to repay, and papers discussed how these decisions are being made;

2) **Release planning** raises questions of timing and of which **requirements to prioritize** and include for a given release.

However, in neither of these cases was explicit attention given to behavioral insights, or into how decision makers arrive at their choices.

**D. Which assumptions on decision making underpin these studies?**

The predominant model of decision making in the relevant papers, so dominant that it is normally not made explicit, is a normative decision making model that builds on a Taylorist perspective on management, focused on efficiency and effectiveness, measured in the most scientifically accurate manner possible. Decision making assumes the presence and validity of normative theories of decision analysis in which clearly defined options are weighed against stated criteria to determine the best choice. The actual choosing is then often presumed to be unproblematic. Rational choice is the standard model, sometimes with explicit awareness of its limitations, often articulated in the frame of, or consistent with, bounded rationality. Yet, awareness is also present that “research has proven that humans make trade-off analyses continuously– if not on the basis of objective measurements then on intuition.” (Van Solingen [A13] pointing to Beach’s Image Theory ). One paper explicitly proposes to contrast ‘rational’ decision making with ‘naturalistic’ frameworks and categorizes frameworks including Simon’s Bounded Rationality, Prospect Theory, Image Theory, and other models [A1]. The paper itself does not conduct empirical work, but a subsequent paper of the same authors does [21].

In this paper, Zannier and Maurer conduct multiple interviews to develop an understanding of two modes of decision making characterized as rational and naturalistic [21]. The findings suggest a distinction between problem structuring and problem solving, and the authors conclude that both modes are relevant and one of them is typically the dominant approach. When the focus of decision making was on structuring, as was most commonly the case in the software design activities studied, naturalistic decision making modes dominated. Where the focus was on problem solving, rational modes dominated. In each modes, aspects of the other were present as well. The authors add that “software designers often use satisficing and singular evaluation in trying different approaches to design” – where one option is checked for plausibility rather than being evaluated against other options, as described by naturalistic decision making frameworks [11].

However, the normative, rational decision making model also dominates the empirical papers that addressed the notion of trade-offs across time more explicitly.

In Lindgren’s study of release planning, intertemporal considerations are foregrounded explicitly in discussions of short- and long-term planning, and an explicit connection is made to the need to repay technical debt [A2]. However, while the paper highlights the need for longer-term perspectives and reports on empirical work, it focuses on larger questions and leaves open how precisely the decision makers acted in these decisions.

In Martini’s study of technical debt management, time is similarly prominent: “the TD theoretical framework instantiates a relationship between the cost and the impact of a single sub-optimal solution over time. In particular, the metaphor stresses the short-term gain given by a sub-optimal solution against the long-term one considered optimal” [A3]. Decisions have to be taken at the right time and have to anticipate uncertain future outcomes. The assumptions that are surfaced about the decision makers suggest that in the presence of perfect information, they would take the correct, optimal decision, surfacing assumptions of rational choice, contingent upon and bounded by the availability of information.

Racheva’s study examines requirements prioritization in an agile environment at inter-iteration time – the moment “when requirements are re-prioritized in the face of project
uncertainties”. At that stage, trade-offs consist in choosing what to do now and what not to do at the next iteration, a decision taken between iterations when requirements are re-prioritized A4. Over time, more information will be delivered and less of the limited resource of time will be available to act on it: “The client can wait to the last responsible moment …to make his decision… The term ‘responsible’ means that the client needs to understand the last point of time to make a decision without affecting the delivery of the project” A4. The framework that is introduced aims to provide a conceptual frame for uncertainty over time by introducing Real Options Analysis. Given the agile focus, it is unsurprising that the research focuses on grounding proposals on empirical insights and being responsive to the actual behavior of practitioners. The underpinning assumptions are implicit, but build clearly on ideas of bounded rationality.

Finally, Wohlin’s paper on release planning aims to provide guidance for which types of criteria practitioners should consider when conducting requirements prioritization for release planning. However, the actual decision making is not discussed A5.

V. CONCLUSIONS AND OUTLOOK

A. Discussion

Trade-off decisions in software engineering have been studied and modeled for a long time. However, the role of time in most studies focuses on time as a limited resource and a ticking clock. However, some work, such as technical debt research, has foregrounded the attention to trade-off decisions across time.

In general, it is difficult to find the most relevant work on such a subtle topic as ‘decision making involving trade-off decisions across time’, because the terminology that can be used to describe it is not stabilized and thus the terms are used in ambiguous and varied ways. This means that we cannot assume we have covered the body of literature that in fact discusses these questions comprehensively. However, because the main goal is to understand common assumptions and norms, a comprehensive identification of all works having studied these aspects in depth is not the primary criterion.

A number of studies have suggested that normative models are inadequate in explaining how people actually take decisions 5. Behavioral perspectives and empirical research are needed to provide new and deeper understanding on the practice of software engineering 13. This suggests that more descriptive research is needed to provide a bottom-up empirically grounded description of decision making. We are aware of some studies within the domain of SE, but none focused on time trade-offs.

There is awareness in parts of the empirical literature that normative decision theory has limited relevance for descriptive and explanatory purposes. Nonetheless, little empirical work surfaced that explicitly pursues empirically grounded, descriptive approaches, and none that studied trade-offs in time in depth.

However, the most explicit discussion of intertemporal choice and trade-offs were found in decisions about technical debt management. This suggests that the body of work on technical debt should be analyzed in more depth to characterize the tension between normative decision theory and descriptive approaches and to identify opportunities to improve our understanding of trade-off decision making in practice.

B. Threats to Validity

1) Internal Validity: Although we followed Kitchenham’s procedure for systematic literature reviews, minor deviations from the protocol should be noted:

- The documented of detailed codes being merged is not comprehensive. These codes were merged carefully and only when the terms were close, as discussed; however, this limits the traceability of analysis.
- Within the TD set, a corpus was constructed to enable future corpus-assisted discourse analysis. However, we were unable to include 4 papers which could not be converted from .pdf to text (presumably due to their encoding).

2) External Validity: The searches were limited to 3 databases, and no snowballing was conducted. This limits the external validity of our findings. However, the databases we used are commonly considered the main sources, and Google Scholar is often seen as the ‘most comprehensive’ source.

By including “time” in our search term, we wanted to get a sense of how time was treated as a dimension in empirical discussions of trade-offs in SE. However, not all relevant papers discuss time in this manner: some are about SE decision making in general. We were not trying to examine the assumptions underlying SE decision making in general, and our results cannot be generalized as such.

Some of the test searches also included the term “requirements engineering”, but this term was later dropped. While it is plausible that many of the results could be captured by software engineering, the search cannot be said to be generalizable to requirements engineering.

3) Construct Validity: Intertemporal choice is not a term that is used in SE literature. Indeed, there appears to be no blanket term for describing the types of decisions and tradeoffs that we attempt to study in this literature review. We thus refrained from the use of terms specific to the domain of intertemporal choice to ensure we identify how the SE community talks about these concerns. We believe this is an adequate measure to tease out intertemporal choice from the larger body of SE literature, but this cannot be guaranteed, as authors may speak about intertemporal choice in different terms that may have been missed by our search.

The search terms are known to be incomplete in the sense that terms related to ‘time’ and ‘decision making’, and disciplinary terms such as “requirements engineering”, have not

TABLE XI

|                      | Number of Papers |
|----------------------|------------------|
| Empirical            | 7                |
| Empirical Component  | 19               |
| Non Empirical        | 12               |
| Literature Review    | 1                |
| Total                | 38               |
been included in the search. This choice was taken since the aim was to identify common assumptions and decision making theories within an acceptable time frame.

4) Reliability: Not all researchers were trained software engineers, constituting a threat to the reliability of the literature review because terms might not be properly understood. This had to be addressed and considered in the setup of the protocol, as discussed above: We conducted iterative coding, reviewed with the internal expert, and erred on the side of caution. In order to ensure reliability in relevancy rating and coding, researchers conducted the work individually and then afterward compared. Where there was discontinuity between voters or where they were not sure how to evaluate a paper, results were obtained by consensus. If errors or ambiguities in the review process were discovered, another iteration was done to correct for that.

The internal and external review were conducted from a perspective of software engineering expertise.

C. Future work

To prepare the empirical study of time trade-offs, an analysis of the technical debt corpus is the logical next step: This work is very clearly bounded and focused on a closely related area of high relevance for the SE community. As part of this literature review, we have prepared a text corpus with the identified 231 papers that can be downloaded and analyzed quantitatively. This will make it possible to use complementary techniques such as corpus-assisted discourse analysis to identify the associations and meanings attributed to time trade-off decisions in the domain of technical debt management.

APPENDIX A

PRETEST SEARCH STRINGS

A. Pretest 1: June 30, 2016
• tradeoff AND software engineering OR requirement* engineering
• intertemporal choice AND software engineering OR requirement* engineering
• behavioral economic* AND software engineering OR requirement* engineering

B. Pretest 2: October 28, 2016
• (trade-off OR tradeoff OR "trade off") OR conflict
AND long-living OR "long living" OR "long lasting" OR "long-lasting" OR longevity OR "long term" OR "end of life" OR end-of-life OR future OR "life cycle" OR "life-cycle" OR lifecycle OR enduring OR temporal OR sustain*
AND "requirement engineering" OR "requirements engineering"

• (trade-off OR tradeoff OR "trade off") OR conflict
AND (long-living OR "long living" OR "long lasting" OR "long-lasting" OR longevity OR "long term" OR "end of life") OR end-of-life OR future OR "life cycle" OR "life-cycle" OR lifecycle OR enduring OR temporal OR sustain*
AND "requirement engineering" OR "requirements engineering"

APPENDIX B

TRADE-OFF DIMENSIONS DESCRIPTIONS

Here we captured self-reported dimensions that we encountered while coding:
• (benefit) as a direct component of cost-benefit analysis
• (competition)
• (complexity) Complexity of project architecture/code
• (labour)
• (methodology) software development methodology or system development life cycle
• (opportunities)
• (ROI) Return on Investment
• (goals) goals
• (techdebt) Technical Debt
• (vendor) vendor viability

APPENDIX C

SELECTED PAPERS

RESEARCH HAS AN EMPirical COMPONENT WHICH STUDIES DECISIONS

[A1] C. Zannier and F. Maurer, “A qualitative empirical evaluation of design decisions,” in ACM SIGSOFT Software Engineering Notes, vol. 30, no. 4. ACM, 2005, pp. 1–7.
[A2] M. Lindgren, R. Land, C. Norström, and A. Wall, “Key aspects of software release planning in industry,” in Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on. IEEE, 2008, pp. 320–329.
[A3] A. Martini and J. Bosch, “An empirically developed method to aid decisions on architectural technical debt refactoring: AnacondaRT,” in Proceedings of the 38th International Conference on Software Engineering Companion. ACM, 2016, pp. 31–40.
[A4] Z. Racheva and M. Daneva, “How do real options concepts fit in agile requirements engineering?” in Software Engineering Research, Management and Applications (SERAM), 2010 Eighth ACIS International Conference on. IEEE, 2010, pp. 231–238.
[A5] C. Wohlin and A. Aurum, “What is important when deciding to include a software requirement in a project or release?” in Empirical Software Engineering, 2005. 2005 International Symposium on. IEEE, 2005, pp. 10–pp.
[A6] T. K. Abdel-Hamid, K. Sengupta, and C. Swett, “The impact of goals on software project management: An experimental investigation,” MIS Quarterly, pp. 531–555, 1999.
[A7] T. Al-Naeem, I. Gorton, M. A. Babar, F. Rahbi, and B. Benatallah, “A quality-driven systematic approach for architecting distributed software applications,” in Proceedings of the 27th international conference on Software engineering. ACM, 2005, pp. 244–253.
[A8] J. B. Cushing, K. M. Winters, and D. Lach, “Software for scientists facing wicked problems lessons from the vistas project,” in Proceedings of the 16th Annual International Conference on Digital Government Research. ACM, 2015, pp. 61–70.
[A9] S. Hatton, “Choosing the right prioritisation method,” in Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on. IEEE, 2008, pp. 517–526.
[A10] J. Karlsson and K. Ryan, “A cost-value approach for prioritizing requirements,” IEEE software, vol. 14, no. 5, pp. 67–74, 1997.
[A11] F. Oliveira, A. Goldman, and V. Santos, “Managing technical debt in software projects using scrum: An action research.” in Agile Conference (AGILE), 2015. IEEE, 2015, pp. 50–59.
RESEARCH HAS AN EMPIRICAL COMPONENT & DISCUSS TRADE-OFFS OR LIKE DECISIONS

[B1] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, “Task allocation in distributed software development aided by verbal decision analysis,” in *Software Engineering Perspectives and Application in Intelligent Systems*. Springer, 2016, pp. 127–137.

[B2] J. S. Aguilar-Ruiz, I. Ramos, J. C. Riquelme, and M. Toro, “An evolutionary approach to estimating software development projects,” *Information and Software Technology*, vol. 43, no. 14, pp. 875–882, 2001.

[B3] J. Azar, R. K. Smith, and D. Cordes, “Value-oriented requirements prioritization in a small development organization,” *IEEE software*, vol. 24, no. 1, 2007.

[B4] G. Büyükozközl and O. Feyzoglu, “A fuzzy-logic-based decision-making approach for new product development,” *International journal of production economics*, vol. 90, no. 1, pp. 27–45, 2004.

[B5] P. Bajaj and V. Arora, “Multi-person decision-making for requirements prioritization using fuzzy ahp,” *ACM SIGSOFT Software Engineering Notes*, vol. 38, no. 5, pp. 1–6, 2013.

[B6] M. Bano and D. Zowghi, “Users’ voice and service selection: An empirical study,” in *Empirical Requirements Engineering (EmpRe), 2014 IEEE Fourth International Workshop on*. IEEE, 2014, pp. 76–79.

[B7] J. Bohnet and J. Dollner, “Monitoring code quality and development activity by software maps,” in *Proceedings of the 2nd Workshop on Managing Technical Debt*. ACM, 2011, pp. 9–16.

[B8] K. Boness and R. Harrison, “A data collection case study supporting requirements oriented prediction and management in software developments,” in *Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International*. IEEE, 2002, pp. 744–746.

[B9] M. Choekiertikul, H. K. Dam, T. Tran, and A. Ghose, “Predicting delays in software projects using networked classification (t),” in *Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on*. IEEE, 2015, pp. 353–364.

[B10] D. Dalcher, O. Benediktsson, and H. Thorbergsson, “Development life cycle management: a multiproject experiment,” in *Engineering of Computer-Based Systems, 2005. ECBS’05. 12th IEEE International Conference and Workshops on the*. IEEE, 2005, pp. 289–296.

[B11] X. Dong, Q.-S. Yang, Q. Wang, J. Zhai, and G. Ruhe, “Value-risk trade-off analysis for iteration planning in extreme programming,” in *Software Engineering Conference (APSEC), 2011 18th Asia Pacific*. IEEE, 2011, pp. 403–404.

[B12] C. Eben, “Technical controlling in software development,” *International Journal of Project Management*, vol. 17, no. 1, pp. 17–28, 1999.

[B13] A. Egyed and D. S. Wilke, “Support for managing design-time decisions,” *IEEE Transactions on Software Engineering*, vol. 32, no. 5, pp. 299–314, 2006.

[B14] S. Erofeev and P. Di Giacomo, “Usage of dynamic decision models as an agile approach to costs taxonomies construction,” in *Commercial-off-the-Shelf (COTS)-Based Software Systems*, 2006. Fifth International Conference on. IEEE, 2006, pp. 8–pp.

[B15] D. Falesi, M. A. Shaw, F. Shull, K. Mullen, and M. S. Keymind, “Practical considerations, challenges, and requirements of tool-support for managing technical debt,” in *Managing Technical Debt (MTD), 2013 4th International Workshop on*. IEEE, 2013, pp. 16–19.

[B16] M. Garg, R. Lai, and S. J. Huang, “When to stop testing: a study from the perspective of software reliability models,” *IET software*, vol. 5, no. 3, pp. 263–273, 2011.

[B17] V. Gaur, A. Soni, and P. Bedi, “An application of multi-person decision-making model for negotiating and prioritizing requirements in a project-oriented paradigm,” in *Data Storage and Data Engineering (DSDE), 2010 International Conference on*. IEEE, 2010, pp. 164–168.

[B18] S. Hassan, N. Bencomo, and R. Bahsoon, “Minimizing nasty surprises with better informed decision-making in self-adaptive systems,” in *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press, 2015, pp. 134–144.

[B19] H. Hedberg and N. Iivari, “Integrating hci specialists into open source software development projects,” in *IFIP International Conference on Open Source Systems*. Springer, 2009, pp. 251–263.

[B20] A. Jedlitschka, M. Ciolkowski, C. Denger, B. Freimut, and A. Schlichting, “Relevant information sources for successful technology transfer: a survey using inspections as an example,” in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. IEEE, 2007, pp. 31–40.

[B21] M. A. Khan, A. Parveen, and M. Sadiq, “A method for the selection of software development life cycle models using analytic hierarchy process,” in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*. IEEE, 2014, pp. 534–540.

[B22] S. U. R. Khan, S. P. Lee, M. Dabbagh, M. Tahir, M. Khan, and M. Arif, “Repiler: a framework for prioritization of software requirements,” *Frontiers of Information Technology & Electronic Engineering*, vol. 17, pp. 750–765, 2016.

[B23] K. Khriss, R. K. Keller, and I. A. Hamid, “Pattern-based refinement schemas for design knowledge transfer,” *Knowledge-Based Systems*, vol. 13, no. 6, pp. 403–415, 2000.

[B24] C.-K. Kim, D.-H. Lee, I.-Y. Ko, and J. Baik, “A lightweight value-based software architecture evaluation,” in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, vol. 2. IEEE, 2007, pp. 464–469.

[B25] M. Kimura and T. Fujiwara, “A new criterion for the optimal software release problems: Moving average quality control chart with bootstrap sampling,” in *International Conference on Advanced Software Engineering and Its Applications*. Springer, 2009, pp. 280–287.

[B26] M. S. Krishnan, T. Mukhopadhyay, and C. H. Kriebel, “A decision model for software maintenance,” *Information Systems Research*, vol. 15, no. 4, pp. 396–412, 2004.

[B27] N. Kumar and K. Srivathsan, “Enterprise risk evaluation and continuous mitigation using the fuzzy-multiattribute decision making-a conceptual approach,” *Journal of the Indian Institute of Science*, vol. 86, no. 6, p. 625, 2013.

[B28] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, “Project management aided by verbal decision analysis approaches: a case study for the selection of the best scrum practices,” *International Transactions in Operational Research*, vol. 22, no. 2, pp. 287–312, 2015.

[B29] R. Neves-Silva et al., “Context sensitive solution for collaborative decision making on quality assurance in software development processes,” in *Intelligent Decision Technologies: Proceedings of the 5th KES International Conference on Intelligent Decision Technologies (KES-IDT 2013)*, vol. 255. IOS Press, 2013, p. 130.

[B30] A. Nguyen-Duc, D. S. Cruzes, and R. Conradi, “On the role of boundary spanners as team coordination mechanisms in organizationally distributed projects,” in *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*. IEEE, 2014, pp. 125–134.

[B31] S. Nidhra, L. P. K. Satish, and V. S. Ethiraj, “Analytical hierarchy process issues and mitigation strategy for large number of requirements,” in *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on*. IEEE, 2012, pp. 1–8.

[B32] C. E. Otero, L. D. Gómez, S. Weisser, and A. Qureshi, “A multicriteria decision making approach for resource allocation in software engineering,” in *Computer Modelling and Simulation (UKSim), 2010 12th International Conference on*. IEEE, 2010, pp. 137–141.

[B33] H. Plate, S. E. Ponta, and A. Sabetta, “Impact assessment for vulnerabilities in open-source software libraries,” in *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*. IEEE, 2015, pp. 411–420.

[B34] Z. Pozgaj, H. Sertic, and M. Boban, “Strategies for successful software development project preparation,” in *Information Technology Interfaces, 2004. 26th International Conference on*. IEEE, 2004, pp. 679–684.

[B35] Z. Racheva, M. Daneva, A. Herrmann, and R. J. Wieringa, “A conceptual model and process for client-driven agile requirements prioritization,” in *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*. IEEE, 2010, pp. 287–298.

[B36] D. M. Raffo, W. Harrison, and J. Vandeville, “Software process decision support: making process tradeoffs using a hybrid metrics, modeling and utility framework,” in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*. ACM, 2002, pp. 803–809.

[B37] G. Ruhe, A. Eberlein, and D. Pfahl, “Trade-off analysis for requirements selection,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 13, no. 4, pp. 345–371, 2003.

[B38] M. Sadiq and S. Sultana, “A method for the selection of software testing techniques using analytic hierarchy process,” in *Computational Intelligence in Data Mining-Volume 1*. Springer, 2015, pp. 213–220.
Research has an empirical component & does not discuss trade-offs or like decisions.

[C1] N. Agarwal and U. Rathod, “Defining success for software projects: An exploratory revelation,” International journal of project management, vol. 24, no. 4, pp. 358–370, 2006.

[C2] V. Badura, A. Read, R. O. Briggs, and G.-J. De Vreede, “Coding for unique ideas and ambiguity: Measuring the effects of a convergence intervention on the artifact of an ideation activity,” in System Sciences (HICSS), 2010 43rd Hawaii International Conference on. IEEE, 2010, pp. 1–10.

[C3] C. G. Bernardo, D. A. Montini, D. D. Fernandes, D. A. da Silva, L. A. V. Dias, and A. M. da Cunha, “Using gym for testing design patterns in real-time and embedded systems on a software production line,” in Information Technology: New Generations, 2009. ITNG’09. Sixth International Conference on. IEEE, 2009, pp. 1397–1404.

[C4] A. Burns and M. A. Rathwell, “A communications environment for cooperation in information systems development,” Software Engineering Journal, vol. 2, no. 1, pp. 9–14, 1987.

[C5] R. P. Buse and T. Zimmermann, “Information needs for software development analytics,” in Proceedings of the 34th international conference on software engineering. IEEE Press, 2012, pp. 987–996.

[C6] J. A. Cottam, J. Hursey, and A. Lumsdaine, “Representing unit test data for large scale software development,” in Proceedings of the 4th ACM symposium on Software visualization. ACM, 2008, pp. 57–66.

[C7] G. A. Dafoulas and L. A. Macaulay, “Facilitating group formation and role allocation in software engineering groups,” in Computer Systems and Applications, ACS/IEEE International Conference on. IEEE, 2001, pp. 352–359.

[C8] J. N. och Dag, V. Gervasi, S. Brinkkemper, and B. Regnell, “Spreading up requirements management in a product software company: Linking customer wishes to product requirements through linguistic engineering,” in Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International. IEEE, 2004, pp. 283–294.

[C9] G.-J. De Vreede, R. M. Davison, and R. O. Briggs, “How a silver bullet may lose its shine,” Communications of the ACM, vol. 46, no. 8, pp. 96–101, 2003.

[C10] J. Ekanayake, J. Tappolet, H. C. Gall, and A. Bernstein, “Tracking concept drift of software projects using defect prediction quality,” in Mining Software Repositories, 2009. MSR’09. 6th IEEE International Working Conference on. IEEE, 2009, pp. 51–60.

[C11] S. K. Ethiraj, P. Kale, M. S. Krishnan, and J. V. Singh, “Where do capabilities come from and how do they matter? a study in the software services industry,” Strategic management journal, vol. 26, no. 1, pp. 25–46, 2005.

[C12] M. Feldgen and O. Chua, “Promoting design skills in distributed systems,” in Frontiers in Education Conference (FIE), 2012. IEEE, 2012, pp. 1–6.
[D1] T. Arao, Y. Machida, K. Toda, R. Yaegashi, and T. Takagi, “Decision-making about software release time using analytic hierarchy process,” in Advanced Applied Informatics (HIAIAI), 2014 IIAI 3rd International Conference on. IEEE, 2014, pp. 751–756.

[D2] P. G. Armour, ”Project portfolios: organizational management of risk,” Communications of the ACM, vol. 48, no. 3, pp. 17–20, 2005.

[D3] D. P. Ballou and G. K. Tayi, “A decision aid for the selection and scheduling of software maintenance projects,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 26, no. 2, pp. 203–216, 1996.

[D4] A. Bertolino, E. Marchetti, and R. Miranda, “Performance measures for supporting project manager decisions,” Software Process: Improvement and Practice, vol. 12, no. 2, pp. 141–164, 2007.

[D5] P. Bose and X. Zhou, “Wwac: Winwin abstraction based decision support,” in ACM SIGSOFT Software Engineering Notes, vol. 24, no. 2. ACM, 1999, pp. 127–136.

[D6] Y. Chen, G. C. Gannod, and J. S. Collofello, “A software product line process simulator,” Software Process: Improvement and Practice, vol. 11, no. 4, pp. 385–409, 2006.

[D7] M. Choetkiertikul and T. Sunetnanta, “A risk assessment model for offshoring using cmmi quantitative approach,” in Proceedings of the 20th International Conference on Systems, Man, and Cybernetics, 2010 Fifth International Conference on. IEEE, 2010, pp. 331–336.

[D8] G. Chroust, “The empty chair: uncertain futures and systemic dichotomies,” Systems Research and Behavioral Science, vol. 21, no. 3, pp. 227–236, 2004.

[D9] D. Di Caprio, F. J. Santos-Arteaga, and M. Tavana, “Information acquisition processes and their continuity: transforming uncertainty into risk,” Information Sciences, vol. 274, pp. 108–124, 2014.

[D10] M. A. A. Elsoud, H. A. Hefny, and E. S. Nasr,”A goal-based technique for requirements prioritization,” in Informatics and Systems (INFOS), 2014 9th International Conference on. IEEE, 2014, pp. SW–18.

[D11] Z. Pei and X. Liu, “f-ocêmo: fuzzy constructive cost model in software engineering,” in Fuzzy Systems, 1992., IEEE International Conference on. IEEE, 1992, pp. 331–337.

[D12] G. Forte, “Managing change for rapid development,” IEEE Software, vol. 14, no. 2, pp. 120–122, 1997.

[D13] A. Gachet and P. Haettenschwiler, “Developing intelligent decision support systems: A bipartite approach,” in International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, 2003, pp. 87–93.

[D14] D. Golpayegani and F. Shams, “Making a tradeoff between adaptation and integration in adaptive service based systems,” in Computer Science and Software Engineering (CSSE), 2015 International Symposium on. IEEE, 2015, pp. 1–7.

[D15] A. K. Jain, L. Ojha, and D. Pfahl, “Towards comprehensive experience-based decision support,” in European Conference on Software Process Improvement. Springer, 2004, pp. 34–45.

[D16] K. G. Koukoskuras and A. C. Georgiou, “A discrete event simulation model in the case of managing a software project,” European Journal of Operational Research, vol. 181, no. 1, pp. 374–389, 2007.

[D17] P. A. Laplante and C. J. Neill, “Modeling uncertainty in software engineering using rough sets,” Innovations in Systems and Software Engineering, vol. 1, no. 1, pp. 71–78, 2005.

[D18] Z. Racheva, M. Daneva, and A. Herrmann, “A conceptual model of client-driven agile requirements prioritization: Results of a case study,” in Proceedings of the 2010 acm-ieee international symposium on empirical software engineering and measurement. ACM, 2010, p. 39.

[D19] A. Redouane, “A decision making model for software design,” in Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on. IEEE, 2007, pp. 2687–2692.

[D20] ———, “A decision making model for web applications design,” in Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on. IEEE, 2006, pp. 651–657.

[D21] L. F. Ribeiro, M. A. d. F. Farias, M. Mendonça, and R. O. Spinola, “Decision criteria for the payment of technical debt in software projects: A systematic mapping study,” IICEIS 2016, p. 572, 2016.

[D22] J. Singh and L. Maurya, “Reliability assessment and prediction of open source software systems,” in Image Information Processing (IC IPP), 2013 IEEE Second International Conference on. IEEE, 2013, pp. 6–11.

[D23] N. D. Singpurwalla, “Determining an optimal time interval for testing and debugging software,” IEEE Transactions on Software Engineering, vol. 17, no. 4, pp. 313–319, 1991.

[D24] J. Steven, “Adopting an enterprise software security framework,” IEEE Security & Privacy, vol. 4, no. 2, pp. 84–87, 2006.

[D25] K. Sullivan, “Opportunity-centered software development,” in Proceedings of the FSE/SDP workshop on Future of software engineering research. ACM, 2010, pp. 369–374.

[D26] P. Vantakvikran and N. Prompoon, “Constructing a process model for decision analysis and resolution on costs selection issue of capability maturity model integration,” in Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on. IEEE, 2007, pp. 182–187.

[D27] Y. Wang and Y. Yuan, “The formal economic model of software engineering,” in Electrical and Computer Engineering. 2006. CCECE’06. Canadian Conference on. IEEE, 2006, pp. 2385–2388.

[D28] T. Wanyama and B. H. Far, “Towards providing decision support for costs selection,” in Electrical and Computer Engineering, 2005. Canadian Conference on. IEEE, 2005, pp. 908–911.

[D29] S. Zierner, P. R. F. Sampaio, and T. Ståhlane, “A decision modelling approach for analysing requirements configuration trade-offs in timeconstrained web application development,” in IEEE, 2006, pp. 144–149.

[RESEARCH DOES NOT HAVE AN EMPIRCICAL COMPONENT & DISCUSS TRADE-OFFS OR LIKE DECISIONS]

[E1] M. Chen and Y. I. Liou, “The design of an integrated group support environment,” in System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on, vol. 4. IEEE, 1991, pp. 333–342.

[E2] J. Cito, “Developer targeted analytics: supporting software development decisions with runtime information,” in Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. ACM, 2016, pp. 892–895.

[E3] T. Dingsøyr, F. O. Bjørnson, and F. Shull, “What do we know about knowledge management? practical implications for software engineering,” IEEE Software, vol. 26, no. 3, pp. 100–103, 2009.

[E4] A. Ejnioui, C. E. Otero, and A. A. Qureshi, “Software requirement prioritization using fuzzy multi-attribute decision making,” in Open Systems (ICOS), 2012 IEEE Conference on. IEEE, 2012, pp. 1–6.

[E5] C. Fiebig and C. Hayes, “Daisy: a design methodology for experience-centered planning support systems,” in Systems, Man, and Cybernetics, 1998, 1998 IEEE International Conference on, vol. 1. IEEE, 1998, pp. 920–925.

[E6] U. Franke, H. Holm, and J. König, “The distribution of time to recovery of enterprise it services,” IEEE Transactions on Reliability, vol. 63, no. 4, pp. 858–867, 2014.

[E7] S. R. Glazewski, “Risk management (is not) for dummies,” DTIC Document, Tech. Rep., 2005.

[E8] R. T. Hans and E. Minkandla, “Modeling software engineering projects as a business: A business intelligence perspective,” in AFRICON, 2013. IEEE, 2013, pp. 1–5.

[E9] M. T. Jelassi and R. A. Beaulac, “An integrated framework for group decision support systems design,” Information & Management, vol. 13, no. 3, pp. 143–153, 1987.

[E10] M. Kalumbilo and A. Finkelstein, “Linking strategy, governance, and performance in software engineering,” in Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering. ACM, 2014, pp. 107–110.

[E11] S. Karvonen, “Computer supported changes in project management,” International Journal of Production Economics, vol. 54, no. 2, pp. 163–171, 1998.

[E12] D. Kirk, “A flexible software process model,” in Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on. IEEE, 2004, pp. 57–59.

[E13] R. Kuehne, C. Wille, and R. Dumke, “Software agents using simulation for decision-making,” ACM SIGSOFT Software Engineering Notes, vol. 30, no. 1, p. 5, 2005.

[E14] E. Kuzana, E. Yana, Y. Sakamoto, Y. Nakamura, and K. Horikawa, “Computer-supported meeting environment for collaborative software development,” Information and Software Technology, vol. 38, no. 3, pp. 221–228, 1996.
REFERENCES

[1] World Commission on Environment and Development, “Our common future,” United Nations, Report, 1987.
[2] B. Penzenstadler, A. Ratnuri, D. Richardson, and B. Tomlinson, “Safety, security, now sustainability: The nonfunctional requirement for the 21st century,” IEEE software, vol. 31, no. 3, pp. 40–47, 2014.
[3] P. G. Neumann, “The foresight saga, redux,” Communications of the ACM, vol. 55, no. 10, pp. 26–29, 2012.
[4] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. M. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, “Requirements: The key to sustainability,” IEEE Software, vol. 33, no. 1, pp. 56–65, 2016.
[5] S. Frederick, G. Loewenstein, and T. O’Donoghue, “Time discounting and time preference: A critical review,” Journal of Economic Literature, vol. 40, no. 2, pp. 351–401, 2002.
[6] G. Loewenstein, D. Read, and R. F. Baumeister, Time and decision: Economic and psychological perspectives of intertemporal choice. Russell Sage Foundation, 2003.
[7] S. Bliff, A. Aurum, B. Boehm, H. Erdogan, and P. Grünbacher, Value-based software engineering. Springer Science & Business Media, 2006.
[8] R. L. Keeney and H. Raiffa, Decisions with multiple objectives: preferences and value trade-offs. Cambridge university press, 1993.
[9] J. Von Neumann and O. Morgenstern, Theory of games and economic behavior. Princeton University Press, 1944.
[10] A. Tversky and D. Kahneman, “Rational choice and the framing of decisions,” Journal of business, pp. S251–S278, 1986.
[11] G. Klein, “Sources of power: How people make decisions”, 1998.
[12] P. Lenberg, R. Feldt, and L.-G. Wallgren, “Towards a behavioral software engineering,” in Proceedings of the 7th international workshop on cooperative and human aspects of software engineering, ACM, 2014, pp. 48–55.
[13] P. Lenberg, R. Feldt, and L. G. Wallgren, “Behavioral software engineering: A definition and systematic literature review,” Journal of Systems and Software, vol. 107, pp. 15–37, 2015.
[14] R. Hofman, “Behavioral economics in software quality engineering,” Empirical Software Engineering, vol. 16, no. 2, pp. 278–293, 2011.
[15] S. Betz, C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. Venters, “Sustainability debt: A metaphor to support sustainability design decisions,” in Proceedings of the Fourth International Workshop on Requirements Engineering for Sustainable Systems, Ottawa, Canada, 2015.
[16] B. Kitchenham, “Procedures for performing systematic reviews,” Keele University, Joint Technical Report, 2004.
[17] IEEE, Software engineering body of knowledge (SWEBOK). IEEE, 2013. [Online]. Available: http://www.swebok.org
[18] E. Cartwright, Behavioral economics. Routledge, 2014, vol. 22.
[19] C. L. Cooper and C. Argyris, The Blackwell encyclopedia of management. Blackwell, 1997.
[20] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, and I. Gorton, “Measure it? Manage it? Ignore it? software practitioners and technical debt,” in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. ACM, 2015, pp. 50–60.
[21] C. Zaninni, M. Chiasson, and F. Maurer, “A model of design decision making based on empirical results of interviews with software designers,” Information and Software Technology, vol. 49, no. 6, pp. 637–653, Jun. 2007. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0950584907000122

ACKNOWLEDGMENT

Part of this work was supported by NSERC through RGPIN-2016-06640, and by the Connaught Fund. Special thanks to Ruzanna Chitchyan and Stefanie Betz for reviewing the literature review protocol.