This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

A Theory of Value for Value-based Feature Selection in Software Engineering

Pilar Rodríguez, Cathy Urquhart, and Emilia Mendes

Abstract— Value-Based Software Engineering stresses the role of value in software related decisions. In the context of feature selection, software features judged to provide higher value take priority in the development process. This paper focuses on what value means when selecting software features. Using grounded theory, we conducted and analyzed semi-structured interviews with 21 key stakeholders (decision-makers) from three software/software-intensive companies, within a context where value-based decision-making was already established. Our analysis led to the building of a theory of value for value-based feature selection that identifies the nature of value propositions considered by key stakeholders when selecting software features (i.e. decision-making criteria for deciding upon software features, as suggested by Boehm (2003)). We found that some value propositions were common to all three company cases (core value propositions), whereas others were dependent upon the context in which a company operates, and the characteristics of the product under development (specific value propositions). Moreover, value propositions vary according to the stakeholder group and the type of feature being assessed. Our study provides significant insight into value in the context of feature selection, and generates new concepts around value-based feature selection such as new value propositions.

Index Terms— Value-based Software Engineering (VBSE), Value Proposition, Decision-making Criteria, Grounded-theory, Theory Development, Feature Selection, Release Planning, Requirements Engineering, Decision-making, Decision-making Theory, Software Value, Software Products, Software-intensive Systems.

1 Introduction

SOFTWARE related decisions have traditionally taken place in a value-neutral setting in which short term aspects such as project costs and schedule were predominantly considered [1][2]. However, in the past 15 years, software organizations have increasingly moved from value-neutral to value-based decision-making, where decisions focus on what is best for the company's overall value creation [3][4][5]. Value-Based Software Engineering (VBSE) is the research area that stresses thinking value in software-related decisions [1]. Nowadays, the wide adoption of agile and lean software development [39], where value is a key concept, has also contributed to enlarging the applicability of VBSE to industry [2].

Feature selection, in the context of requirements engineering, is an area particularly impacted by the concept of value [5]. Features represent needs that define the behavior of a software/software-intensive product. They are gathered via meetings with customers or other stakeholders [6] and, once selected, are refined during a requirements elicitation process [7][8]. Value-based requirements engineering, to which feature selection belongs, is the first component cited in Boehm's VBSE agenda [1]. The inclusion or exclusion of specific features, which can take place within the context of, for example, a product's road mapping or release planning, will affect a product's value [9] and may impact vital company's processes, such as product pricing and marketing strategy. The potential features suitable for a given software release are many, compared to the features that can be realistically implemented when taking into account constraints such as time, cost and capability. In the context of value-based feature selection, features judged to provide more value take higher priority in being chosen for development. However, what value means for the stakeholders selecting those features (individually and collectively) has hitherto been unexplored [5] [10].

Previous research shows that value is an abstract concept in the software domain that involves many diverse aspects (e.g. [2][10][11][5]). In the context of VBSE, Biffl et al. [2] refer to value as follows: '*driven by both individual and collective goals, these stakeholders all hope to derive some benefit, whether tangible or intangible, economic or social, monetary or utilitarian, or even aesthetic or ethical. By the term value, we refer to this ultimate benefit, which is often in the eye of the beholder and admits multiple characterizations*'. Ojala [11], in his historical review of value, states that value '*has been defined in literature using economic, moral, aesthetic, social, political, religious, and judicial values*'. Overall, little knowledge exists on how value of a software/software-intensive system is actually defined [12][5][10]. In practice, even if not in a formalized way, value is determined as a set of value propositions (i.e. criteria/factors used to determine the value of an item)[1], usually not explicitly stated and subjectively employed by decision-makers [1][13][5]. Some studies have

- *P. Rodríguez is with M3S Group, University of Oulu, Oulu, Finland. E-mail: pilar.rodriguez@oulu.fi*
- *C. Urquhart is with the Faculty of Business and Law, Manchester Metropolitan University, United Kingdom. E-mail: c.urquhart@mmu.ac.uk*
- *E. Mendes is with the Department of Computer Science and Engineering, Blekinge Institute of Technology, Karlskrona, Sweden. E-mail: emilia.mendes@bth.se*

[1] The term value proposition is used in this paper in the way as it was firstly defined and employed within the context of VBSE [1][2]. This means that a value proposition encompasses any criterion/factor/aspect that is used to decide upon the benefits of an item, in our case a feature, when its value is being assessed. The scientific literature related to our research has used different terms to refer to the same concept. Value proposition is the term originally used by Boehm [1] and used also in his following works with other researchers (e.g. [58] and [2]). Other authors have used terms such as decision making criteria, value factors and values (e.g. [9][13-17]). More recently, works such as [3] and [10] have used the term value aspect.

focused on eliciting stakeholders' value propositions within the context of software product management (e.g. [5][9-10][13-18]). However, they have neither focused on the theoretical bases of value [19], nor upon the context of software feature selection.

Therefore, the goal and main contribution of this paper is to present a theory that focuses on value propositions that key stakeholders use when selecting features for software/software-intensive[2] product release planning. We focus on theory building because Software Engineering, as a relatively young discipline, lacks strong conceptual and theoretical foundations [20]. Thus, there is a growing interest in the development of theories in the field [21-26].

We applied Grounded Theory Method (GTM) to ground the theory in practice. Grounded Theory Method has proved a useful approach to theory generation in many fields [27] and it is gaining momentum in SE research (e.g. [20][28-30]). We built our theory from case study research in a similar way as described by Eisenhardt [31]. Our initial case study [5] helped us identify relevant stakeholders' value propositions when selecting features for release planning in a telecommunications company. From that case study (of Company A herein), we learned that stakeholders' value propositions seemed to be complementary in the decision process. We also discovered that whilst some value propositions were rather general, others seemed to be more product specific. In this work, we extend our research by studying two additional cases (Companies B and C) that helped us better understand the particularities of value propositions for feature selection, and build the theory outcome of this research. We use the findings from 21 transcribed interviews with key stakeholders from these three software companies in which a context of value-based decision-making was already established. As detailed in Section 3.1, two of the investigated companies developed software-intensive systems that differed significantly between them (Companies A and B), while the third one developed a standalone software solution (Company C). Such diversity enabled us to make fruitful comparisons between business domains/contexts, and, in line with [31], search for cross case patterns, thus increasing the likelihood of a robust and accurate theory.

The research problem addressed by this paper is "*how is value characterized with regard to value-based feature selection for release planning in the context of software and software-intensive products?*" Five research questions (RQ) underpin the development of our theory:

1. RQ1. What are the value propositions employed by key stakeholders when making decisions relating to feature selection for release planning?
2. RQ2. What are the overlapping and distinct value propositions between different stakeholders involved in value-based feature selection?
3. RQ3. What are the value propositions particular to a stakeholder's group?
4. RQ4. Are there any distinct value propositions between different types of features?
5. RQ5. Are there any overlapping and distinct value propositions between different company contexts such as business domains or types of products?

It has to be noted that although RQ1 has been the focus of our research from the beginning, including our previous work [5], RQ2-RQ5 emerged during our grounded theory (GT) journey. Evolving RQs can and do occur when building theories [31] and, particularly, when conducting GT as new concepts emerge from the analysis [20][27].

Our research makes a threefold contribution: first, our theory contributes to the body of knowledge of VBSE. Our contribution is unique, as only few studies have been able to research the details of value in the software domain [5][10]. Confidentiality issues often prevent research access to companies on this commercially sensitive topic. In our study, we offer a rare window into the details of value propositions employed when selecting features and the rationale behind them. Second, practitioners can use the theory outcome of this research as a tool to better understand the value criteria that (should) drive their decisions when selecting features. Third, we also demonstrate how GTM for theory building can be usefully employed to provide detailed insights into the basis of feature selection.

The paper is structured as follows: Section 2 presents related work in order to scope our research. The study design and the methodology employed to analyze value propositions and develop the theory are described in Section 3. Section 4 presents our findings, which include the empirically developed theory of value for value-based feature selection, as well as a detailed description of each of its components. Section 5 discusses how our findings answer our RQs and integrates them in light of related literature. Finally, the paper concludes in Section 6.

2 Background and Related Work

Our research began with a non-committal literature review as an accepted way to do a literature review without compromising our application of GTM [32][33]. The goal of a non-committal literature review is to limit the exposure to literature to prevent bias during data analysis, but at the same time to allow for researchers to position their research in the topic's body of knowledge. This position also assumes that further literature searching occurs once the theory is produced. The literature is further developed in Section 5, when our emergent theory is discussed in light of related work.

2.1 Value in Software Engineering

The *economic* value of software-related decisions, mainly in the form of short-term aspects such as costs, has been studied from the very early days of SE as a discipline (e.g. [34-36]). Accordingly, cost and cost estimation techniques have

Given the disparity of terms and the lack of consistency, we use the term value proposition in line with the way it was originally defined and used within the context of VBSE [1][2]. Please note that this understanding differs from the view of value proposition as a single criterion that focuses upon the benefits consumers get when buying a particular product or service (as used in works such as [77] and [78]).

[2] Systems in which software development and/or integration are dominant considerations (e.g. embedded systems) (ISO/IEC/IEEE 42010).

populated the SE literature ([37-38]). However, despite over fifty years of history [11], value as a concept did not take center stage in SE until 2003, when Boehm published his call to arms paper on value [1]. In that paper, Boehm suggested that much of SE practice and research was carried out in a value neutral setting. Furthermore, "*a 'separation of concerns'* [was] *practiced, in which the responsibility of software engineers* [was] *confined to turning software requirements into verified code*" [1]. This suggests that little attention was being paid to the *value* that software components added to the final product [1]. Boehm's article coined the concept of VBSE, stressing the importance of thinking about value in software-related decisions by balancing short and long-term aspects [2]. Value-Based Software Engineering introduced a wider view of value including perspectives such as "*relative worth, utility, or importance*" [2]. It also included the concept of *success critical stakeholders,* referring to all stakeholders who need to be involved in system definition and development [1].

Today, VBSE is a well-established research area. A recent systematic mapping study on the topic [19] identified 134 studies, including studies related to this research: Value-Based (VB) requirements engineering (30 studies), VB decision-making (7 studies), and theory of VBSE (7 studies referring to the same 4+1 theory). However, none of these studies have delved into the theoretical roots of value, and the nature of value propositions. The recent popularity of agile and lean software development [39], and its emphasis on delivering value to customers, have also driven the VBSE agenda forward, enlarging its scope [2], and popularizing terms such as *customer value* [40-41]. Overall, while there is an increasing use of value as a concept in the context of software development, particularly in industry [39], thus far there has been little research on understanding what exactly value means within SE [5][10].

2.2 Value-based feature selection

Feature selection is a part of the requirements engineering process [42]. It is the step in which product features are elicited from different sources (e.g. customers, legislative bodies, market reports, internal feedback, etc.) and their relative importance is judged. In value-based feature selection, features judged to provide more value for key stakeholders (i.e. decision makers) take higher priority in the development process.

Previous research on requirements and feature prioritization has tended to focus on methods and models for release planning (e.g. [43-45]), requirements prioritization techniques [46], and studies on managing dependencies between requirements/features (e.g. [47-48]). This body of knowledge has often included an inherent dimension of importance, based on complex multi-criteria, which resembles value. Only a few prioritization techniques have explicitly considered value though [46]. Thus, its main focus has not been on understanding value or providing empirical evidence on value, leaving the notion of value vague or implicit [46]. By contrast, our study focuses on understanding, via knowledge explicitation, the value propositions that stakeholders consider when selecting a given feature.

Some SE literature reviews and mapping studies have explicitly focused on value propositions [3][49][50]. However, in these studies, the concept of value is given from multiple scenarios rather than with a specific focus on value based feature selection. Jan and Ibrar [49]'s mapping on value aspects within the context of SE shows that research contributions in the area are often isolated, and with a limited choice of value aspects; for example, focusing only on cost, or on product characteristics such as usability. Khurum et al. [3] added additional literature from economics, business and marketing research to the work from [49] and proposed a classification of 62 value components. This classification aimed to be general enough to represent the views of all stakeholders who make any decision relating to any software product. It was later extended to also include a project management perspective [50].

Although these studies help us understand the state of the art on value propositions, they have some shortcomings when understanding the concept of value in the context of feature selection. First, they do not look into how different companies interpret the concept of value [10] but provide large classifications of value propositions that are difficult to apply in practice because not all propositions may be suitable for every context (e.g. feature selection) [5][10]. Further, these classifications are mostly, or entirely, not grounded in practice, but deductively developed from related literature, including not only literature from the software domain but from other domains such as marketing and economics. This is a risk when trying to understand value because, in what seems to be common throughout previous research in this topic, selection criteria are often suggested only with illustrative purposes and, therefore, there is scarce empirical evidence that supports their relevance in practice [13][10][5].

That said, similar to our first case study [5], some empirical studies have also focused on concrete value propositions that are used for feature/requirement selection in SE ([9-10][13-18]). However, none of these studies went further to provide insights on the theoretical basis that underspin the concept of value in this context (feature/requirement selection). Indeed, most of them employed research methods such as surveys, which limits the richness of data and details available [51]. These studies are further developed in Section 5.

2.3 Theories in Software Engineering

This section discusses theory building in SE and, in particular, empirically based theories (i.e. theories that are built or modified based on empirical research) [21]. Before starting the discussion, it is convenient to clarify the meaning of theory. Theory, as a concept, is a source of continuing debate [21-23]. In this study, we adhere to the definition provided by Shirley Gregor in [52], where a theory is defined as an attempt to generalize local observations into more abstract and universal knowledge, aiming to explain or predict a phenomenon. Gregor distinguishes between four primary goals of a theory: analysis & description, explanation, prediction and prescription. Our theory's goal is to analyze and describe value propositions fundamental to value-based feature selection for release planning.

As an applied discipline, a large portion of SE research over time has focused on technological aspects (e.g. programming languages) or process-related methods to promote better ways of developing software [53]. However, the theoretical foundations of the field have comparatively received little focus [21-22]. Arguments in favor of building and using theories are widely reported in the SE literature (e.g. [54], [21-25]). Theories structure facts and knowledge in a concise manner, offering common conceptual frameworks and facilitating knowledge communication [21]. Johnson et al. argue that without the support of theories, SE would be relegated to a costly trial and error process [22]. Moreover, theories enable not only statistical generalization but also analytical generalization in situations in which statistical generalization is not desirable or possible [21]. Overall, a field without theories remains blind. The need for theories in SE is evidenced by recent research [21-23][26]. Particularly, theories grounded in practice are getting momentum in SE (e.g. [20][29][30]).

Examples of existing theories in SE include general theories [23] such as [55], [24], and [25], and middle range theories [23], such as [56] and [57]. However, in the area of VBSE, we can find only one theory: the 4+1 theory of VBSE [58]. The theory centers on the success-critical stakeholders win-win Theory W [57], which stresses the importance of identifying the values that are fundamental for the success-critical stakeholders to achieve and maintain a win-win state. However, the nature of the concept of value, and concrete value propositions are not considered in Theory W.

Why should we care about a theory of value for feature selection? One issue that the scientific community in SE seems to agree on is that SE theories should be useful to the software industry [21][23] and provide answers to significant questions for the discipline [22]. Johnson et al. [22], who reflected on the role of theories in SE, use the example of how to specify system requirements as one of the most significant questions in SE because, although many methods and requirements specification techniques and languages exist, very few explicit theories, if any, explain why certain requirements are selected [22]. Although value propositions are often referenced as an integral decision-making part, they are poorly documented and rarely explicitly utilized, even within development processes that emphasize value (e.g. agile). As a result, decision-making is usually carried out based on implicit and tacit value propositions, being subject to ambiguity and vagueness, which are likely to bias decisions [5][10]. By creating a theory of value for value-based feature selection we aim at i) increasing our understanding and analysis of what value represents in decision making in software development and, concretely, feature selection during release planning, and ii) providing practitioners with a tool that they can use to better understand the value criteria that should drive their decisions when selecting features for a release.

2.4 Research gaps

In summary, the next research gaps motivate this study:

- Software companies are progressively moving to value-based decision-making [1][5][10], so there is now a pressing need to understand in detail decision-making processes within the context of VBSE.
- Very few studies have focused on value in the context of feature selection [5][13][22], despite feature selection for release planning being an essential activity for the success of software products.
- The few studies that do focus on value propositions in SE mainly tend to have a general scope and lack specificity in describing value propositions. To date, no theory generation work has been carried out in this area.
- The lack of theoretical foundations in SE [21-22]. In particular, the existence of only a single VBSE theory [58], which has no particular focus on the concept of value (value propositions).

3 Study Design and Research Methodology

This study draws on the principles of GT [59] and applies a multiple-case study strategy [60-61]. This is a similar approach used by Eisenhardt's (1989) seminal paper in management on building theory from case studies [31]. We followed the guidelines for case study by Yin [60] and its adaptation to SE by Runeson and Höst [61], the guidelines by Urquhart for GTM [27] and the guidelines by Stol et al. [20] on how to report GT studies in SE. From an epistemological position, the study is positivist [20], where results from interviews' coding are checked by more than one researcher for validation purposes. We adopted the Glaserian version of GTM [27] because it has been characterized as indelibly positivist by some commentators (e.g. [62])[3].

The study is part of a larger research effort, the VALUE framework, which focuses on decision-making within the context of software product management [4][65]. Herein we present our complete foundational and emergent theory of value for value-based feature selection. Some of the value propositions used in our theory building have already been presented in two previous articles [4][5]. In Mendes et al. [4], we presented the overall research framework to which our GT study belongs, together with a preliminary analysis of the interviews conducted in Company C. This current paper further develops the analysis of case C's interviews to create robust foundations for our theory. Rodríguez et al. [5] focused on understanding value propositions used for selecting features in Company A. We build on the findings from [5], using the previous open

[3] It is widely accepted that GT is '*a neutral container into which any philosophy can be poured*' [64], meaning that it is a research method which can be comfortably used in all paradigms. However, it is true to say that there has long been debate about whether GT is inherently positivist or interpretivist. Some people characterise Glaserian GT as positivist, and Straussian GT as interpretivist, given that Glaser was influenced by Lazarsfeld (a renowned sociologist and statistician), and that Glaser has written a book on quantitative GT [63]. By contrast, Strauss came from symbolic interactionism. However, other authors disagree with this classification by considering that GT is based on emergence and interpretation, where there is no single objective reality that is independent from the observer. Bryant [62] characterised Glaser's reliance on emergence from data as phenomenalist and GT as indelibly positivist. As such, we felt that using Glaserian GT from a positivist perspective was reasonable. Other scientific literature has taken as similar position with regard to positivist GT (e.g. [78]).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

TABLE 1
COMPANY CASES – CONTEXTUAL INFORMATION (ADAPTED FROM PETERSEN AND WOHLIN [69])

Context facet	Context element	Company A	Company B	Company C
Product	**Name**	Product A	Product B	Product C
	Domain	Telecommunications	Telecommunications	Security software
	Product type	Embedded system	Embedded system	Desktop application
	Size	Commercial product, large-scale	Commercial product, large-scale	Commercial product, large-scale
	Maturity	New product (second release under development)	New product (second release under development)	Mature company product (one of the company's key products).
	Customization	Tailored to different customers	Tailored to different customers	Tailored to different customers
Process	SW development process	Scrum-based. Continuous/incremental development of features. Features written in natural language and managed using product backlogs.	Scrum-based. External releases every quarter, internal releases every two weeks. Features written in natural language and managed using a product backlog.	Scrum-based. Continuous/incremental development of features. Features written in natural language and managed using a product backlog.
Practices and tools for feature selection	Practices	Iterative feature selection process (monthly meetings at strategic level, weekly meetings at development level)	Iterative feature selection process (monthly meeting at steering group level, weekly meetings at change mgmt. group/product development program).	Iterative feature selection, encompassing continuous discussion loops between activities (product definition, design and development).
	Tool support	Two product backlogs: Company Product Backlog and Dev. Product Backlog. Commercial decision analysis tool at company level. MS Excel at development.	Customize web portal to receive new feature requests (used by all stakeholders, including customers). JIRA for managing the product backlog.	JIRA for managing the product backlog.
People	**Roles involved in feature selection**	Three SPMs located at the company headquarters. Product owner, project managers and technical stakeholders located at the development unit.	Steering group involving different roles, including product manager, business managers, technical leads, and project managers.	Three stakeholders (program manager, product manager and product marketing director), who centralize information from other stakeholders.
Organization	**Size**	Large organization	Medium organization	Large organization
	Organizational model	Several locations distributed worldwide	Nine locations distributed in four countries	Several locations distributed worldwide
	Organizational unit(s) involved in the study	Company's headquarters (SPMs) Development unit (in charge for developing Product A)	Company's headquarters	Company's headquarters
	Distribution	Decision making team distributed in two locations (company headquarters and development unit).	Decision making team located at the company's headquarters. System locally developed at the company's headquarters.	Decision making team located at the company's headquarters. System development distributed in several locations.
Market	**Number of customers**	Market-driven development (over 150 customers)	Market-driven development (over 10 customers)	Market-driven development (over 200 customers)
	Setting	Business-to-business (B2B)	Business-to-business (B2B)	Business-to-business (B2B) and Business-to-consumer (B2C)
	Constraints	Market characterized by uncertainty and short time-to-market	Market characterized by uncertainty and short time-to-market	Market characterized by uncertainty and short time-to-market

coding from the interviews in Company A, and add all the three coding steps (i.e. open, selective and theoretical coding) for the interviews obtained from Company B and Company C, in order to build our grounded theory. In the next sections, we explain each phase of the research design.

3.1 Case selection

In selecting our cases, we followed the guidelines by Yin [60] and Urquhart [27]. As Yin says on p.31 '*a fatal flaw in doing case studies is to conceive of statistical generalization as the method of generalizing the results of the case*' [60]. In GT, we also generalize to a theory, not to a population and, therefore, we need to ensure that our cases are representative of the substantive area for which the theory is produced [27]. Theoretical sampling, which identifies data based on gaps in the emerging theory, is seen as ideal in GTM [20]. However, researchers in SE are frequently limited by the possibilities for selecting cases and, in practice, many cases are chosen based on availability [61].

We followed a strategy of analytic generalization where our cases were selected for analytical reasons (i.e. being particularly suitable for illuminating and extending relationships and logic among constructs [66]). We used a mix of convenience [67] and purposive [68] sampling. A crucial aspect for our study was that our companies were already consciously considering value in their software feature decisions [4][5]. In our three companies, decision makers expressed their decision in terms of value and decision-making teams involved not only technical stakeholders but also different backgrounds, such as sales and marketing. Thus, purposively, we selected companies believed to represent rich cases for our study.

In addition, our cases complement each other, in line with the concept of theoretical sampling. We noticed from our study with Company A [5] that some value propositions identified seemed dependent upon the type of product Company A develops and the business domain in which it operates. Through Companies B and C we studied whether this was also the case. The second and third case studies supported the emerging category '*specific value propositions*'. We used differentiated cases, as follows: Company A operates in the telecommunications domain and develops an embedded product that is part of the network. Company B also operates in the telecommunications domain and develops an embedded product as well. However, in the case of Company B, the embedded product is a phone device. Therefore, Companies A and B operate in the same domain, but, crucially, develop different kinds of products. In the case of Company C, it develops security software solutions; thus, the product is not an embedded system but a software application. Therefore, the companies we investigated share some similarities as they all develop software (embedded or not) and select software features for the upcoming releases based on value. However, they differ with regard to their corresponding business domain (Company C differs from Companies A and B) and the type of product that they develop. Our strategy aimed to generate *core value propositions* by similarity in data, while at the same time also checking for this category's usefulness. We hypothesized that data diversity could give us insights about *specific value propositions*. As described by Glaser and Strauss ([59] p.58), varying the conditions between cases allows for '*identifying/developing fundamental differences under which category and hypothesis vary*'.

Regarding the software development process employed by the companies, they all used Agile-based software development characterized by iterative development. With their own particularities, in all three cases, releases were periodically designed, and features that would be part of a concrete release periodically selected. Table 1 provides a detailed description of the context of our company cases based on the checklist and suggestions provided in [69]. Note that some details were anonymized or excluded to comply with the non-disclosure agreements (NDAs) signed with our case companies. Main differences between company cases are highlighted in Table 1.

3.2 Data collection

Data was gathered via 21 interviews with key stakeholders. These stakeholders represent decision-makers involved in feature selection for release planning with a diversity of roles, including strategic product managers, product owners, different kinds of project managers and technical stakeholders. Appendix A presents an overview of the Interviewees (I). Note that each company uses its own terminology to refer to equivalent roles. For example, strategic product managers in Company A carry out very similar tasks to program and product managers in Companies B and C. It is important to note also that the number of interviews in these types of study is of necessity limited as the study targets at key strategic personnel and roles in each organization [9].

A champion in each of the companies (I10, I16 and I21 in Appendix A) helped us identify the stakeholders to be interviewed. In relation to Company A, we interviewed all key stakeholders involved in features selection for Product A; therefore, we can claim that we have a complete picture of value in this case [5]. Regarding Company B, our company champion selected the most important stakeholders involved in feature selection for Product B (at least one representative from each stakeholder group). For the third company, Company C, we interviewed the three stakeholders who compose the feature selection team for Product C and are the accountable persons for deciding upon its features. These stakeholders centralize information needed when selecting features from other stakeholders, both inside and outside the organization (e.g. customers, development teams, maintenance teams, etc.).

The interviewees were domain experts with a wide experience in the software industry (Mdn. 20 years, 1st qu. 15.0, 3rd qu. 21.00) and in the business domain inside their companies (Mdn. 4.0 years, 1st qu. 1.0, 3rd qu. 17.00); they also had good experience in feature selection tasks (Mdn. 24 months, 1st qu. 12.0, 3rd qu. 36.00).

In each of the three cases, also usual in the software industry [4], decisions were mostly based on decision makers' tacit knowledge. Thus, to fulfill our RQs, the interviews aimed at eliciting the factors that are subjectively used by decision makers when deciding upon the value of a feature. The interviews were semi-structured and largely conversational to give room for participants to express their understanding of value when selecting features. To phrase our questions, preparatory meetings were held with company champions to understand each case's feature selection process and own terminology. We designed an interview script (available in Appendix B) composed of three sections: 1) warm-up questions, including demographic and context setting questions (approximately 5 min); 2) value propositions elicitation questions, in order to elaborate into the details of the set of value aspects that the interviewee considers when deciding upon a feature (45 min); and 3) wrap-up questions, to ascertain any missing relevant topic that the interviewee would like to discuss prior to concluding the interview (10 min). Our focus was only on selecting software features and, therefore, we did not target at the products' hardware side in cases A and B. In all cases, we made it clear upfront that the focus was on software features' selection for products' releases. All interviews were planned to last for approximately one hour. The actual interview length is showed in Appendix A. Most interviews were face-to-face, and all of them were voice recorded and transcribed. All the interviews were conducted by the first and third authors together.

3.3 Data analysis

Data analysis proceeded from open coding (identifying categories, properties and dimensions) through selective coding (clustering around categories), to theoretical coding (abstracting the theory and establishing relationships) - Glaserian version of GT [27]. We utilized the principle of constant comparison as adopted by Glaser and Strauss [59]. Continuously comparing codes between single interviews, different interviews and different cases helped densify our major categories of *core* and *specific* value propositions, and supported us following a particular analytical path. Moreover, it helped us get a more nuanced understanding of what value propositions might consist of.

When coding the interviews during open coding, we did not restrict value to any existing definition or classification of value propositions but rather stayed open to what the data could be telling us (inductive analysis). The list of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

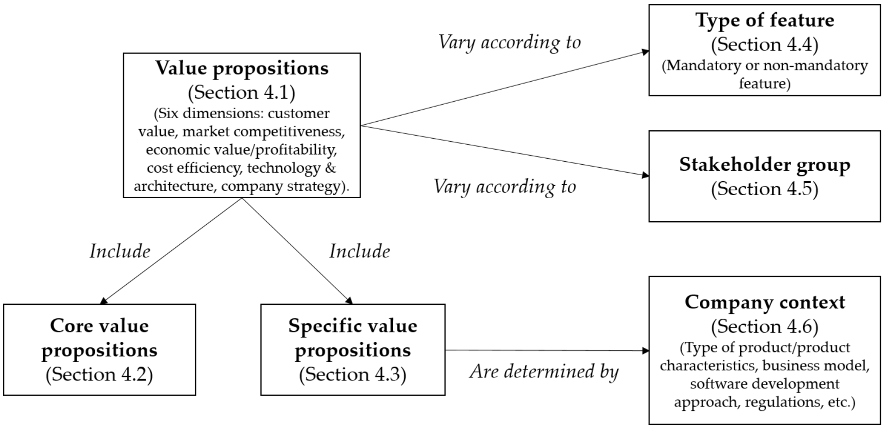

Value propositions
(Section 4.1)
(Six dimensions: customer value, market competitiveness, economic value/profitability, cost efficiency, technology & architecture, company strategy).
Vary according to
Type of feature
(Section 4.4)
(Mandatory or non-mandatory feature)
Vary according to
Stakeholder group
(Section 4.5)
Include
Include
Core value propositions
(Section 4.2)
Specific value propositions
(Section 4.3)
Are determined by
Company context
(Section 4.6)
(Type of product/product characteristics, business model, software development approach, regulations, etc.)

Fig. 1. Theory of value for value-based feature selection (software features) in software/software intensive product release planning

value propositions identified using open coding from each interview was emailed to the corresponding interviewee for validation purposes. We received feedback from 6/21 interviewees and made modifications accordingly. During selective coding, the open codes were grouped into larger categories, which acted as the foundation for the major categories of our theory. Definitive categories were established when no more open codes emerged from the analysis – this is known as theoretical saturation. In theoretical coding, the major categories that emerged from the selective coding were related to each other and the relationships between them considered to build our theory. We used theoretical memos to help us theorize about categories and relationships [70]. NVivo was used in the analysis. Appendix C presents an example to illustrate the analysis process.

4 Findings

Figure 1 presents the theory of value for value-based feature selection in release planning derived from our GT study. The theory is composed of six constructs:

1. Value propositions
2. Core value propositions
3. Specific value propositions
4. Type of feature
5. Stakeholder group
6. Company context

Our analysis revealed that *value propositions* for feature selection (software features) can be classified into two groups: *core value propositions* and *specific value propositions*. Core value propositions include propositions that were common to the three cases and, therefore, suggest a pattern. Specific value propositions are dependent on the *company context* and, therefore, were specific to each company case. We also found out that value propositions vary according to two factors: 1) the *type of feature* being assessed (e.g. mandatory features as opposed to non-mandatory ones), and 2) the *stakeholder group* (e.g. different stakeholder groups focused on different value propositions).

The next subsections describe each of these constructs and their relationships. The relationships in our theory represent association but not causality. We use illustrative quotations from the interviews, as they were identified in the analysis, to provide supporting evidence to our findings [27]. Sometimes we needed to make slight modifications to some quotations to comply with our NDAs. Whenever this occurred, we used the tag: [*modified text*].

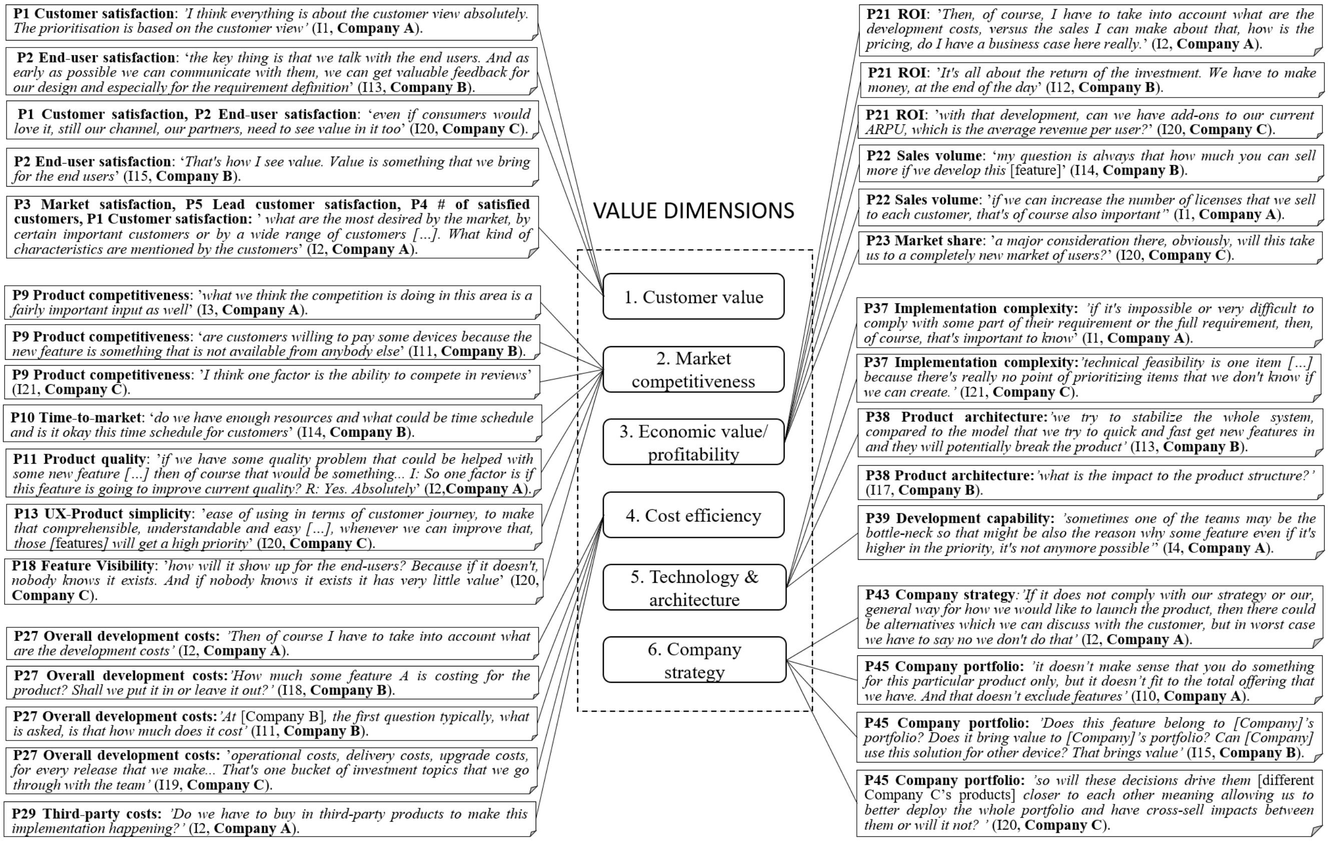
4.1 Value propositions

A value proposition is any kind of aspect, criterion or consideration that is used by key stakeholders (i.e. decision makers), within the context of value-based decision-making, to decide whether a software feature should be included in a release. Thus, every value proposition characterizes an aspect/criterion/consideration that can be impacted upon by the choice of features. For example, a value proposition is *Customer satisfaction* because when the value of a feature is assessed, the impact of that feature on customer satisfaction is considered. Similarly, another value proposition is *Company's brand* as the impact that a feature will have upon the company as a brand (e.g. being aligned with the image that the company wants to project) is taken into consideration as well. Many value propositions emerged from the interviews. Some of them were semantically similar, despite interviewees using a different terminology to refer to the same concept (e.g. *consumer satisfaction* and *end-user satisfaction*). We analyzed each proposition in detail and aggregated those that were semantically equivalent. The complete list of value propositions that emerged in our study (47) is included in Appendix D.

We categorized value propositions into six dimensions: 1) *customer value* (P1-P8), 2) *market competitiveness* (P9-P20), 3) *economic value/profitability* (P21-P26), 4) *cost efficiency* (P27-P36), 5) *technology & architecture* (P37-P42) and 6) *company strategy* (P43-P47). Illustrative quotations for some value propositions are shown in Figure 2 to clarify the way in which these dimensions emerged during the analysis.

Propositions belonging to the *customer value* dimension refer to aspects that are relevant from a customer's perspective. Our analysis showed that the customer view, especially those features desired by key customers (P5, Appendix D) or by a large amount of customers (P4) seemed highly important when selecting features for a release. *Customer value* aspects were those mentioned the most in the interviews (387 instances in NVivo – see Appendix D).

Market competitiveness comprises value propositions that reflect a product's competitiveness in the market in terms


VALUE DIMENSIONS
1. Customer value
2. Market competitiveness
3. Economic value/ profitability
4. Cost efficiency
5. Technology & architecture
6. Company strategy

P1 Customer satisfaction: *'I think everything is about the customer view absolutely. The prioritisation is based on the customer view'* (I1, **Company A**).

P2 End-user satisfaction: '*the key thing is that we talk with the end users. And as early as possible we can communicate with them, we can get valuable feedback for our design and especially for the requirement definition*' (I13, **Company B**).

P1 Customer satisfaction, P2 End-user satisfaction: '*even if consumers would love it, still our channel, our partners, need to see value in it too*' (I20, **Company C**).

P2 End-user satisfaction: '*That's how I see value. Value is something that we bring for the end users*' (I15, **Company B**).

P3 Market satisfaction, P5 Lead customer satisfaction, P4 # of satisfied customers, P1 Customer satisfaction: ' *what are the most desired by the market, by certain important customers or by a wide range of customers [...]. What kind of characteristics are mentioned by the customers*' (I2, **Company A**).

P9 Product competitiveness: '*what we think the competition is doing in this area is a fairly important input as well*' (I3, **Company A**).

P9 Product competitiveness: '*are customers willing to pay some devices because the new feature is something that is not available from anybody else*' (I11, **Company B**).

P9 Product competitiveness: '*I think one factor is the ability to compete in reviews*' (I21, **Company C**).

P10 Time-to-market: '*do we have enough resources and what could be time schedule and is it okay this time schedule for customers*' (I14, **Company B**).

P11 Product quality: '*if we have some quality problem that could be helped with some new feature [...] then of course that would be something... I: So one factor is if this feature is going to improve current quality? R: Yes. Absolutely*' (I2,**Company A**).

P13 UX-Product simplicity: '*ease of using in terms of customer journey, to make that comprehensible, understandable and easy [...], whenever we can improve that, those [*features*] will get a high priority*' (I20, **Company C**).

P18 Feature Visibility: '*how will it show up for the end-users? Because if it doesn't, nobody knows it exists. And if nobody knows it exists it has very little value*' (I20, **Company C**).

P27 Overall development costs: *'Then of course I have to take into account what are the development costs'* (I2, **Company A**).

P27 Overall development costs:*'How much some feature A is costing for the product? Shall we put it in or leave it out?'* (I18, **Company B**).

P27 Overall development costs:*'At* [Company B]*, the first question typically, what is asked, is that how much does it cost'* (I11, **Company B**).

P27 Overall development costs: '*operational costs, delivery costs, upgrade costs, for every release that we make... That's one bucket of investment topics that we go through with the team'* (I19, **Company C**).

P29 Third-party costs: *'Do we have to buy in third-party products to make this implementation happening?'* (I2, **Company A**).

P21 ROI: '*Then, of course, I have to take into account what are the development costs, versus the sales I can make about that, how is the pricing, do I have a business case here really.*' (I2, **Company A**).

P21 ROI: '*It's all about the return of the investment. We have to make money, at the end of the day*' (I12, **Company B**).

P21 ROI: '*with that development, can we have add-ons to our current ARPU, which is the average revenue per user?*' (I20, **Company C**).

P22 Sales volume: '*my question is always that how much you can sell more if we develop this* [feature]' (I14, **Company B**).

P22 Sales volume: '*if we can increase the number of licenses that we sell to each customer, that's of course also important"* (I1, **Company A**).

P23 Market share: '*a major consideration there, obviously, will this take us to a completely new market of users?*' (I20, **Company C**).

P37 Implementation complexity: *'if it's impossible or very difficult to comply with some part of their requirement or the full requirement, then, of course, that's important to know*' (I1, **Company A**).

P37 Implementation complexity:*'technical feasibility is one item [...] because there's really no point of prioritizing items that we don't know if we can create.'* (I21, **Company C**).

P38 Product architecture:*'we try to stabilize the whole system, compared to the model that we try to quick and fast get new features in and they will potentially break the product'* (I13, **Company B**).

P38 Product architecture:*'what is the impact to the product structure?'* (I17, **Company B**).

P39 Development capability: *'sometimes one of the teams may be the bottle-neck so that might be also the reason why some feature even if it's higher in the priority, it's not anymore possible"* (I4, **Company A**).

P43 Company strategy*:'If it does not comply with our strategy or our, general way for how we would like to launch the product, then there could be alternatives which we can discuss with the customer, but in worst case we have to say no we don't do that'* (I2, **Company A**).

P45 Company portfolio: *'it doesn't make sense that you do something for this particular product only, but it doesn't fit to the total offering that we have. And that doesn't exclude features'* (I10, **Company A**).

P45 Company portfolio: *'Does this feature belong to [Company]'s portfolio? Does it bring value to [Company]'s portfolio? Can [Company] use this solution for other device? That brings value'* (I15, **Company B**).

P45 Company portfolio: *'so will these decisions drive them* [different Company C's products] *closer to each other meaning allowing us to better deploy the whole portfolio and have cross-sell impacts between them or will it not? '* (I20, **Company C**).

Fig. 2. Value dimensions for value-based feature selection (software features) and illustrative quotations for some value propositions.

of added value compared to similar products. Interviewees considered the features offered by other competitors' products, and what benefits these features provide to customers, when considering their own value proposition. Interviewees frequently mentioned Product quality (P11) – and aspects defining product quality, and Time-to-market (P10) as important for being competitive.

The *economic value/profitability* dimension includes propositions that relate to the impact that the implementation of a feature would have upon a company's wealth, mainly in monetary terms. Aspects such as ROI (P21), sales volume (P22), capability to increase customer base/market share (P23) and impact that the implementation of the feature would have on a product's price (P24) were mentioned during the interviews. However, it is interesting to notice that economic value was not the center of focus of our interviews (120 instances in NVivo, compared to 387 for customer value and 273 for market competitiveness).

A related dimension is *cost efficiency*, which represents the several costs that implementing a feature would entail. At the end, the economic value/profitability of a feature will be impacted by the incurred costs when implementing that feature. For instance, some features might be deemed too costly to implement. Costs were distributed in different categories such as testing costs (P30) or investment in third-part components (e.g. subcontractors) (P29).

The *technology & architecture* dimension centers on technical aspects such as product architecture (P38) and existing technical capability to implement a feature (P39). If the technology platform makes a feature difficult to implement or introduces complexity, again a feature may be deferred or not implemented.

Finally, the *company strategy dimension* focuses on strategic alignment and aspects that provide overall direction to the company in terms of organizational goals and plans designed to achieve these goals. So it might be that a particular feature does not fit with the company's business direction, as one interviewee explained: '*so if it does not comply with our strategy or our general way for how we would like to launch the product, then, there could be alternatives, which we can discuss with the customer, but in worst case we have to say no, we don't do that, we don't sell that*' (I2, Company A).

Clearly, some of these dimensions are interrelated. For instance, stakeholders tended to consider economic value and cost efficiency together. That said, stakeholders mentioned these propositions separately and with sufficient frequency to convince us that they should remain separate dimensions in our theory. Moreover, although some value propositions clearly provide value and should be maximized (e.g. customer satisfaction, ROI), other propositions, which are also important, could be interpreted as 'anti-values', given that they should be minimized instead (e.g. costs). I20 explained it in the following way '[value] *it's all about maximizing the positive and minimizing the negative*'. We also realized that some propositions were core to all companies, whereas some were specific to a company's context, as we explain in the next sections.

4.2 Core value propositions

Core value propositions refer to those propositions used in all three company cases, thus representing commonality in the choice of value propositions. As stated in Sections 1 and 3, there was a wide diversity between the three company cases. Thus, we theorized that those core value propositions are commonly used in value-based feature selection for release planning.

Twenty-four core value propositions were identified, which include propositions such as customer satisfaction (P1), product competitiveness (P9), time-to-market (P10), return-on-investment – ROI (P21), reusability (P28), implementation complexity (P37), development capability (P39) and product strategy (P44). For example, customer satisfaction (P1) refers to the impact that the implementation of the feature will have upon the satisfaction of the company's direct customer(s). In a similar vein, reusability (P28) refers to what extent a feature can be developed by reusing existing code or can be reused in the future. Core value propositions and corresponding descriptions are detailed in Appendix D (see last column, 'Core/Specific').

Often, core value propositions denoted general aspects, such as product quality (P11) or overall development costs (P27), which were further developed through specific value propositions. For example, costs were considered in both general and specific ways. Many times interviewees referred to development costs in general. For example, I2 (Company A) mentioned '*then of course I have to take into account what are the development costs*', and I14 (Company B) '*I provide development price and cost, totally how it looks like from a financial point of view*'. Similarly, I19 (Company C) explained that '*for the overall decision, and to understand really the investment levels we are talking about, we get, at least a high level estimate, covering of course the entire solution, what's gonna be the R&D investment*'. However, specific costs were also mentioned in each company case (see Section 4.3), driven by different contextual aspects such as product's particularities, software development approach or the product business model (see Section 4.6).

4.3 Specific value propositions

We labelled as Specific value propositions any propositions that were not common to all three Company cases. Specific here relates to context-specific or company-specific. Twenty-three specific value propositions emerged from our analysis (see Appendix D).

For example, specific value propositions in case A are *P6 – Overall customer solution*, *P42 - Technical relevance* and *P25 – Time-to-profit*. The rationale behind these propositions is as follows: Company A offers a wide solution to its customers, to which Product A is only one piece of such solution. Therefore, Product A's customers usually do not buy only Product A but also other related products. The impact that the implementation of a feature in Product A will have upon the overall solution offered to customers is important in this case (P6). Further, Product A's customers frequently request features that are relevant in other company's products (of the overall solution) but technically irrelevant for Product A. Assessing the technical relevance of features requested by customers is important in this context (P42). In addition, Company A has a specific policy to control the period when the company starts making profit out of a feature, which needs to be considered when selecting features as well (P25).

Another example of specific value proposition, in Case B this time, is *P41 - Certifications*. Product B is highly dependent of security and safety regulations. Assessing the impact that implementing a feature would have upon those certifications is very important for Product B. Examples of specific value propositions for Case C include *P8 – Relevance for target segment*, *P7 - Customer partnership*, *P20 - Competitive delivery channel* and *P36 – Delivery costs*. Related to P8, Company C works a lot with segmentation models. Therefore, a feature that is relevant for Product C's target segment provides higher value. In addition, contrarily to Product A, which is a direct part of Company A's customers businesses, Product C is not an essential part of Company C's customers businesses. Thus, Company C needs to convince their customers about the value that Product C offers to end-users, so that customers are willing to sell Product C to its customers (Product C's end-users), instead of products from competitors. Because of this situation, the impacts that a feature will have upon improving customer partnership (P7) and making the delivery of Product C more competitive (P20) and less costly (P36) are important when selecting features for Product C.

As described in Section 4.2, some specific value propositions can be seen as specializations of core value propositions. We found this situation in two aspects: *costs* and *product quality*. Regarding the value dimension *Cost efficiency*, details on certain costs were case-specific. For example, costs associated to hardware are relevant in Cases A and B, but do not make sense in Case C, which develops a desktop application (e.g. *P34 – Material costs*). Another example can be found in Case B, where costs are frequently caused by the need of special development infrastructures to meet customer's security requirements (P35), as described by I16, '*we have a recent case where one of these kind of classified features was sold to customer, or at least offered to customer. We identified straight away that our normal work environment is not enough for that feature. It needed high security facilities and that's, of course, our internal investment*'. Related to product quality (P11), under the value dimension *Market competitiveness*, although some quality aspects were common to the three cases (product performance (P12) and user-experience - product simplicity (P13)), other quality factors were very case specific and referred to what 'quality' was considered to be in each specific company and product (e.g. product ubiquity (P14), product security (P16), protection promise (P17), feature visibility (P18) and end-user learning (P19))).

4.4 Type of feature

Our analysis revealed that the choice of value propositions differed depending on the type of feature being assessed. Some features are simply needed because if they were not implemented, the product would not work. The decision whether to select such features is rather straightforward

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

and value propositions that are commonly used when selecting features are not in fact applied in these cases. Instead, one main value proposition drives the whole decision (e.g. customer satisfaction, implementation complexity). In particular, we classified features into two categories, 1) mandatory and 2) non-mandatory features.

1) Mandatory features are features that must be chosen for the release. One main value proposition makes the feature mandatory (e.g. the decision is obvious based on that single aspect). We identified 4 types of mandatory features:

1.1) Basic customer features are features considered elementary by customers, who assume these features are part of the product. The Customer value dimension/customer satisfaction proposition drives the decision for this kind of features. For example, I1 (Company A) explained that '*sometimes, for example, you have these very basic things that I mentioned first. But, maybe they* [customers] *don't even think about it. But I know they need this feature but when I have a discussion with the customer, they don't mention it because they just take it for granted that, of course, you have all these basic things'*. Similarly, I20 (Company C) described that *'sometimes you need to do stuff and features that simply enable you to stay in the business but they don't have any actual growth and added value embedded in them. They are simply, they're like a foundation. They are simply there for defensive strategy for you to be in the market overall […]. Many, many times organizations are fooled to look at the upsides only. And, then, they start to disregard these crucial elementary features that are needed there. It would be like if you were in hotel business and you would consider that toilet paper is so basic and you would make the wrong decision of let's not invest in toilet paper any more. […]. So I don't think your hotel business would do extremely well even if you had some unique gadgets in your hotel room. So you need to take care of the basics*'.

1.2) Business case enablers refer to features that are mandatory because they either enable or disable a whole customer or business case (i.e. features that must be implemented if a business case wants to be achieved/kept). Value propositions such as Lead customer satisfaction (P5) guide the decision for this kind of features. According to I4 (Company A), these features are highly prioritized: *'Sometimes some feature is mandatory. Let's say for example, this one customer wants to* [use our products] *next year* […]*. If they want to utilise our equipment, then, we need to be able to comply with* [certain customer's characteristics]. *Otherwise, they will not be able to utilise our stuff. So, that kind of feature goes to the top. It's like enabler for the whole thing. Actually, there are more than one of that kind of features right now. If I remember correctly there are five*'.

1.3) Technical mandatory features are features that are mandatory from a technical point of view (i.e. the product would not work without that feature). The Technical & architecture dimension commands decision on these features. For example, I1 (Company A), described that in their case '*there are certain interfaces defined, so* [Product A] *has to fit into what is already there […]. We cannot ask the core* [system] *to make adaptations because we have limitations in our* [product]. *So that boils down to that, there are then certain features that are needed* […]. *Otherwise, the product will not work in the* [system] *basically*'.

1.4) Features to meet external regulations. Finally, we found some features that are mandatory in order to meet with certain market's regulations or standards (P41). '*There are certain mechanisms in different countries and certain regulations that we have to provide certain features which we cannot sell. They just have to be there. Sometimes countries or regions like the European Union come up with new regulations where they feel this is important to secure the security of users […]. We have to just follow these regulations because otherwise we cannot anymore sell the product in certain markets*' (I2, Company A).

2) With regard to *non-mandatory features*, they are not viewed as compulsory and, therefore, the decision is more complex than for mandatory features (i.e. assessing the value of such features is less evident) and many of the value propositions presented in Appendix D need to be considered. Non-mandatory features include not only end-user features, but also features generated internally such as technical improvements.

4.5. Stakeholder group

A stakeholder group represents a group of people who can affect or are affected by the selection of features for release planning. Members belonging to the same group share interests and responsibilities when selecting features (e.g. product managers, project managers, etc.). While analyzing the interviews, we noticed that stakeholders that shared responsibilities, tended to also focus on the same value propositions. To systematically carry out our analysis, we arranged key stakeholders into three stakeholder groups, depending on whether their main responsibility was closer to *business* or *development* (see Appendix A, fourth column). We assigned interviewees to the *business group* if their primary duties focused on strategic product decisions, from product roadmaps to feature prioritization (e.g. strategic product managers). Interviewees assigned to the *development group* were those who were close to the product's technical side and dealt with development aspects, such as development capability, planning on how an implementation will be done, quality issues or technical aspects. This stakeholder group included roles such as project managers, quality managers and other technical managers. We assigned roles such as product owners to the third group *business/development,* as they serve as a bridge between business and development in our company cases.

Appendix E provides a detailed analysis of value propositions according to each individual stakeholder and stakeholder group (i.e. from business to development). Some value propositions were mentioned by almost every stakeholder, independently of his/her group (e.g. *P1 - Customer satisfaction* (all), *P10 - Time-to-market* (17/21), *P27 - Overall development costs* (all), *P37 – Implementation complexity* (18/21) and *P39 – Development capability* (17/21)). However, when analyzing the value propositions mentioned by each group in more detail, we noticed that different stakeholder groups emphasized different aspects in the interviews. That is, we identified patterns among stakeholder groups that indicate that depending on the interviewee's role and main responsibilities, s/he tended to pay more attention to different value propositions. For example, business stakeholders (i.e. I1-I3, I11-I13, I19-I21) focused more on value propositions from the *Customer value* dimension (P2 was mentioned by 7/9 stakeholders from the business group and only by 4/8 from the development group. Similarly, ratios for P3, P4 and P5 are 8/9 vs. 3/8, 7/9 vs. 3/8 and 5/9 vs. 2/8 respectively). This is natural as interviewees from this group are in constant contact with customers (to bring customers' voices into the development), company's business intelligence units and strategic product management groups. Further, value propositions from the dimensions *Economic value/profitability* and *Company's strategy* were mainly mentioned by business stakeholders. Indeed, a notable number of stakeholders from the development group did not mention value propositions from these dimensions at all (four stakeholders in each case).

On the other hand, stakeholders belonging to the development group (i.e. I5-I9, I16-I18) provided more details on the dimensions *Cost efficiency* and *Technology & architecture*. For example, testing costs (P30-P32) were mainly mentioned by technical stakeholders, particularly I8 and I9. Similarly, *P34 – Material costs* and *P35 – Development infrastructure costs* were also pointed out by technical stakeholders or stakeholders belonging to the business/dev. group. However, it is interesting to notice that costs on *Third-party products (P29)* and *Maintenance (P33)* were mentioned by business stakeholders. The impact that the implementation of a feature will have upon *Product architecture (P38)* and *Development capability (P39)* had a particular focus in the development group as well.

Finally, related to the middle-way group, representing a link between business and development (I10, I14, I4 and I15), we could not find a clear tendency. It seems that their value propositions are distributed along the six value dimensions, although a particular focus can be observed on the *Customer value* and *Technology & architecture* dimensions. This may be explained by the fact that as they are in constant contact with both groups, business and development stakeholders (e.g. product owners), they tend to have a more balance view of value propositions.

4.6 Company Context

Company context refers to the conditions, environment and settings in which a company operates and, therefore, characterize it (enabling and/or constraining it) [71]. During our analysis, we noticed that value propositions that we categorized as specific related to particular aspects of the product, the company's business model or the approach used by the company to develop their product. Concretely, we identified the following patterns:

Product's specific characteristics. Value propositions were different between embedded systems (Products A and B) and pure software solutions (Product C). For example, value propositions related to the hardware side of the product were only considered for Products A and B (e.g. P34 and P40). I2 and I5 (Company A) referred to this aspect in the following extracts from the interviews: '*we have to give feedback what kind of capabilities the hardware has to have to serve a certain software feature*' (I2); '*if the customer is just asking something that your hardware platform is not supporting… you just cannot make whatever you want if it's just not sitting in that platform*' (I5). Similarly, I17 (Company B) commented '*currently we have one this kind of very good example, where we are implementing or improving one of our features. At the same time, we have to make quite huge modification to hardware and the decision that shall we make this or not. I have said that yes, we have to do that, but we have to manage the schedule because we have huge material stock. It's the expensive part, it doesn't make sense to scrap all those*'. In addition, our analysis suggested that testing, specifically integration and validation testing, is particularly demanding in software intensive systems, being an aspect considered when selecting features. According to our interviewees, testing activities need to consider both software and hardware aspects, and expensive testing infrastructures are needed in this context. Testing costs were explicitly mentioned when assessing the value of a feature in Cases A and B (e.g. P30, P31, P32). Integration & verification costs (P31) and costs in testing equipment (P31) were emphasized in Case A.

Value propositions related to quality were also dependent on product characteristics (e.g. P14, P16, P17). Certain product aspects, such as Product B's security needs and end-user difficulties in understanding Product C's functionality, gave room to specific value propositions, such as *P35 – Development infrastructure costs* and *P41 – Certifications* (in Product B), and *P18 – Feature visibility* and *P19 - End-user learning* (in Product C).

Business model. Some specific value propositions have their roots in aspects related to our cases' business model. Our three cases operate in a business-to-business (B2B) setting. However, as mentioned in Section 4.3, the product distribution is different in each company case, thus driving company-specific value propositions. Company A seldom sells Product A in isolation but as part of a wider customer solution. *Overall customer solution (P6)*, considering also other Company A's products, is assessed when selecting features for Product A. Moreover, evaluating a feature's *technical relevance (P42)* is important when assessing its value, because customers usually request features that are relevant in other company products but do not technically make sense in Product A. In the case of Company C, the company has specific challenges when distributing Product C due to customers' difficulties in understanding security principles. Thus, educating customers about the importance of security, as a business, is important for Company C (P7). Moreover, given the amount of customers that use Product C (over 200), *reducing delivery costs (P36)* and *making the delivery channel as competitive as possible (P20)* are essential for Company C's business.

Software product development process. The type of development process followed by a company also drove some specific value propositions. For example, Case C's emphasis on market segmentation and creating products for a certain market niche led *P8 - Relevance for target segment*, being frequently mentioned (particularly by I21). Similarly, the specific value proposition *P25 – Time to profit*, is a result of a Company A's internal policy.

As a summary, Table 5 provides an overview of the theory including its constructs, relationships, instantiations, and scope. A critical evaluation, according to Sjøberg et

TABLE 5

SUMMARY OF CONSTRUCTS, RELATIONSHIPS, INSTANTIATIONS AND SCOPE OF THE THEORY

<table>
<tr><td colspan="2"><b>Theory Overview</b>
The theory of value for value-based feature selection for release planning proposes that value for feature selection (software features) is characterized by a set of value propositions that can be <i>core value propositions</i> or <i>specific value propositions</i>. Specific value propositions are dependent on <i>company contextual aspects</i> such as the type of product being developed. Moreover, value propositions vary according to the <i>stakeholder group</i> that assesses the value of a feature and the <i>type of feature</i> being assessed.</td></tr>
<tr><td colspan="2"><b>Theory Components</b></td></tr>
<tr><td><b>1. Constructs</b></td><td><b>Instantiation (example)</b></td></tr>
<tr><td><b><i>C1. Value proposition</i></b>: any aspect, criterion or consideration that is used by key stakeholders when deciding whether a software feature should be included in a release in the context of value-based decision-making.</td><td>Time-to-market, ROI, Product security, Hardware costs.</td></tr>
<tr><td><b><i>C2. Core value propositions</i></b>: value propositions used by all three Company cases and that seem to be commonly used when selecting software features (represent a pattern).</td><td>Customer satisfaction, Time-to-market, ROI, Overall development cost.</td></tr>
<tr><td><b><i>C3. Specific value propositions</i></b>: case/context-specific value propositions.</td><td>Feature visibility, Product security, Hardware costs.</td></tr>
<tr><td><b><i>C4. Type of feature</i></b>: e.g. mandatory and non-mandatory feature.</td><td>Basic customer feature, Technical mandatory feature.</td></tr>
<tr><td><b><i>C5. Stakeholder group</i></b>: group of stakeholders that share interests and responsibilities when selecting features.</td><td>Business managers, Product owners, Technical stakeholders.</td></tr>
<tr><td><b><i>C6. Company context</i></b>: contextual factors that characterize the company such as business domain and type of product under development.</td><td>Software intensive system, Telecommunications domain.</td></tr>
<tr><td><b>2. Relationships</b></td><td><b>Instantiation</b></td></tr>
<tr><td>R1. Value propositions used to decide upon software features include core value propositions.</td><td>Customer satisfaction is a core value proposition
Time-to-market is a core value proposition.</td></tr>
<tr><td>R2. Value propositions used to decide upon software features include specific value propositions.</td><td>Product security is a specific value proposition for Case B.</td></tr>
<tr><td>R3. Specific value propositions are determined by company contextual factors.</td><td>Hardware costs is a specific value proposition for companies developing embedded systems.</td></tr>
<tr><td>R4. Value propositions used to decide upon software features vary according to the type of feature.</td><td>Value propositions are not consistently considered to decide upon mandatory features.</td></tr>
<tr><td>R5. Value propositions used to decide upon software features vary according to the stakeholder group making the feature's assessment.</td><td>Testing costs are relevant for technical stakeholders but are not commonly considered by business managers.</td></tr>
<tr><td colspan="2"><b>3. Scope</b> (Sjøberg et al. [21])
Scope of interest: value-based feature selection for release planning in the context software and software intensive products.
Scope of validity: value-based feature selection for release planning in companies similar to the companies of this study.</td></tr>
</table>

al.'s [21] six criteria for evaluating empirically-based theories in SE (testability, empirical support, explanatory power, parsimony, generality and utility) is also provided in Appendix F.

5 DISCUSSION

This section first discusses how our findings answer the RQs posed in Section 1. Then, it discusses our findings in the light of existing literature and presents implications for research and practice. Finally, it discusses on validity.

5.1 Key findings

This section summarizes the key findings of our study.

RQ1: What are the value propositions employed by key stakeholders when making decisions relating to features' selection for release planning? Value, in the context of feature selection, encompasses a wide range of propositions. Appendix D shows the list of value propositions that key stakeholders from our company cases employ when selecting features. We identified 47 value propositions that were distributed in six value dimensions: *customer value*, *market competitiveness*, *economic value/profitability*, *cost efficiency*, *technology & architecture* and *company strategy*. The Customer value dimension was by far the most frequently mentioned in our interviews (all stakeholders made reference to some Customer value proposition, and 387 instances were coded in NVivo). The value dimensions *Company strategy* and, surprisingly, *Economic value/profitability* were the least mentioned (15 and 16 stakeholders, and 98 and 120 instances in NVivo, respectively). However, the amount of empirical data that supports these dimensions justifies their importance. From the 47 propositions identified in the study, 24 were common to all three companies. This finding suggests that stakeholders from different companies share a similar understanding on a significant number of propositions to be considered when assessing a feature's value. These common value propositions find also support in the related literature (see Section 5.2).

RQ2. What are the overlapping and distinct value propositions between different stakeholders involved in value-based feature selection? Many value propositions overlap among stakeholder groups and individual stakeholders; particularly in the case of core value propositions (see Appendix E). 33 value propositions (70%) were mentioned by the three stakeholder groups (by at least one stakeholder per group) – 87% in the case of core value propositions (21/24) and 52% in the case of specific value propositions (12/23). Moreover, five value proposition were mentioned by more than 80% stakeholders: *P1 Customer satisfaction* and *P27 Overall development costs*, which were mentioned by all stakeholders (21/21), and *P10 Time-to-market*, *P37 Implementation complexity* and *P39 Development capability* that were mentioned by 17, 18 and 17 stakeholders respectively. These findings suggest that stakeholders from different groups share many propositions when assessing the value of a feature. However, there were also 14 value propositions that were ignored by some stakeholder groups (3 core and 11 specific). Stakeholders from the business group did not mention specific costs (such as *P31 I&V costs*, *P32 Testing equipment costs* and *P35 Development infrastructure costs*) and very seldom mentioned propositions such as *P34 Material costs*. In the case of the development group, propositions related to the *Customer value* dimension such as P6 and P7 were ignored and propositions such as P3-P5 were very seldom mentioned. Similarly, this group did not mention propositions P25 (*Economic value dimensio*n) and P44 (*Company's strategy dimension*) and very seldom mentioned propositions P21-P24, and P36 and P45-P47 from these dimensions. Eight propositions were overlooked by the middle-way group (e.g. P23, P29, P41, P53), which were fairly distributed among value dimensions.

RQ3. What are the value propositions particular to a stakeholder's group? We also identified differences between stakeholder groups in regard to their main focus when selecting features. As described in Section 4.5, stakeholders from the business group tended to focus more on value propositions from the *Customer Value*, *Market competitiveness, Economic value/profitability* and *Company's strategic* dimensions, while stakeholders from the development group were disposed to *Cost efficiency* and *Technology & architecture* dimensions. Stakeholders that are a bridge between business and development (e.g. product owners) had a balanced view among value dimensions. Although only four value propositions were exclusive to a stakeholder group (P14, P25, P36, P44), several value propositions were mainly mentioned by a single group. For example, P6, P7 and P8 (belonging to the *Customer value* dimension) were specific for business stakeholders (or the middle group). Similarly, detailed costs under the *Cost efficiency* dimension, such as P31, P32 and P35, were specific to the development group (or the middle group). This result suggests that decision makers from different stakeholder groups provide complementary views that altogether form the value of a feature. Therefore, if a particular group would not be part of the feature selection team, there would be a high risk that some value propositions (particularly specific value propositions) would be overlooked.

RQ4. Are there any distinct value propositions between different types of features? We found that value propositions employed to decide upon software features are considered differently in the case of mandatory and non-mandatory features. For mandatory features (e.g. basic customer features or technical mandatory features) discussions to assess their value are based on a main value proposition that makes the decision evident (e.g. customer satisfaction, regulations). Stakeholders know that those features need to be implemented, even if the value that they provide is limited to one proposition. Therefore, not all value propositions are thoughtfully considered during those discussions as the value proposition that makes the feature mandatory is enough to make the decision.

RQ5. Are there any overlapping and distinct value propositions between different company contexts such as business domains or type of products? We found that many value propositions (24) were common to the three company cases, which suggests a trend in the use of value propositions that could be commonly applicable when deciding upon features. However, we also found that many other propositions were case or context specific (23). A detailed analysis of those propositions revealed that they emerged from particularities of the company context. Product, process and business characteristics are the main contextual factors influencing specific value propositions in our company cases. There may be other contextual factors that impact on specific value propositions, and this is a possible direction for extending the scope of our theory.

5.2 Theoretical integration

Table 6 presents the ways in which our research relates to existing literature (i.e. empirical studies on value propositions for requirement/feature selection). Our theory extends existing research in several ways:

First, our theory extents previous research by postulating a theory that assembles important pieces for value-based feature selection. Our theory classifies value propositions into six value dimensions, which include not only a business view (i.e. customer value, market competitiveness, economic value/profitability and company strategy) but also a project/technical focus (e.g. cost efficiency and technology & architecture). Further, our theory categorizes value propositions into core and specific propositions, and relates value propositions to stakeholder group, type of feature and company's context.

Second, our findings confirm previous survey studies' insights towards a consistency in value propositions applied when selecting features for release planning [9] [13]. Indeed, 71% of our core value propositions (17) are also mentioned in related empirical studies (see Appendix G), which further legitimize their importance (e.g. customer satisfaction, competitiveness, time-to-market, development costs and implementation complexity). We extend the knowledge on core value propositions by adding seven new core value propositions that were not considered by previous empirical research but emerged in our study, such as number of customers, lead customer satisfaction, customer base/market share, reusability and company's brand. Further research is however needed to confirm or refuse the importance of those propositions as core.

Third, almost half of the value propositions identified in our study were context-specific, which suggests that using the same value propositions across different companies is not always the norm. Thus, our findings provide a word of caution when creating check-lists for assessing value and trying to aggregate value propositions through secondary studies, as context needs to be properly considered (not all value propositions are applicable for every case and in every kind of decision). Recent work by other researchers points out also into this direction (e.g. [72]).

Fourth, our theory related specific value propositions to company context and, particularly, product characteristics, business model and software development process. Barney et al. [14-16] also mentioned that the context in which a software product exits might affect value propositions used in the software domain. Their particular focus was on

TABLE 6

SUMMARY OF THE WAYS IN WHICH THE THEORY RELATES TO EXISTING RESEARCH (THEORETICAL INTEGRATION)

Overall comment: The grounded theory of value for value-based feature selection in software/software-intensive products generated in this research is a first of its kind in the field. It assembles together pieces to be considered when deciding the value of a feature.

Theory component	**Comment on how the theory relates to existing literature**	**Comment on how the theory extends existing literature**
Value proposition	Previous empirical studies on value propositions: 1. Survey studies ([9], [13-18]): • 13 predefined value propositions (9 emerged in our study and 4 were not mentioned by our interviewees[4]). • A business over a project/product perspective. • Respondents proposed additional criteria, which suggests that predefined survey criteria were not enough to define value in the respondents' contexts. 2. Interview-based studies: • Alahyri et al. [10] stresses a project focus on short-term value propositions (interviewees: process responsible and product owners). • Rodríguez et al. [5] presents a detailed description of value propositions for Case A. 3. Secondary studies: the Value Software Map (VSM) [3] classifies value propositions into 62 value components. Generic classification (any decision/any product).	• It classifies value propositions for feature selection into six dimensions (i.e. customer value, market competitiveness, economic value/profitability, cost efficiency, technology & architecture and company's strategy), including business and project/technical aspects. • It empirically identifies 27 new value propositions – 7 core and 20 specific (e.g. # of customers, overall customer solution, feature visibility, opportunity cost) – see Appendix G. • It provides deeper analysis of value propositions compared to related work (particularly survey studies) and provides concrete definitions of value propositions (see Appendix D). • The focus on feature selection makes elicited value propositions more specific, with clear use cases compared to related work (e.g. secondary studies).
Core value proposition	• [9] and [13] discerned the possibility of a pattern on value propositions. • Many value propositions mentioned in previous research are part of our core value propositions (e.g. customer satisfaction, competitiveness, time-to-market, development cost, and implementation complexity).	• It confirms a patter on value propositions (core value propositions). • It extends it by showing the nature of core value propositions. • It provides new core value propositions (e.g. # of customers, lead customer satisfaction).
Specific value proposition	-	• Specific value propositions have not been discussed in the literature. • We theorized that specific value propositions will be common across product type, but more work is needed. Examples of specific value propositions are presented in Appendix D. • It indicates that a pattern on value propositions can be only partially achieved.
Type of feature	-	• The influence of type of feature on value propositions has not been recognized in existing literature. • It relates value propositions to type of feature (mandatory vs. non-mandatory feature).
Stakeholder group	• The 4+1 theory of VBSE by Boehm and Jain [58], which centers in the win-win Theory W (an enterprise will succeed if and only if it makes winners of its success-critical stakeholders) • Rodríguez et al. [5] presents a detailed per-stakeholder group analysis for Case A.	• It confirms that value propositions differ among stakeholder groups (win-win theory). • It extends it by providing a detailed per-stakeholder analysis (e.g. stakeholder groups have a similar understanding on main propositions but a distinct primary focus).
Company context	Barney et al. [14-16], mentioned that "*other factors might influence the criteria used to select and prioritize release requirements— for example, product maturity, requirement source, customer type, contract type, and size of the customer base. These factors require further study*".	• It relates value propositions to company context and, particularly, product characteristics, company's business model and software development process.

product maturity (i.e. different value propositions are used depending on the product maturity). We found some insights in this direction as well. For example, I19 (Company A) mentioned that '*Product C, is pretty much a well-established product. But, then, when we look at* [other Company C's products], *they are completely new businesses[…], they will have different metrics, like subscriber acquisition, downloads, conversion rates and so forth, while, of course,* [Product C] *is an existing established business so we would have a bit different metrics, driving revenues and minimizing turns, those are very clear targets for us*'. However, the frequency of occurrences related to this category was not enough to be included in our theory. Future work on this direction is needed.

Fifth, our theory relates value propositions with type of feature. To the best of our knowledge, this new aspect has not been mentioned by previous empirical studies on value propositions for feature/requirement selection.

Sixth, our theory adds several new value propositions. Besides the seven new core value propositions mentioned in the second item of this list, most of our specific value propositions have not been mentioned in related empirical studies (20/23). See Appendix G for details.

5.3 Implications for research and practice

***Implications for researchers and future work*:** theory building in an applied discipline such as SE is a continuous and

[4] A more detailed analysis of value propositions identified as relevant in these studies can be found in Appendix G.

iterative process of proposing, testing, and modifying theories [21]. The constructs and relationships that compose our theory open areas for future research to help extend and refine this theory. Categories that depend on contextual aspects such as type of product/business domain will benefit from additional cases that extend their scope. Core value propositions for specific domains could be discovered when analyzing and integrating cases from similar domains. Additional theoretical sampling may also reveal that some value propositions that we classified as core (since they appeared in our three cases) are specific value propositions impacted by contextual factors that did not emerge in our interviews. Any future work could also look at the associations in more detail and explore causalities. Glaser advances some 42 'coding families' to help researchers find associations between categories [73], and the types of associations possible are accordingly many and varied. We took care to 'ground' the associations in the data, as described by Urquhart [27], where we made sure that each association presented in the diagram has firm empirics at its base. However, we did not focus on causality in our work. Associations between concepts could be further elaborated on to identify causalities that could perhaps be tested in the future using different research methods such as formal experiments. In addition, the theory could also be extended by involving stakeholder groups that have not participated in this study, such as customers.

A common use of GTM is to generate concepts for large scale surveys. Thus, a survey might be a future route for testing our emergent theory as well. That said, based on our experiences in this study, we believe that context-specific research that allows one to understand the details of value in concrete contexts, should play a prominent role to advance the area. In our sister discipline of IS, there have been calls for more specific context based research, based on the idea that in a rush to universalism, valuable context is lost [74]. Contextualization makes research models more accurate and the interpretation of results more robust [75]. Although our study shows a pattern across our company cases (core value propositions), aiming to have a complete, well-defined taxonomy of universally applied value propositions may be challenging (as suggested by our findings, where many value propositions are context dependent). Our interviewees also reflected on this aspect. *'it's very difficult to come up with like a template of these are always the factors to look into. Yeah there's probably a hundred criteria there but depending on the type of the feature they obviously have totally different weight*' (I20, Company C). In this sense, it is also important that researchers clearly define the context of the studied company cases to facilitate the task of identifying commonalities and differences between contexts.

***Implications for practitioners*:** The most important takeaways of this research for practitioners are as follows: when deciding upon the value of a feature, practitioners need to consider the set of value propositions that make sense in their own context since a significant number of value propositions are context dependent (specific value propositions). Therefore, it is likely that value propositions that are relevant for some are irrelevant for others. Similarly, value propositions that are important for some firms will not appear anywhere else. That said, some value propositions seem to be commonly applicable (i.e. core value propositions). Our ultimate aim is to provide systematic support for addressing value concerns in value-based feature selection for release planning. Practitioners can employ our theory as a tool to understand what governs their decisions and the way in which they apply value in their companies. Particularly, propositions that emerged as core value propositions in our research can be used as a checklist for decision makers reflecting on value for feature selection. Value dimensions, as more abstract level entities, could also be considered when reasoning what value means in a specific context. This may be useful for the many companies that use agile and lean approaches where value is also a key consideration. Companies that share contextual factors with our companies may also benefit more widely from the specific value propositions identified in this research. In addition, our findings, showing that different stakeholder groups tend to focus on different value dimensions, suggest that it is important that decision making teams include a wide range of stakeholders. Varied stakeholders can bring different perspectives that need to be considered when deciding upon the value of a feature, so that important aspects, particularly specific value propositions, are not overlooked.

5.4 Validity Discussion

This section is based on the validity criteria set out by Yin [60] in his seminal book on case study research and his suggested validation tactics. We also considered the criteria proposed by Maxwell [76] for qualitative research. Further details on validity are provided in Appendix H.

Theoretical validity - Construct validity. In building theories from qualitative research, construct validity closely matches the first aspect considered by Maxwell for theoretical validity: the validity of concepts [76]. According to Maxwell, this validity depends on whether there is consensus within the community about the terms used to characterize a phenomenon. We aimed to employ terms widely used in SE (e.g. stakeholder). However, whenever there was lack of consistency in their use, we employed the terminology used in VBSE (e.g. value proposition). According to Yin [60], construct validity can be increased by: first data triangulation (using multiple sources of evidence – in our case multiple stakeholders), secondly by establishing a chain of evidence, something that is an advantage of GTM due to the coding procedure, and thirdly to have the findings reviewed by key informants, which was also a feature of our research design. In addition, we had preparatory meetings with company champions to minimize misconceptions between researchers and practitioners. Moreover, construct validity was supported by theoretical saturation as the most important constructs in our theory were 'saturated', i.e. had many instances in the data.

Theoretical validity - Internal validity. Yin [60] points out that internal validity is a concern only for causal or explanatory case studies, and we have been at pains to say that we identified associations in our emergent theory, not causation. If we think of internal validity as an issue of inference, then it closely matches the second aspect considered by Maxwell for theoretical validity: the validity of relationships [76]. In this sense, it is important that all rival explanations have been considered, that the research evidence is convergent. Triangulation within our research design has assisted with these concerns. Theory triangulation was practiced by dint of multiple models being built from the coding, before the final emergent theory was both challenged by the coauthors and the associations tested by ensuring they could be grounded in the data. Participant triangulation encompassed the entire population involved in feature selection in Cases A and C, and covered a representative group of those in Case B. Our interviewees had a wide experience in software development and value-based feature selection tasks, which mitigates interviewee's inadequate view on value due to a lack of expertise.

Reliability. Yin [60] notes that subsequent researchers should be able to follow the same steps and arrive at the same findings. It is our hope that the methodology outlined in this paper is sufficiently detailed for other researchers to follow. Closely related to reliability, Maxwell [76] refers to *descriptive validity*, as descriptive accuracy, and *interpretive validity*, concerned to accuracy from a participants' perspective. Regarding the former, all our interviews were audio recorded and transcribed. Investigator triangulation was also applied in all phases of our research. Two investigators were present at every interview, which ensured that standard questions were adhered to, and that the meaning ascribed by one investigator when coding was able to be challenged because both investigators were present (to minimize biased views). Moreover, both investigators (A1 and A3) individually analyzed interview I4 (the first conducted in a chronological order) and compared the results to check their level of agreement (open coding). The level of agreement was high 75% (see Appendix H). A1 analyzed the remaining interviews and A3 reviewed each analysis prior to delivering for validation. Emerging themes and categories were also extensively discussed among the three authors to ensure reliable selective and theoretical coding and clarity in constructs and propositions. Related to interpretive validity, the findings were also checked with the respondents to ensure we were representing their feature selection practices accurately.

External validity - Generalizability. As previously stated, theories from cases generalize to theory rather than to a population [60] (analytical generalization vs. statistical generalization). Thus, rather than making a Level 1 inference about a population on the basis of empirical data collected about a sample, we make a Level 2 inference where a previously developed theory is used as a template with which to compare the empirical results of the cases [60]. This process is put forward in Section 5.2 and Table 6, which demonstrates the path towards generalization to a theory. In addition, the varying roles and wide experience of those involved in feature selection enabled us to cover different decision-making perspectives. We believe that our findings can be generalized to other companies sharing similar contexts to those detailed herein. Still, it is important to note that, as described in Section 5.3, the theory put forward in this paper can have its scope extended as additional cases are analyzed and integrated. To facilitate this task, Appendix A in [5] details the steps to integrate additional cases.

6 Conclusions

Numerous software companies embrace today the paradigm shift led by the VBSE agenda, placing value at center stage in the SE landscape. However, there is little empirical evidence to explain what value means in practice. The broad aim of this study was to investigate value in the context of value-based feature selection for release planning in software and software-intensive products. Through a Grounded Theory study, involving 21 key stakeholders from three software/software-intensive companies, we identified that value propositions are characterized by:

1. *Core value propositions*: value propositions commonly applied when selecting software features.
2. *Specific value propositions*: context-specific value propositions.

Further, value propositions are comprised in function of

3. *Stakeholder group*: different stakeholder groups have a distinct primary focus when assessing the value of a feature.
4. *Type of feature*: value propositions are differently considered when assessing the value of features that are mandatory (e.g. feature to meet a regulation) compared to non-mandatory features.
5. *Company context*: value propositions are influence by contextual factors, particularly, product characteristics, company's business model and software development process.

We presented a detailed analysis of value for value-based feature selection, identifying also 27 new value propositions (20 specific and 7 core) that have not been recognized in the related empirical literature. Understanding the nature of value for feature selection will help software companies, and their decision making teams, better comprehend value and provide them with a tool to improve their value-based decision making. Future work comprises extending the theory by integrating additional cases.

Acknowledgments

We would like to thank the companies that collaborated with us for its commitment and in-kind time. This research has been carried out within the FiDiPro project number 40150/14, which was funded by Tekes (the Finnish Funding Agency for Technology and Innovation).

References

[1] B. Boehm, "Value-Based Software Engineering: Reinventing "Earned Value" Monitoring and Control, SIGSOFT Software Engineering Notes, vol. 28, no 2, p. 3, 2003.

[2] S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Grünbacher, "Value-Based Software Engineering", Springer-Verlag New York, Inc, 2005.

[3] M. Khurum, T. Gorschek, and M. Wilson, "The software value map - an exhaustive collection of value aspects for the development of software intensive products", J. of Software: Evolution and Process, vol. 25, no 7, p. 711-741, 2012.

[4] E. Mendes, P. Rodriguez, V. Freitas, S. Baker, and M.A. Atoui, “Towards improving decision making and estimating the value of decisions in value-based software engineering: the VALUE framework”, Software Quality Journal, p. 1-50, 2017.

[5] P. Rodríguez, E. Mendes, B. Turhan, “Key Stakeholders’ Value Propositions for Feature Selection in Software-intensive Products: An Industrial Case Study”, TSE, 2018.

[6] M. Lindgren, R. Land, C. Norström, and A. Wall, “Towards a Capability Model for the Software Release Planning Process - Based on a Multiple Industrial Case Study”, In Proc. 9th International Conference on Product Focused Software Process Improvement, 2008b.

[7] N. Riegel, “Guiding requirements elicitation using a prioritization framework”, In: REFSQ 2013 Workshop Proceedings, pp. 133–144, 2013.

[8] N. Riegel, and J. Doerr, “An analysis of priority-based decision heuristics for optimizing elicitation efficiency”, In: Salinesi, C., van de Weerd, I. (eds.) REFSQ 2014. LNCS, vol. 8396, pp. 268–284. Springer, Heidelberg, 2014.

[9] C. Wohlin, and A. Aurum, "Criteria for selecting software requirements to create product value: An industrial empirical study." Value-based Software Engineering. Springer Berlin Heidelberg, 179-200, 2005a.

[10] H. Alahyari, R. B. Svensson, and T. Gorschek, “A study of value in agile software development organizations”, Journal of Systems and Software, 125, 271-288, 2017.

[11] P. Ojala, “Implementing a value-based approach to software assessment and improvement”, PhD Thesis, University of Oulu, 2006.

[12] E. Day, and M.R. Crask, “Value assessment: the antecedent of customer satisfaction”, Journal of consumer satisfaction dissatisfaction and complaining behavior, 13, 52-60, 2000.

[13] C. Wohlin, and A. Aurum, “What is important when deciding to include a software requirement in a project or release?” In 2005 International Symposium on Empirical Software Engineering, IEEE, 2005b.

[14] S. Barney, A. Aurum and C. Wohlin, "Quest for a silver bullet: Creating software product value through requirements selection", 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA'06, 2006.

[15] S. Barney, C. Wohlin, G. Hu and A. Aurum, “Creating software product value in China”, Software, IEEE, 26(4), 84-90, 2009.

[16] S. Barney, A. Aurum, C. Wohlin, “A product management challenge: Creating software product value through requirements selection”, Journal of Systems Architecture, 54(6), 576-93, 2008.

[17] G. Hu, A. Aurum, and C. Wohlin, “Adding Value to Software Requirements: An Empirical Study in the Chinese Software Industry”. ACIS 2006 Proceedings, 7, 2006.

[18] T. J. Latha, and L. Suganthi, “An empirical study on creating software product value in India–an analytic hierarchy process approach”. International Journal of Business Information Systems, 18(1), 26-43, 2015.

[19] N. Salleh, F. Mendes, and E. Mendes, ”A Systematic Mapping Study of Value-based Software Engineering”, Euromicro 2019.

[20] K.J. Stol, P. Ralph, and B. Fitzgerald, “Grounded theory in software engineering research: a critical review and guidelines”, In Proceedings of the 38th International Conference on Software Engineering, pp.120-131, ACM, 2016.

[21] D. I. Sjøberg, T. Dybå, B. C. Anda, and J. E. Hannay, “Building theories in software engineering”, In guide to advanced empirical software engineering (pp. 312-336). Springer, 2008.

[22] P. Johnson, M. Ekstedt, and I. Jacobson, "Where's the theory for software engineering?" IEEE software 29.5: 96, 2012.

[23] R. Wieringa, and M. Daneva, "Six strategies for generalizing software engineering theories." Science of computer programming 101, pp. 136-152, 2015.

[24] C. Wohlin, D. Šmite, and N. B. Moe, “A general theory of software engineering: Balancing human, social and organizational capitals”, Journal of Systems and Software, 109, 229-242, 2015.

[25] P. Johnson, and M. Ekstedt, “The Tarpit–A general theory of software engineering”, Information and Software Technology, 70, 181-203, 2016.

[26] SEMAT initiative, www.semat.org, last accessed: 11.01.2020

[27] C. Urquhart, “Grounded Theory for Qualitative Research: A Practical Guide”, Sage, 2013.

[28] S. Adolph, W. Hall, and P. Kruchten, “Using grounded theory to study the experience of software development”, Empirical Software Engineering, 16 (4), pp.487-513, 2011.

[29] R. Hoda, J. Noble, and S. Marshall, “Self-organizing roles on agile software development teams”, IEEE Trans Softw Eng, 39 (3), pp.422-444, 2013.

[30] T. Sedano, P. Ralph, and C. Péraire, “Software development waste”, 39th International Conference on Software Engineering, ICSE, 2017.

[31] K. M. Eisenhardt, "Building theories from case study research", Academy of management review. 14(4), pp. 532-550, 1989.

[32] A. McCallin, "Grappling with the literature in a grounded theory study." Contemporary Nurse 15.1-2 pp: 61-69, 2003.

[33] C. Urquhart and W. Fernandez, "Using grounded theory method in information systems: the researcher as blank slate and other myths", Journal of Information Technology, 28(3), pp. 224-236, 2013.

[34] B. Boehm, “Software Engineering economics”, vol. 197. Englewood Cliffs (NJ): Prentice-hall, 1981.

[35] M.E. Porter ME, “Competitive advantage”. New York, Free Press, 1985.

[36] A. Weinstein and W.C. Johnson, “Designing and delivering superior customer value: concepts, cases, and applications”, CRC press, 1999.

[37] M. Usman, E. Mendes, F. Weidt, and R. Britto, “Effort estimation in agile software development: a systematic literature review”. PROMISE, pp: 82-91, 2014.

[38] M. Jorgensen, and M.A. Shepperd, “Systematic Review of Software Development Cost Estimation Studies”, IEEE Transactions on Software Engineering, Vol. 33, No. 1, 2007.

[39] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula, “Survey on agile and lean usage in Finnish software industry”, In ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 139-148, 2012.

[40] M. Poppendieck, and T. Poppendieck, “Lean Software Development: An Agile Toolkit”, Addison-Wesley, 2003.

[41] P. Rodríguez, J. Partanen, P. Kuvaja, and M. Oivo, “Combining lean thinking and agile methods for software development: A case study of a Finnish provider of wireless embedded systems”, in 47th Hawaii International Conference on System Sciences, pp. 4770-4779, 2014.

[42] M. Asadi, E. Bagheri, B. Mohabbati, and D. Gašević, “Requirements engineering in feature-oriented software product lines: an initial analytical study”, in 16th International Software Product Line Conference, pp. 36-44, 2012.

[43] D. Greer, and G. Ruhe, “Software release planning: an evolutionary and iterative approach”. Information and software technology, 46(4), 243-253, 2004.

[44] M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal, “Software product release planning through optimization and what-if analysis”. Information and Software Technology, 50(1), 101-111, 2008.

[45] M. Svahnberg, T. Gorschek, R. Feldt, R. Torkar, S.B. Saleem, and M.U. Shafique, “A systematic review on strategic release planning models”. Information and software technology, 52(3), 237-248, 2010.

[46] P. Achimugu, A. Selamat, R. Ibrahim and M.N.R. Mahrin, “A systematic literature review of software requirements prioritization research”, Information and Software Technology, 56(6), pp.568-585, 2014.

[47] A.G. Dahlstedt, and A. Persson, “Requirements interdependencies-moulding the state of research into a research agenda”. In 9th Int. Workshop on Requirements Engineering: Foundation for Software Quality, pp. 55-64, 2003.

[48] D. Mougouei, “Factoring requirement dependencies in software requirement selection using graphs and integer programming”. In 31st IEEE/ACM International Conference on Automated Software Engineering, pp. 884-887, ACM, 2016.

[49] N. Jan and M. Ibrar, “Systematic Mapping of Value-based Software Engineering – A Systematic Review of Value-based Requirements Engineering”, Masters thesis MSE-2010:40, 2010, Blekinge Institute of Technology, Sweden, 2010.

[50] K. Wnuk and E. Mendes, “The Project Management Perspective on Software Value: A Literature Review”, in Proceedings of KKIO 2015.

[51] B.J Oates, “Researching Information Systems and Computing”, Sage publications, 2006.

[52] S. Gregor, "The nature of theory in information systems", MIS quarterly, pp. 611-642, 2006.

[53] V. Garousi, and M. Mäntylä, “Citations, research topics and active countries in software engineering: A bibliometric study”, Computer Science Review, 19, 56-77, 2016.

[54] V. R. Basili, “The role of experimentation in software engineering: past, current, and future”, Proceedings of the 18th international conference on Software engineering (pp. 442-449). IEEE Computer Society, 1996.

[55] P. Johnson and M. Ekstedt, GUTSE, Grand Unified Theory of Software Engineering, Industrial Info Systems, 2005.

[56] E. Bjarnason, K. Smolander, E. Engström, and P. Runeson, “A theory of distances in software engineering”, Information and Software Technology, 70, pp. 204-219, 2016.

[57] B. Boehm, and R. Ross, “Theory-W software project management principles and examples”, IEEE Transactions on Software Engineering, 15(7), 902-916, 1989.

[58] B. Boehm, and A. Jain, "An initial theory of value-based Software Engineering" Value-Based Software Engineering. Springer Berlin Heidelberg, 17-39, 2005.

[59] B.G. Glaser, A.L. Strauss, “The Discovery of Grounded Theory: Strategies for Qualitative Research”, Aldine Transaction, 1967.

[60] R. K. Yin, “Case study research: Design and methods”, Sage publications, 1994.

[61] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in Software Engineering." Emp. Software Engineering, vol. 14, no 2, pp. 131-164, 2009.

[62] A. Bryant, “Re-grounding Grounded Theory,” J. of Information Technology Theory and Application, 2002, 4:1, pp. 25-42.

[63] B. G. Glaser, “Doing Quantitative Grounded Theory”. Sociology Press, 2008.

[64] K. Charmaz, “Constructing grounded theory”. Sage, 2014.

[65] E. Mendes, B. Turhan, P. Rodríguez and V. Freitas, "Estimating the Value of Decisions Relating to Managing and Developing Software-intensive Products and Projects", Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering. ACM, 2015.

[66] K. M. Eisenhardt, and M. E. Graebner, M. “Theory building from cases: Opportunities and challenges”, *Academy of management journal*, *50*(1), 25-32, 2007.

[67] M. N. Marshall, “Sampling for qualitative research”. Family practice, 13(6), 522-526, 1996.

[68] T. Schwandt, “Qualitative Inquiry”, London, Sage, 1997.

[69] K. Petersen and C. Wohlin, "Context in industrial Software Engineering research." Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE Computer Society, 2009.

[70] B. G. Glaser, “Theoretical sensitivity: Advances in the methodology of grounded theory”, Sociology Pr., 1978.

[71] F. Welter, “Contextualizing entrepreneurship—conceptual challenges and ways forward”. *Entrepreneurship theory and practice*, *35*(1), 165-184, 2011.

[72] L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh, “The case for context-driven software engineering research: Generalizability is overrated”, *IEEE Software*, *34*(5), 72-75, 2017.

[73] B. J. Glaser, “The Grounded Theory Perspective III: Theoretical Coding”, MillValley:CA, Sociology Press, 2005.

[74] R. M. Davison, and M.G. Martinsons, “Context is king! Considering particularism in research design and reporting”, *Journal of Information Technology*, *31*(3), 241-249, 2016.

[75] Z. Cheng, A. Dimoka and P. Pavlou, “Context may be King, but generalizability is the Emperor!”, Journal of Information Technology, 31(4), 257-264, 2016.

[76] J. A. Maxwell, “Understanding and Validity in Qualitative Research,” Harvard Educational Review; Cambridge (62:3), p. 279, 1992.

[77] Payne, Adrian, and Pennie Frow. 2014. “Developing Superior Value Propositions: A Strategic Marketing Imperative.” Journal of Service Management 25 (2): 213–27.

[78] Levina, N., & Ross, J. W. (2003). From the vendor's perspective: exploring the value proposition in information technology outsourcing. MIS quarterly, 331-364.

Pilar Rodríguez. PhD 2013 Computer Science; MSc. 2008 Computer Science; BSc 2006 Computer Science; Assistant professor at Universidad Politécnica de Madrid (Spain) and docent at University of Oulu (Finland). Her research centers on VBSE, agile software development and human factors in SE. She has published in premier Software Engineering journals and conferences. She is a member of the Review Board of EMSE and has served as reviewer in leading academic SE forums (e.g. TSE, EMSE, IST).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

Cathy Urquhart. PhD 1999 Information Systems. Cathy Urquhart is Professor of Digital Business at Manchester Metropolitan University Business School. She is the author of *Grounded Theory for Qualitative Research*, published by Sage. She is a Senior Editor for the European Journal of Information Systems, an Editorial Board Member for Information Systems Journal and an Associate Editor for *Information Technology and Development*. She is a subject expert for the Association of Business Schools Scientific Journal Ranking Committee in the UK. She is a Fellow of the Association of Information Systems. Her website can be found here https://www2.mmu.ac.uk/business-school/about-us/our-staff/otehm/profile/index.php?id=899

Emilia Mendes. PhD 1999 Computer Science; MSc. 1992 Computer Science; BSc 1986 Computer Science; Full Professor in Computer Science at the Blekinge Institute of Technology (Sweden), Tekes-funded Finnish Distinguished Professor at the University of Oulu (Finland) (2015-2018). Main research areas of interest are empirical software engineering, VBSE, and the use of machine learning techniques to contexts such as healthcare, and sustainability. She has published widely and over 200 refereed papers, plus two books as solo author – both in the area of cost estimation. She is on the editorial board of TSE and IST, and has been former editorial board member in journals such as SQJ and EMSE. Further details are available here: https://scholar.google.se/citations?user=Mz4IbXgAAAAJ&hl=en&oi=ao

APPENDIX A: INTERVIEWEES' PROFILE

Company	I#	Interviewee Role (SW and/or HW)	Stakeholder Group	Experience SW industry (years)	Experience current role	Setting	Interview length
A	I1	Strategic product manager (SW)	Bus.	20 years (all of them in Company A)	5 years	Face-to-face	66 minutes
	I2	Strategic product manager (SW)	Bus.	20 years (17 years in Company A)	1.5 years	Face-to-face	70 minutes
	I3	Strategic product manager (SW/HW)	Bus.	25 years (all of them in Company A)	2 years	Face-to-face	46 minutes
	I4	Product owner (SW)	Bus./Dev.	15 years (3 years in Company A)	12 months	Face-to-face	84 minutes
	I5	Project manager (SW)	Dev.	20 years (1 year in Company A)	10 months	Face-to-face	57 minutes
	I6	Project manager (SW)	Dev.	15 years (1 year in Company A)	10 months	Face-to-face	28 minutes
	I7	Project manager (HW)	Dev.	19 years (9 months in Company A)	9 months	Face-to-face	61 minutes
	I8	System verification manager (SW)	Dev.	27 years (20 years in Company A)	2.5 years	Face-to-face	50 minutes
	I9	Continuous integration manager (SW)	Dev.	10 years (8 months in Company A)	8 months	Face-to-face	49 minutes
	I10	Ecosystem manager (SW/HW)	Bus./Dev.	23 years (3 years in company A)	3 years	Face-to-face	53 minutes
B	I11	Product manager (SW/HW)	Bus.	25 years (3 years in Company B)	3 years	Face-to-face	57 minutes
	I12	Sales-business dev. director (SW/HW)	Bus.	20 years (10 years in Company B)	5 years	Face-to-face	62 minutes
	I13	Requirements manager (SW)	Bus.	22 years (all of them in Company B)	3 years	Face-to-face	74 minutes
	I14	Program manager (SW/HW)	Bus./Dev.	18 years (all of them in Company B)	14 years	Face-to-face	61 minutes
	I15	Product owner (SW)	Bus./Dev.	16 years (1 year in Company B)	1 year	Face-to-face	69 minutes
	I16	Quality manager (SW)	Dev.	20 years (10 years in Company B)	4 years	Face-to-face	57 minutes
	I17	Operation manager (SW/HW)	Dev.	15 years (3 years in Company B)	3 years	Face-to-face	41 minutes
	I18	Head material sourcing team (HW)	Dev.	21 years (1 month in Company B)	1 month	Face-to-face	45 minutes
C	I19	Program manager (SW)	Bus.	15 years (4.5 years in Company C)	4.5 years	Telco	63 minutes
	I20	Product manager (SW)	Bus.	16 years (4 years in Company C)	1.5 years	Telco	77 minutes
	I21	Product marketing director (SW)	Bus.	17 years (11 years in Company C)	1.5 years	Face-to-face	99 minutes

APPENDIX B: INTERVIEW SCRIPT

INTRODUCTION

The goal of our interview is to understand, based on your knowledge and experience, the set of factors that you use in order to decide which features are to be included in a given release. These factors will represent what value means to you when you are selecting features for a given release.

A NOTE FOR INTERVIEWER

To show the drawing with the specific step on which we will focus the interview (if any).

This interview forms a part of a set of interviews that we are conducting in [Company X] with the goal of eliciting important factors (value factors) that are used by different stakeholders when selecting features for a given release. Such understanding will lead to: i) a common vocabulary regarding what value means to different stakeholders, ii) the use of such characterization in future decision making meetings; iii) the gathering of data from the decision making meetings that will later contribute towards building a company specific value estimation model. We appreciate your participation in this activity. We understand that the factors that are elicited are confidential for your company and this interview may contain sensitive data; therefore, we are going to use this interview as an instrument of data collection in an **anonymous** form. The interview will be recorded and transcribed (as previously agreed). At any point during the interview you can request to go **off the record**. The confidentiality of the interview's content will be dealt with using the following protocol:

❑ The **interview summary** will be sent to you for review (e.g. assessing whether the summary represents clearly the knowledge you provided us with in the interview). The results of the interview will be dealt with using an anonymous form. Similarly, from a research perspective, **all information** will be **treated as confidential** and published only after obtaining [Company X]'s consent.

This interview should last no longer than **1 hour.** However, we ask the interviewee allows for some extra time (max. 30 minutes) if needed. During the interview we will go through a set of context-related questions, questions concerning your role/work in the company, and questions regarding factors that you employ during feature selection.

A NOTE FOR INTERVIEWER

During the interview we will go through the following types of questions:

- **Warm-up questions**: demographic and context setting questions. Describing the interviewee work responsibilities related to feature selection (5 min).
- **Value factors elicitation questions**: elaboration of the set of value aspects that the interviewee considers when deciding upon a feature (45 min).
- **Wrap-up questions**: closing and final questions. Any missing topic relevant for this research that the interviewee would like to discuss (10 min).

Any questions before we start?

INTERVIEW QUESTIONS

1. WARM-UP QUESTIONS:

Q1.1: **Tell us something about your work experience?** [Subject's experience/role may impact his/her perspective on value aspects]

- ✓ *How long have you been with* [Company X]*?*
- ✓ *What is your current role? How long have you been in this role?*
- ✓ *How many product releases have you been involved with?*
- ✓ *How long have you worked for in industry? What other company experience have you had? Same industry and domain?*
- ✓ *Which were your previous positions/roles?*

2. ELICITATION OF VALUE FACTORS:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

A NOTE FOR INTERVIEWER Alternative questions aim to collect the same information. There is no need for asking all questions once the needed information has been collected.

[To show the drawing to the interviewee – if any] Let's focus on release X [last release of Product Z]. You have obtained the update file containing features as a 'wish-list' for release X. It contains a list of features that are in the 'wish-list' for release X.

Q2.1: What was important for you when you look at the set of features targeting at release X?
Q2.2: What did you look at to decide if a feature should be implemented or not?
Q2.3: What information/data did you ask to your stakeholders to understand the value that a feature would provide to release X? What information did you provide to the decision making?
Q2.4: What was the information from the file that you used to determine the set of features that could be included in release X? Did you consider all the features that were in the wish-list, or you considered only the top-ranked features?
Q2.5: Once these features were selected, what additional information did you need in order to decide if these features would be really included in release X?

- ✓ *Are all the features in the 'wish-list' sent off to a feasibility study? If not, how do you decide which ones to send for a feasibility study?*
- ✓ *What information did you need back from the feasibility study in order to help with selecting the features for release X?*

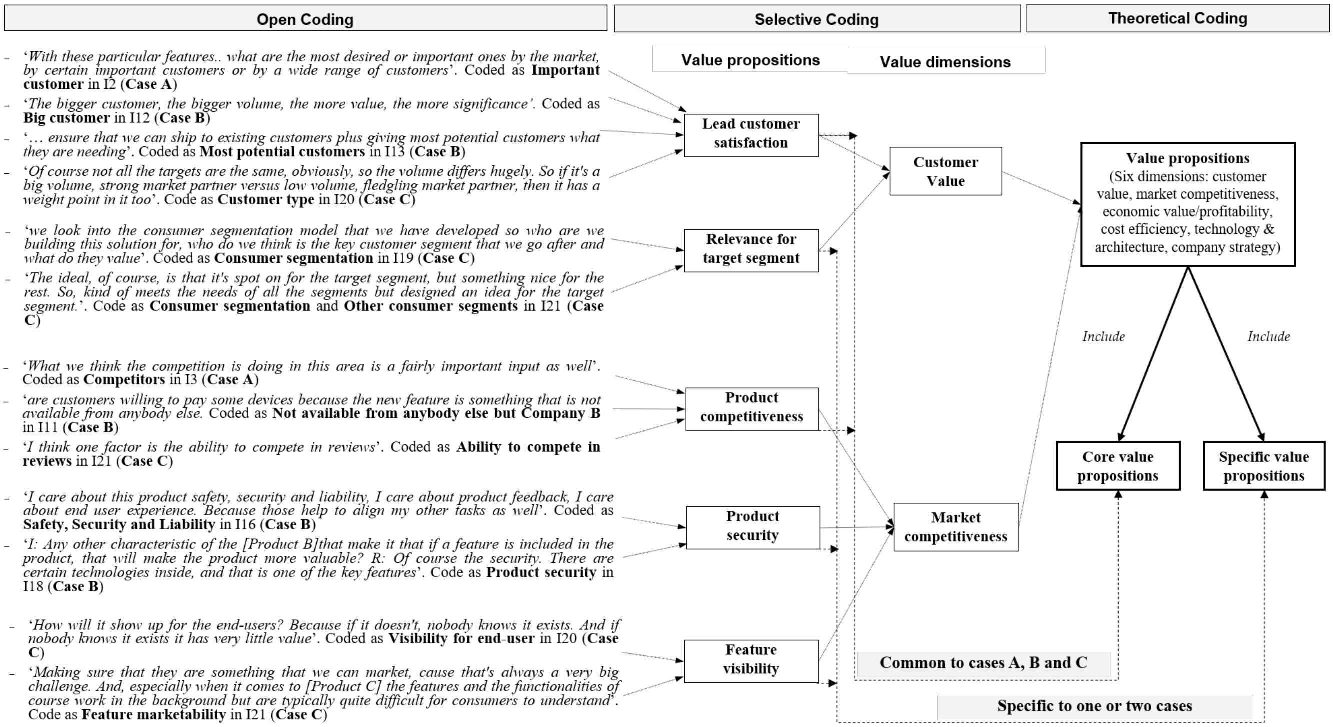
Q2.6: The list of features is updated every month. How/what were the updated information used in order to select features for release X?
Q2.7: Was it possible for other features that were not originally part of the wish-list to also be selected for release X? For example, features that should have been included in a previous release but did not make it. In relation to these additional features, what information did you use to decide which features to select for release X?

***3.* <u>WRAP-UP QUESTIONS:</u>**

Q3.1: Do you think that the factors you use during features' selection are always the same for every release? If not, what would be additional factors? Are there any other factors that you have used in a different release and not in release X?
Q3.2: Any related issues that we missed but you would like to reflect on?

Thank you for your time and participation in this interview. After we analyze the interview results, we will send you a summary of findings for checking on any inconsistencies with today's interview discussion. We hope you can get us **feedback within one week** after we send you the summary of findings. Also, if you would like we can provide you a full transcript of the interview.

APPENDIX C: DATA ANALYSIS – ILLUSTRATIVE EXAMPLE



- '*With these particular features.. what are the most desired or important ones by the market, by certain important customers or by a wide range of customers*'. Coded as **Important customer** in I2 (**Case A**)
- '*The bigger customer, the bigger volume, the more value, the more significance*'. Coded as **Big customer** in I12 (**Case B**)
- '*… ensure that we can ship to existing customers plus giving most potential customers what they are needing*'. Coded as **Most potential customers** in I13 (**Case B**)
- '*Of course not all the targets are the same, obviously, so the volume differs hugely. So if it's a big volume, strong market partner versus low volume, fledgling market partner, then it has a weight point in it too*'. Code as **Customer type** in I20 (**Case C**)
- '*we look into the consumer segmentation model that we have developed so who are we building this solution for, who do we think is the key customer segment that we go after and what do they value*'. Coded as **Consumer segmentation** in I19 (**Case C**)
- '*The ideal, of course, is that it's spot on for the target segment, but something nice for the rest. So, kind of meets the needs of all the segments but designed an idea for the target segment.*'. Code as **Consumer segmentation** and **Other consumer segments** in I21 (**Case C**)
- '*What we think the competition is doing in this area is a fairly important input as well*'. Coded as **Competitors** in I3 (**Case A**)
- '*are customers willing to pay some devices because the new feature is something that is not available from anybody else.* Coded as **Not available from anybody else but Company B** in I11 (**Case B**)
- '*I think one factor is the ability to compete in reviews*'. Coded as **Ability to compete in reviews** in I21 (**Case C**)
- '*I care about this product safety, security and liability, I care about product feedback, I care about end user experience. Because those help to align my other tasks as well*'. Coded as **Safety, Security and Liability** in I16 (**Case B**)
- '*I: Any other characteristic of the [Product B]that make it that if a feature is included in the product, that will make the product more valuable? R: Of course the security. There are certain technologies inside, and that is one of the key features*'. Code as **Product security** in I18 (**Case B**)
- '*How will it show up for the end-users? Because if it doesn't, nobody knows it exists. And if nobody knows it exists it has very little value*'. Coded as **Visibility for end-user** in I20 (**Case C**)
- '*Making sure that they are something that we can market, cause that's always a very big challenge. And, especially when it comes to [Product C] the features and the functionalities of course work in the background but are typically quite difficult for consumers to understand*'. Code as **Feature marketability** in I21 (**Case C**)

Open Coding
Selective Coding
Theoretical Coding
Value propositions
Value dimensions
Lead customer satisfaction
Relevance for target segment
Product competitiveness
Product security
Feature visibility
Customer Value
Market competitiveness
Value propositions (Six dimensions: customer value, market competitiveness, economic value/profitability, cost efficiency, technology & architecture, company strategy)
Include
Include
Core value propositions
Specific value propositions
Common to cases A, B and C
Specific to one or two cases

APPENDIX D: VALUE PROPOSITIONS (CORE AND SPECIFIC)

Id	Proposition	Description	Case A (n=10)	Case B (n=8)	Case C (n=3)	Totals	Core/ Specific
	Customer value		10[5] (140)[6]	8 (150)	3 (97)	**21 (387)**	
P1	Customer satisfaction	Impact that the implementation of the feature will have upon the satisfaction of the company's direct customer(s).	10 (67)	8 (40)	3 (26)	21 (133)	C
P2	End-user satisfaction	Impact that the implementation of the feature will have upon the satisfaction of product's end-users.	4 (13)	7 (77)	3 (39)	14 (129)	C
P3	Market satisfaction	Impact that the implementation of the feature will have upon the market in which the company operates (e.g. go-to-market model).	7 (25)	5 (11)	3 (7)	15 (43)	C
P4	# of customers	Impact that the implementation of the feature will have upon the amount of company's direct customers that will get satisfied (e.g. number of customers that have explicitly requested the feature).	4 (8)	5 (13)	2 (3)	11 (24)	C
P5	Lead customer satisf.	Impact that the implementation of the feature will have upon the satisfaction of important company's customer(s)	5 (24)	2 (4)	1 (2)	8 (30)	C
P6	Overall customer solution	Impact that the implementation of the feature will have upon the overall solution that the company offers to its direct customers (customer's portfolio).	2 (3)	0 (0)	2 (3)	4 (6)	S
P7	Customer partnership	Impact that the implementation of the feature will have upon making company's business the business of its customers as well.	0 (0)	2 (5)	3 (10)	5 (15)	S
P8	Relevance target segment	To what extent the feature is relevant to the product's end-user target segment (e.g. characterization of potential end-users).	0 (0)	0 (0)	3 (7)	3 (7)	S
Market competitiveness			10 (111)	8 (84)	3 (78)	**21 (273)**	
P9	Product competitiveness	Impact that the implementation of the feature will have upon the product's situation regarding to competitors.	5 (21)	5 (20)	3 (15)	13 (56)	C
P10	Time-to-market	Impact that the implementation of the feature will have upon the time-to-market of the release (i.e. time when the feature will be available for sale).	9 (46)	6 (16)	2 (3)	17 (65)	C
P11	Product quality	Impact that the implementation of the feature will have upon the quality of the product in general.	6 (25)	3 (12)	1 (3)	10 (40)	C
P12	Product performance	Impact that the implementation of the feature will have upon the performance of the product from a technical point of view (e.g. failure rate, speed, frequency coverage, etc.).	4 (8)	3 (10)	1 (1)	8 (19)	C
P13	UX-Product simplicity	Impact that the implementation of the feature will have upon the simplicity of using the product from an end-user's point of view.	2 (2)	3 (13)	3 (14)	8 (29)	C
P14	Product ubiquity	Impact that the implementation of the feature will have upon Product A's capability to support existing services in the network in a transparent way.	1 (3)	0 (0)	0 (0)	1 (3)	S
P15	Overall release content	To what extent the implementation of the feature will contribute to provide a compact package to be released (e.g. value of the feature when considering also other release's features).	4 (6)	0 (0)	0 (0)	4 (6)	S
P16	Product security	Impact that the implementation of the feature will have upon the security characteristics of Product B.	0 (0)	4 (13)	0 (0)	4 (13)	S
P17	Protection promise	Impact that the implementation of the feature will have upon the functionality that is promised by Product C.	0 (0)	0 (0)	3 (10)	3 (10)	S
P18	Feature visibility	To what extent the feature is visible for end-user (end-user can see that the feature has been implemented).	0 (0)	0 (0)	3 (19)	3 (19)	S
P19	End-user learning	Impact that the implementation of the feature will have upon teaching end-users about the product and its characteristics.	0 (0)	0 (0)	2 (7)	2 (7)	S
P20	Competitive deliver channel	Impact that the implementation of the feature will have upon the delivery channel's competitiveness (i.e. customers as delivery channels).	0 (0)	0 (0)	3 (6)	3 (6)	S
Economic value / profitability			6 (34)	7 (59)	3 (27)	**16 (120)**	
P21	ROI	Feature return on investment in monetary terms.	5 (16)	5 (25)	3 (5)	13 (46)	C
P22	Sales volume	Impact that the implementation of the feature will have upon the number of product units that will be sold.	2 (10)	6 (23)	3 (4)	11 (37)	C
P23	Customer base/market share	Impact that the implementation of the feature will have on the market share (e.g., whether the implementation of the feature will help engage new customers).	2 (3)	3 (6)	3 (14)	8 (23)	C

[5] Number of interviewees that mentioned the value proposition in the company case.

[6] Number of instances registered in NVivo for the value proposition in the company case. Please notice that counts in NVivo are exclusively intended to show the supporting evidence (level of theoretical saturation) of each value proposition and do not seek any kind of statistical generalisation as a goal.

Id	Proposition	Description	Case A (n=10)	Case B (n=8)	Case C (n=3)	Totals	Core/ Specific
P24	Product's price	Impact that the implementation of the feature will have upon the price of the product.	1 (1)	3 (5)	0 (0)	4 (6)	S
P25	Time-to-profit	Feature's time-to-profit (i.e. when the company will start making profit out of the feature).	1 (1)	0 (0)	0 (0)	1 (1)	S
P26	Opportunity cost	Impact that the implementation of the feature will have upon opportunity cost (i.e. benefit that could have been received by selecting an alternative feature).	2 (3)	0 (0)	3 (4)	5 (7)	S
Cost efficiency			10 (91)	8 (75)	3 (27)	**21 (194)**	
P27	Overall dev. costs	Overall cost/effort of implementing the feature (design, implementation, testing, verification and integration work, etc.).	10 (55)	8 (36)	3 (6)	21 (98)	C
P28	Reusability	Impact that the implementation of the feature will have upon reusing content (e.g. to what extent the feature can be reused).	2 (2)	4 (10)	2 (3)	8 (15)	C
P29	Third-party costs	Impact that implementing the feature will have upon making investments on third-party products or services (subcontractors).	2 (2)	4 (10)	1 (1)	7 (13)	C
P30	Testing cost	Cost/effort of testing the feature (low level testing).	5 (10)	1 (2)	0 (0)	6 (12)	S
P31	I&V cost	Cost/effort of the complete integration and verification of the feature (end-to-end verification).	3 (10)	0 (0)	0 (0)	3 (10)	S
P32	Testing equipment	Impact that the implementation of the feature will have upon the need to invest in testing equipment to be able to test the feature.	3 (9)	0 (0)	0 (0)	3 (9)	S
P33	Maintenance	Impact that the implementation of the feature will have upon post-deployment/maintenance costs.	0 (0)	4 (7)	3 (6)	7 (13)	S
P34	Material costs	Impact that the implementation of the feature will have upon material costs (hardware), including existing and new material.	3 (3)	3 (7)	0 (0)	6 (10)	S
P35	Dev. Infrastructure	Impact that the implementation of the feature will have upon costs related to tools needed to develop it (e.g. secure facilities).	0 (0)	3 (3)	0 (0)	3 (3)	S
P36	Delivery cost	Impact that the implementation of the feature will have upon the company's delivery channel efficiency (e.g. upgrade costs).	0 (0)	0 (0)	2 (12)	2 (12)	S
Technology & architecture			10 (82)	8 (71)	3 (17)	**21 (171)**	
P37	Implementation complexity	Impact that the implementation of the feature will have upon the complexity of implementing the product (e.g. limitations or challenges in development or technical dependencies).	8 (25)	7 (19)	3 (10)	18 (54)	C
P38	Product architecture	Impact that the implementation of the feature will have upon the architecture of the product (e.g. to what extent the implementation of the feature will enhance internal quality).	4 (5)	2 (2)	2 (3)	8 (10)	C
P39	Development capability	Impact that the implementation of the feature will have upon the capability for developing this feature and other features, considering both knowledge and resources.	9 (33)	6 (26)	2 (4)	17 (63)	C
P40	Hardware impact	Impact that implementing the feature will have from a hardware point of view (e.g. whether implementing the feature will require changes on the hardware).	3 (14)	4 (9)	0 (0)	7 (23)	S
P41	Certifications	Impact that the implementation of the feature will have upon secure certifications needed for the product.	1 (1)	3 (15)	0 (0)	4 (16)	S
P42	Technical relevance	To what extent the feature is relevant from a technical point of view (a feature may be technically irrelevant in this product although it is relevant in other company's products).	3 (5)	0 (0)	0 (0)	3 (5)	S
Company strategy			7 (39)	5 (22)	3 (37)	**15 (98)**	
P43	Company's strategy	Impact that the implementation of the feature will have upon supporting the company's strategy (i.e. to what extent the feature complies with the company's market strategy).	4 (13)	2 (3)	2 (6)	8 (22)	C
P44	Product's strategy	Impact that the implementation of the feature will have upon building the product concept that the company is looking for.	3 (6)	2 (11)	3 (17)	8 (34)	C
P45	Company's portfolio	Impact that the implementation of the feature will have upon the company's portfolio as the total offering.	5 (14)	2 (2)	2 (5)	9 (21)	C
P46	Business enabler	Impact that the implementation of the feature will have upon boosting other business or products and enabling new value streams (e.g. cross-sells with other company products).	3 (5)	2 (3)	2 (2)	7 (10)	C
P47	Company's brand	Impact that the implementation of the feature will have upon the company as a brand (e.g. company's prestige).	1 (1)	3 (3)	2 (7)	6 (11)	C

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2020.2989666, IEEE Transactions on Software Engineering

APPENDIX E: STAKEHOLDERS' VALUE PROPOSITIONS RECONCILIATION

					Business Group						◄────►					Development Group						
Value Proposition	**I1**	**I2**	**I3**	**I11**	**I12**	**I13**	**I19**	**I20**	**I21**	**I10**	**I14**	**I4**	**I15**	**I5**	**I6**	**I7**	**I8**	**I9**	**I16**	**I17**	**I18**	**TOTAL**
Customer Value	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**21/21**
P1 Customer satisfaction	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	21/21
P2 End-user satisfaction	1	1		1		1	1	1	1	1	1		1		1				1	1	1	14/21
P3 Market satisfaction	1	1	1	1	1		1	1	1	1	1	1	1		1	1			1			15/21
P4 # of customers	1	1		1	1	1		1	1		1				1		1		1			11/21
P5 Lead customer satisfaction	1	1			1	1		1				1		1	1							8/21
P6 Overall customer solution		1					1	1		1												4/21
P7 Customer partnership					1		1	1	1				1									5/21
P8 Relevance target segment							1	1	1													3/21
Market competitiveness	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**21/21**
P9 Product competitiveness	1	1	1	1	1		1	1	1	1			1			1			1		1	13/21
P10 Time-to-market	1	1	1		1	1	1	1		1	1	1	1	1	1	1	1		1	1		17/21
P11 Product quality	1	1	1			1		1			1	1				1		1	1			10/21
P12 Product performance	1	1						1			1	1			1					1	1	8/21
P13 UX-Product simplicity	1					1	1	1	1				1				1		1			8/21
P14 Product ubiquity	1																					1/21
P15 Overall release content		1	1									1			1							4/21
P16 Product security					1								1						1		1	4/21
P17 Protection promise							1	1	1													3/21
P18 Feature visibility							1	1	1													3/21
P19 End-user learning							1		1													2/21
P20 Competitive deliv. channel							1	1	1													3/21
Economic value/ prof.	**x**	**x**	**-**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**-**	**-**	**x**	**-**	**-**	**x**	**x**	**16/21**
P21 ROI	-1	1		1	1	1	1	1	1	1	1	1		1			1				1	13/21
P22 Sales volume	1	1		1	1	1	1	1	1		1		1							1		11/21
P23 Customer/market share	1	1		1		1	1	1	1												1	8/21
P24 Product's price		1			1						1										1	4/21
P25 Time-to-profit		1																				1/21
P26 Opportunity cost	1						1	1	1								1					5/21
Cost efficiency	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**21/21**
P27 Overall dev. cost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	21/21
P28 Reusability				1	1		1		1				1	1	1					1		8/21
P29 Third-party costs		1		1	1	1	1								1						1	7/21
P30 Testing cost				1								1		1		1	1	1				6/21
P31 I&V cost												1					1	1				3/21
P32 Testing equipment																1	1	1				3/21
P33 Maintenance				1	1	1	1	1	1											1		7/21
P34 Materials		1									1	1				1				1	1	6/21
P35 Dev. Infrastructure											1								1		1	3/21
P36 Delivery cost							1		1													2/21
Tech. & Architecture	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**21/21**
P37 Implement. complexity	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1	1	1		1	18/21
P38 Product architecture						1	1		1			1		-1	1		1	1		1		8/21
P39 Development capability	1	1		1			1		1	1	1	1	1	1	1	1	1	1	1	1	1	17/21
P40 Hardware impact		1	1			1					1	1								1	1	7/21
P41 Certifications		1			1	1													1			4/21
P42 Technical relevance	1	1										1										3/21
Company's Strategy	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**-**	**-**	**x**	**x**	**x**	**-**	**x**	**-**	**x**	**-**	**-**	**15/21**
P43 Company strategy	1	1					1	1		1			1				1		1			8/21
P44 Product's strategy	1	1	1		1	1	1	1	1													8/21
P45 Company's portfolio	1	1	1	1			1	1		1			1		1							9/21
P46 Business enabler		1		1		1	1	1		1				1								7/21
P47 Company's brand					1			1	1	1			1						1			6/21

APPENDIX F: THEORY EVALUATION

Evaluation of our theory according to Sjøberg et al. (2008)'s [21] six criteria for evaluating empirically-based theories in SE:

Criteria	Aspects to be considered to evaluate each criterion	Evaluation of our theory
1. Testability		
Options that the theory offers to be tested (i.e. empirical refutation or falsifiability) and refined	1.1 Constructs and propositions need to be clear and free from ambiguities	Constructs are relationships are described in Section 4. When possible, we formulated constructs and relationships using terms that are well-established in the SE literature (e.g. value proposition, stakeholder group, feature, and company's context).
	1.2 Hypothesis can be deduced from the theory's propositions so that they can be tested	We provide assumptions and logical justification for the hypothesis that can be deduced from our theory in Section 4 and Table 5 (e.g. specific value propositions are determined by company contextual factors that are specific for a company and cannot be generalized).
	1.3 The scope of the theory is clearly specified	The scope of interest and the scope of validity of our theory are defined in Table 5.
2. Empirical support		
Degree to which a theory is supported by empirical evidence.	2.1 Theory components are supported by empirical data.	Our theory is grounded on empirical data gathered from interviews (all constructs and relationships are grounded in data). For example, Appendix D shows the number of instances found in the interviews for each value proposition.
3. Explanatory power		
Theory's ability to provide explanations of why something happens	3.1 Analogy (degree to which a theory is supported by analogy to well-established theories)	By using GTM, our theory has been generated from scratch in SE and, therefore, we did not use any analogy to other established theories during the theory building process. However, our theory is integrated in the body of knowledge in Section 5.
	3.2 Explanatory breadth (degree to which a theory accounts for all known observations within its scope)	By applying GTM, our goal is not to generate to a population but to generalize to a theory. Our theory is the first of its kind in the field and it is based on three industrial cases. Further research will help integrate new cases to extend its scope.
4. Parsimony		
Value provided by each of the elements included in the theory	4.1 Unnecessary constructs and propositions should be excluded so that the theory is comprehensible	Theories built using GT tend to be complex due to the richness of the data to which the researcher is exposed [27]. We have attempted to use a minimum number of constructs and propositions in this theory as presented in Figure 1. Details on specific constructs (e.g. value propositions) are included as appendices to keep the theory comprehensible.
5. Generality		
How setting-independent the theory is	5.1 Extent to which a theory has a wide scope and how setting-independent the theory is.	The scope and generalizability of the theory was extended by using differentiated cases. We paid attention to precisely describe context information of our companies, as far as our NDAs allowed us to do so, in Section 3 to help practitioners compare their respective contexts with the contexts of our company cases. Future work comprises extending the scope of the theory by exploring those value propositions that are influenced by business domain and other company context characteristics.
6. Utility		
Utility of the theory in practice	6.1 SE theories should be useful to the software industry because of the applied nature of the SE discipline.	The theory serves as an input to value-based feature selection in the context of release planning for software and software intensive products by describing the nature of the value propositions applied when deciding upon software features. For example, it warns practitioners about the importance of understanding their own context in order to define their specific value propositions. Although we found a pattern of value propositions that suggests that some value propositions might be generally applicable to value-based feature selection decision (core value propositions), many other value propositions are context-dependent and, therefore, need to be locally elicited (specific value propositions). Moreover, we provide a set of core value propositions that can be used as a starting checklist when deciding upon software features (see Appendix D). Concerning specific value propositions, we provided detailed descriptions allowing practitioners to understand the way of thinking of our companies when defining their own specific value propositions. For a more detailed description of implications for practitioners, the reader is referred to Section 5.3.

APPENDIX G: VALUE PROPOSITIONS INTEGRATION TO RELATED EMPIRICAL STUDIES

Value Propositions (included in the theory and first case study [5])	**Wohlin and Aurum, 2005a [13]** # Companies: 2 # Subjects: 13 # Countries: -	**Wohlin and Aurum, 2005b [37]** # Companies: 9 # Subjects: 33 # Countries: -	**Barney et al., 2006 [12]** # Companies: 1 # Subjects: - # Countries: 1	**Hu et al., 2006 [38]** # Companies: 6 # Subjects: 72 # Countries: 3	**Barney et al., 2008 [39]** # Companies: 3 # Subjects: 26 # Countries: 2	**Barney et al., 2009 [36]** # Companies: 11 # Subjects: 107 # Countries: 3	**Latha and Suganthi, 2015 [40]** # Companies: - # Subjects: 26 #Countries: 1	**Alahyari et al. 2017 [10]** # Companies: 14 # Subjects: 23 # Countries: 1
Customer value	**Customer value*							
P1 Customer satisfaction (C)	***Req.'s issuer***[5]	***Req.'s issuer***	***Req.'s issuer***	***Customer satisf.***	***Req.'s issuer***	***Customer satisf.***	***Req.'s issuer***	
P2 End-user satisfaction (C)				*Req.'s issuer*		*Req.'s issuer*		***End-user performance, usability***
P3 Market satisfaction (C)								
P4 # of customers (C)								
P5 Lead customer satisfaction (C)	***Stakeholder priority of req.***[6]	***Stakeholder priority of req.***	***Stakeholder priority of req.***		***Stakeholder priority of req.***		***Stakeholder priority of req.***	
P6 Overall customer solution (S)								
P7 Customer partnership (S)								***Customer relationship***
P8 Relevance target segment (S)								
Market competitiveness	** Market technology trends*		** Competitive advantage*					
P9 Product competitiveness (C)	*Competitors*	*Competitors*	***Competitors***	*Competitors*	***Competitors***	*Competitors*	*Competitors*	***Competitiveness***
P10 Time-to-market (C)	***Delivery date/ calendar time***	***Delivery date/ calendar time***	*Delivery date/ calendar time*	***Calendar time***	*Delivery date/ calendar time*	*Calendar time*	***Delivery date/ calendar time***	***Delivery process w.r.t. time***
P11 Product quality (C)				***SW features***		***SW features***		***Perceived qual./ actual quality, Functionality, NFRs***
P12 Product performance(C)								***End-user performance, Reliability***
P13 UX-Product simplicity(C)								***Usability***
P14 Product ubiquity (S)								
P15 Overall release content (S)								
P16 Product security (S)								
P17 Protection promise (S)								
P18 Feature visibility (S)								
P19 End-user learning (S)	*Support for education/training*	*Support for education/training*	*Support/education/training*		*Support/education/training*		***Support for education/training***	***Knowledge of feature value for customer***
P20 Competitive deliv. Channel (S)								
Economic value/profitability								
P21 ROI (C)	***Development cost-benefit***	***Development cost-benefit***	***Development cost-benefit***		***Development cost-benefit***		*Development cost benefit*	***Revenue, business value***
P22 Sales volume (C)								
P23 Customer base/market share (C)								
P24 Product's price (S)								
P25 Time-to-profit (S)								
P26 Opportunity cost (S)								
Cost efficiency								
P27 Overall development cost (C)	** Product cost*			***Development cost*** ** Total ownership costs*		*Development cost*		***Cost***
P28 Reusability (C)								
P29 Third-party costs (C)								
P30 Testing cost (S)								
P31 I&V cost (S)								
P32 Testing equipment (S)								
P33 Maintenance (S)	*Maintenance*	*Maintenance*	*Maintenance*		*Maintenance*		*Maintenance*	***Maintainability***
P34 Materials (S)								
P35 Dev. Infrastructure (S)								
P36 Delivery cost (S)								

For survey studies (i.e. [13], [37], [12], [38], [39], [36] and [40]), the table includes *all propositions* that were considered in the survey and highlights in bold those ***propositions*** that the study found relevant for selecting requirements/features (at least for one of the studied cases). [10] is an interview-based study where preliminary options were not predefined by researchers but propositions were bottom-up identified from the interview data.

(*): propositions that were not part of the criteria predefined by researchers but were suggested by survey's respondents are marked with *.

[5,6] '*Requirement's issuer*' and '*Stakeholder priority of requirement*' were associated to the Customer value dimension because they were connected to customers in the original studies.

Value propositions that are not mentioned in previous empirical studies are highlighted in grey.

APPENDIX G: VALUE PROPOSITIONS INTEGRATION TO RELATED EMPIRICAL STUDIES (CONTINUATION)

Value Propositions (included in the theory and first case study [5])	**Wohlin and Aurum, 2005a [13]** # Companies: 2 # Subjects: 13 # Countries: -	**Wohlin and Aurum, 2005b [37]** # Companies: 9 # Subjects: 33 # Countries: -	**Barney et al., 2006 [12]** # Companies: 1 # Subjects: - # Countries: 1	**Hu et al., 2006 [38]** # Companies: 6 # Subjects: 72 # Countries: 3	**Barney et al., 2008 [39]** # Companies: 3 # Subjects: 26 # Countries: 2	**Barney et al., 2009 [36]** # Companies: 11 # Subjects: 107 # Countries: 3	**Latha and Suganthi, 2015 [40]** # Companies: - # Subjects: 26 #Countries: 1	**Alahyari et al. 2017 [10]** # Companies: 14 # Subjects: 23 # Countries: 1
Technology and architecture								
P37 Implementation complexity (C)	*Complexity* *Req. dependency.*	*Complexity* *Req. dependency*	*Complexity* *Req. dependency*	*Complexity* *Req. dependency*	*Complexity* *Req. dependency*	*Complexity* *Req. dependency*	*Complexity* *Req. dependency*	
P38 Product architecture (C)	*System impact* *Evolution*	*System impact* *Evolution*	*System impact* *Evolution* _*Operating architecture_	*Evolution*	*System impact* *Evolution*	*Evolution*	*System impact* *Evolution*	***Non-functional req., Hedonic value***
P39 Development capability (C)	***Resources/ competences***	*Resources/ competences*	***Resources/ competences***	*Resources*	***Resources/ competences***	*Resources*	*Resources and competences*	
P40 Hardware impact (S)								
P41 Certifications (S)								
P42 Technical relevance (S)								
Company strategy								
P43 Company strategy (C)	_* Strategic importance/alignment_			***Business strategy***		***Business strategy***		
P44 Product's strategy (C)	_* Function is promised/sold_[7]		***Function is promised/sold***		***Function is promised/sold***		***Function is promised/sold***	
P45 Company's portfolio (C)								
P46 Business enabler (C)			_*New business_					
P47 Brand (C)								
Other propositions (factors that did not emerge in our study)	*Req.'s volatility*	*Req.'s volatility*	*Volatility* _*Adherence to design parameters_	*Req. volatility* *Extra cost* *After-sale support* _*Business model_ _* Industry character_	*Volatility*	*Req. volatility* *Extra cost* *After-sale support*	*Req. volatility*	***WoW and Tools*** ***To keep positive attitude*** ***Innovation, knowledge of organization***

[7] '*Function is promised/sold*' makes reference to the strategy of the product from the point of view of the expectation of stakeholders to see the requirement met is taken into account.

APPENDIX H: STUDY VALIDITY – PROVISIONS FOR SECURING TRUSTWORTHINESS OF THE STUDY

Appendix H presents details on the mitigation strategies that we applied to minimize case study specific validity threats. For a detail description of each mitigation action, the reader is referred to [5] (Section 5.4), as the same validity procedures were considered in the three case studies.

Criteria	Description	Threats	Mitigation strategies
Construct validity	To what extent operational measures represent the concepts being studied according to the research questions.	– Relevance of the case to address the research questions. – Rigor in data collection (e.g. misconceptions such as questions of the interview being interpreted in a different way by researchers and participants).	– Value-based decision making in place. – Company own motivation to take part in the research. – Walkthrough meetings with company representatives to set-up the study and prepare the interview script. – Analysis of individual interviews emailed to the interviewees for feedback, including interview quotations in order to keep a clear chain of evidence from the interview's data to the value propositions, and traceability between propositions and evidence.
Internal validity	The extent to which other confounding aspects/factors may influence the results that are identified.	– Interviewee's inaccurate view on value propositions.	– Selection of participants (all/ a representative sample of key stakeholders who participate in feature selection meetings). – Very experienced participants.
External validity	To what extent the findings of the case study are of interest to other people outside the investigated case.	– Appropriateness of the company case. – Representativeness of the company case.	– Company thinking in terms of value. – Detailed description of the cases' context. – Theoretical integration (detailed comparison to related studies that demonstrates the path towards generalization). – Steps to analyze/integrate additional cases in [5] - Appendix A.
Reliability	To what extent the study can be replicated obtaining the same results.	– Measurement bias (e.g. reliability of the measurement instrument/raw data). – Researcher bias (e.g. preconceptions of researchers in data collection/analysis and inappropriate use of analysis methods). – Participant bias (e.g. subject's subjectivity, willingness to provide reliable data).	– Interview script/questions prepared in accordance to the RQs. – Interviews recorded and transcribed. – Well-established coding techniques and tool support for data analysis (NVivo). – Researcher triangulation in both data collection and analysis. Data collection: The first and third authors conducted together all interviews. Data analysis: The first and third authors individually analyzed interview I4 (the first conducted in a chronological order) and compared the results in order to check their level of agreement (open coding). Overall, the level of agreement was high 75%) - from the 16 value propositions identified from interview I4 (see Appendix E), 12 value propositions were commonly identified by both researchers and four value propositions were agreed upon during the meeting. The main differences were in the terminology used to code value propositions. Thus, it was agreed that the third author would also review every individual report before it was emailed to the participant for feedback. Selective and theoretical coding were discussed among the three authors. – CCM helped saturate value propositions and dimensions. – Participants own motivation to be interviewed. – Individual interviews, NDA's and guarantee for anonymity.