CrossMark

Towards improving decision making and estimating the value of decisions in value-based software engineering: the VALUE framework

Emilia Mendes[1] · Pilar Rodriguez[2] · Vitor Freitas[2] ·
Simon Baker[3] · Mohamed Amine Atoui[1,2,3]

Abstract To sustain growth, maintain competitive advantage and to innovate, companies must make a *paradigm shift* in which both short- and long-term value aspects are employed to guide their decision-making. Such need is clearly pressing in innovative industries, such as ICT, and is also the core of Value-based Software Engineering (VBSE). The goal of this paper is to detail a framework called VALUE—improving decision-making relating to software-intensive products and services development—and to show its application in practice to a large ICT company in Finland. The VALUE framework includes a mixed-methods approach, as follows: to elicit key stakeholders' tacit knowledge regarding factors used during a decision-making process, either transcripts from interviews with key stakeholders are analysed and validated in focus group meetings or focus-group meeting(s) are directly applied. These value factors are later used as input to a Web-based tool (Value tool) employed to support decision making. This tool was co-created with four industrial partners in this research via a design science approach that includes several case studies and focus-group meetings. Later, data on

✉ Emilia Mendes
  emilia.mendes@oulu.fi

  Pilar Rodriguez
  pilar.rodriguez@oulu.fi

  Vitor Freitas
  vitor.freitas@oulu.fi

  Simon Baker
  simon.baker@cl.cam.ac.uk

  Mohamed Amine Atoui
  amine.atoui@gmail.com

[1] University of Oulu and Blekinge Institute of Technology, Oulu, Finland

[2] University of Oulu, Oulu, Finland

[3] University of Cambridge, Cambridge, UK

🔄 Springer

key stakeholders' decisions gathered using the Value tool, plus additional input from key stakeholders, are used, in combination with the Expert-based Knowledge Engineering of Bayesian Network (EKEBN) process, coupled with the weighed sum algorithm (WSA) method, to build and validate a company-specific value estimation model. The application of our proposed framework to a real case, as part of an ongoing collaboration with a large software company (company A), is presented herein. Further, we also provide a detailed example, partially using real data on decisions, of a value estimation Bayesian network (BN) model for company A. This paper presents some empirical results from applying the VALUE Framework to a large ICT company; those relate to eliciting key stakeholders' tacit knowledge, which is later used as input to a pilot study where these stakeholders employ the Value tool to select features for one of their company's chief products. The data on decisions obtained from this pilot study is later applied to a detailed example on building a value estimation BN model for company A. We detail a framework—VALUE framework—to be used to help companies improve their value-based decisions and to go a step further and also estimate the overall value of each decision.

1 Introduction and motivation

Many ICT companies worldwide use only cost and effort estimates when making decisions relating to their software/software-intensive products. However, given today's cutthroat product and services industries, to rely solely upon such short-term decisions is misguided (Biffl et al. 2005). To sustain growth, maintain competitive advantage and to innovate, companies must make a *paradigm shift* in which both short- and long-term value aspects are employed to guide their decision-making. Such need is clearly pressing in innovative industries, such as ICT (Mendes et al. 2015), and is also the core of value-based software engineering (VBSE) (Boehm 2003). Boehm (2003) very clearly has put forward that much of the current software engineering practice and research is carried out in a value-neutral setting, where

- Every requirement, use case, object, test case and defect is treated as equally important
- "Earned value" systems track project cost and schedule, not stakeholder or business value
- A "separation of concerns" is practised, in which the responsibility of software engineers is confined to turning software requirements into verified code.

As part of a 3-year research project (2015–2017) funded by the Finnish Funding Agency for Technology and Innovation, we have the opportunity to collaborate with three large and one small ICT companies operating in Finland to support them improve their decision-making process relating to software/software-intensive products. To this aim, we are employing a mixed-methods approach (Easterbrook et al. 2007), named the VALUE framework (VALUE—improving decision-making relating to software-intensive products and services development), which is detailed throughout this paper using as basis some empirical findings and data resulting from our collaboration with one of the large ICT companies.

All of our partner companies in this project already employ a value-based approach to their decision-making process; however, in all cases, their decisions are based on key stakeholders'

tacit knowledge, and without tool support, or value estimation. They employ a value-based approach to decision making in a wide range of decision types, as follows: company A (the company focus of this paper) and another one of our industry partners apply value-based decision making in the context of product road mapping and feature selection. Our third industry partner employs a value-based approach when planning its products; however, in this case, the company decides upon ideas instead of features (i.e. selecting the ideas that will be implemented in the next product). Finally, our fourth industrial partner uses a value-based decision approach in a completely different scenario, when deciding upon external research programs in which to apply for; that is, when selecting the external research programs that the company will join based on the value that those programs are expected to provide if approved.

All decisions are carried out within the scope of a complex knowledge domain (e.g. product management, project management), and therefore, they present an uncertain nature (Darwiche 2010; Mendes 2012; Lempert et al. 2004). Note that herein, uncertain means that the knowledge is based on beliefs and therefore cannot be assumed to be absolute with deterministic outcomes (Pearl 1988).

The decision-making literature advocates that a suitable solution to support decision-making under uncertainty is to make explicit decision makers' mental models (explicitation of mental models) (Fabri et al. 2011; Steiger and Steiger 2007; Steiger 2010); once made explicit, such models can be compared during different decision scenarios and hence provide better understanding of the situation at hand (Fabri et al. 2011; Chermack, 2003; Steiger 2010).

In addition, decisions (how one sees, thinks or acts in the world) are influenced by decision makers' mental models (Foos et al. 2006; Steiger 2010); therefore, updating and enriching these mental models leads to an improved decision-making process (Mendes 2012; Steiger 2010; Dyba 2003; Fabri et al. 2011; Foos et al. 2006). Finally, mental models (a.k.a. representations and cognitive maps (Chermack 2003)) are enhanced through the use of a knowledge creation process (Garvin 1998; Lempert et al. 2004; Nonaka and Toyama 2003), and the relationship between both is detailed in Sect. 2.

The VALUE framework (detailed in Sect. 3) includes a mixed-methods approach, as follows: to make explicit key stakeholders' tacit knowledge relating to the factors employed during a decision-making process (Sect. 2.1), either transcripts from interviews with those stakeholders are analysed and followed by focus groups meetings attended by them or directly elicited in one or more focus-group meeting(s) attended by them (Sect. 4). These value factors are later used as input to a Web-based tool employed to support decision-making (Value tool, described in Sect. 5). This tool was co-created with all of our four industrial partners via a design science approach that includes several case studies and focus-group meetings. Later, data on decisions, gathered using the Value tool, plus additional input from key stakeholders, are used, in combination with the expert-based knowledge engineering of Bayesian networks (EKEBN) process (Sect. 2.2), coupled with the WSA method (Sect. 6), to build and validate a company-specific value estimation model. The application of our proposed framework to one of our four ICT industry partners (company A henceforth) is presented herein. Further, we also provide a detailed example, partially using real data on decisions, gathered via a pilot study with company A, of a value estimation BN model for this company.

Within the abovementioned context, the prediction technique chosen was Bayesian Network (BN) (Jensen 1996), selected for the following main reasons: (i) it is a technique used successfully in a diverse range of domains (e.g., Darwiche (2010) provides a very good overview of BNs' use in a range of different domains), which also include software engineering (further details on the use of BNs in Software Engineering is given in Sect. 2.4); (ii) it

enables models to be created based solely on domain expert knowledge, data or using a combination of both; (iii) it embeds the uncertainty that is inherent to the knowledge domain being modelled, and also the inter-relationships between predictors and response variables and (iv) its tool implementation enables the use of numerous "what-if" scenarios, which can support the decision-making process (e.g. Mendes and Mosley 2008; Mendes 2012b; Misirli and Bener 2014; Radlinski 2010). This technique was applied using an adaptation of the knowledge engineering of Bayesian networks process (Woodberry et al. 2004) (EKEBN), introduced in Sect. 3 and detailed in Sect. 6.

This is the first time within the context of VBSE that such framework solution is being suggested for value-based decision making, and also the first time that a genuine prediction technique has been put forward to estimate the overall value of decisions within the context of software/software-intensive product management and development. However, note that we see the use of a value estimation model as complementary to other data visualisation techniques in helping improve value-based decision-making, rather than as the only support. We are dealing herein with both strategic and operational decision-making processes, thus we are providing a wider range of decision-making visualisation options in our Web-based Value tool (Shen-Hsieh and Schindl 2002) that mix the use of a common visual language for data (e.g. bar charts) (e.g. Davenport 2013) and what-if scenarios with probabilistic thinking (e.g. BNs) (e.g. Mendes 2014; Misirli and Bener 2014). The use of such mix has also been agreed upon during meetings with the participating industry partners.

The contribution of this paper is threefold: (i) to detail our proposed VALUE framework, aimed to support our industry partners, and other companies who wish to improve their value-based decision-making; (ii) to present and discuss some empirical results relating to two of the five parts of that framework, i.e. the elicitation of value factors and their use with tool support for decision-making and (iii) to describe, via a research validation, the last three parts of the framework, where an example, partially based on real data on decisions gathered during the pilot study, explains step-by-step an approach to building and validating a value estimation model using BNs.

Note that there are situations when a new technique/method/solution is invented and described without prior adoption in an industrial setting because its uptake can take some time from months to even years (Wieringa and Heerkens 2006). We put forward that such delay may be one of the main, if not the main, reasons as to why there are several refereed conference and journal papers providing research validation via examples (Shaw 2003). Such examples can be used for instance, as means to encourage collaboration with industry (e.g. Holland and Dawson 2011; Fouad et al. 2011; Ghazinoory et al. 2014; Hamzah et al. 2010), or to provide a possible solution to a problem for which there is already evidence of industrial interest (e.g. Alégroth 2013; Hayes and Offutt 2006). The research validation we are detailing herein falls under the latter case.

The remainder of the paper is organised as follows: Sect. 2 provides some background and related work information in relation to (i) the conceptual framework of knowledge creation, (ii) Bayesian networks, (iii) value-based decision making and (iv) Bayesian networks applied to decision-making in software engineering. Section 3 describes the VALUE framework presenting its five main parts. Section 4 presents the first part of the framework—A1, detailing how we explicitated the knowledge relating to value factors from key stakeholders in company A. Section 5 describes the second part of the framework—A2, via a pilot study using our bespoke Web-based Value tool, embedding the value factors from A1, which supports decision-making. Section 6 takes the reader through the EKEBN process applied to building a value estimation

model for company A, using real decisions data gathered from A2, complemented with an example; this covers the other three parts of the VALUE framework—A3 to A5. Section 7 provides our comments on implications of this work to research and practice, followed by a discussion on threats to the validity of our work in Sect. 8; finally, conclusions and comments on future work are presented in Sect. 9.

2 Background and related work

2.1 A conceptual framework of knowledge creation

Knowledge creation is one of the three different processes (knowledge creation, transfer and application) embedded into a conceptual framework called the theory of organisational knowledge creation by Nonaka and Toyama (2003), which is used as basis for all the improvement actions that will be detailed throughout this paper. This framework is cited in numerous knowledge management studies (e.g. Dyba 2003; Garvin 1998; Schneider 2009) and has also been previously used to guide improvement activities in software process improvement and effort prediction studies, with promising results (Arent and Nørbjerg 2000; Aurum et al. 2003; Mendes 2012; Mendes 2014). It organises the knowledge process into four different stages (Nonaka and Toyama 2003) (see Fig. 1):

1. Tacit to tacit, where experiences, skills and expertise are shared between individuals (I, Fig. 1). Such knowledge transferring takes place via socialisation only, which leads to learnt skills and shared mental models, and without any written manuals or guides
2. Tacit to explicit, where tacit knowledge is 'translated' by an individual or by a group (G, Fig. 1) into an explicit (tangible) representation. The end result is a tangible representation of tacit knowledge. Note that the explicitation of tacit knowledge has numerous advantages, which include a wider sharing of existing expert knowledge within an organisation (e.g. Dyba 2003; Garvin 1998; Schneider 2009)
3. Explicit to explicit, where explicit knowledge from different groups is gathered, combined, edited and diffused. Here, all the knowledge being combined represents tangible knowledge that had already been previously represented explicitly from tacit knowledge. There are numerous advantages to an organisation in combing explicit knowledge, which include the creation of a common understanding about a particular practice/process, which can be used as basis for best practice standards to be disseminated and followed throughout the organisation (O, Fig. 1)
4. Explicit to tacit, where explicit knowledge is absorbed by individuals in groups within the organisation via action and practice thus enhancing those individuals' mental models

Fig. 1 The four different stages of the theory of organisational knowledge creation. Image prepared by Ms. Jacy Rabelo
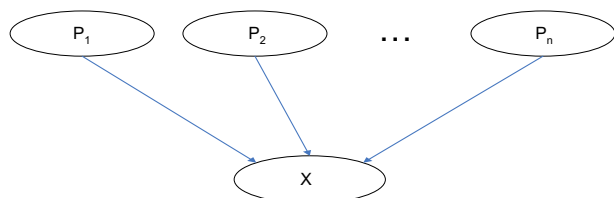
As it is represented in Fig. 1, knowledge creation is meant to be a continuous process traversing all four stages as an integral part of it, i.e. to be a knowledge spiral (Nonaka and Toyama 2003).

2.2 Bayesian network

A Bayesian network (BN) is a model that supports reasoning with uncertainty due to the way in which it incorporates existing knowledge of a complex domain (Pearl 1988). This knowledge is represented using two parts. The first, the qualitative part, represents the structure of a BN as depicted by a directed acyclic graph (digraph) (see Fig. 2). The digraph's nodes represent the relevant variables (factors) in the domain being modelled, which can be of different types (e.g. observable or latent, categorical). The digraph's arcs represent the causal relationships between variables, where relationships are quantified probabilistically (Pearl 1988, Heckerman 1995).

The second, the quantitative part, associates a conditional probability table (CPT) to each node, its probability distribution. A parent node's CPT describes the relative probability of each state (value) (Fig. 2, nodes 'customer retention' and 'customer satisfaction'); a child node's CPT describes the relative probability of each state conditional on every combination of states of its parents (Fig. 2, node 'overall value'). So, for example, the relative probability of overall value being 'positive' conditional on customer retention and customer satisfaction being both 'negative impact' is zero or the relative probability of overall value being positive conditional on customer retention and customer satisfaction being both 'positive impact' is 100.
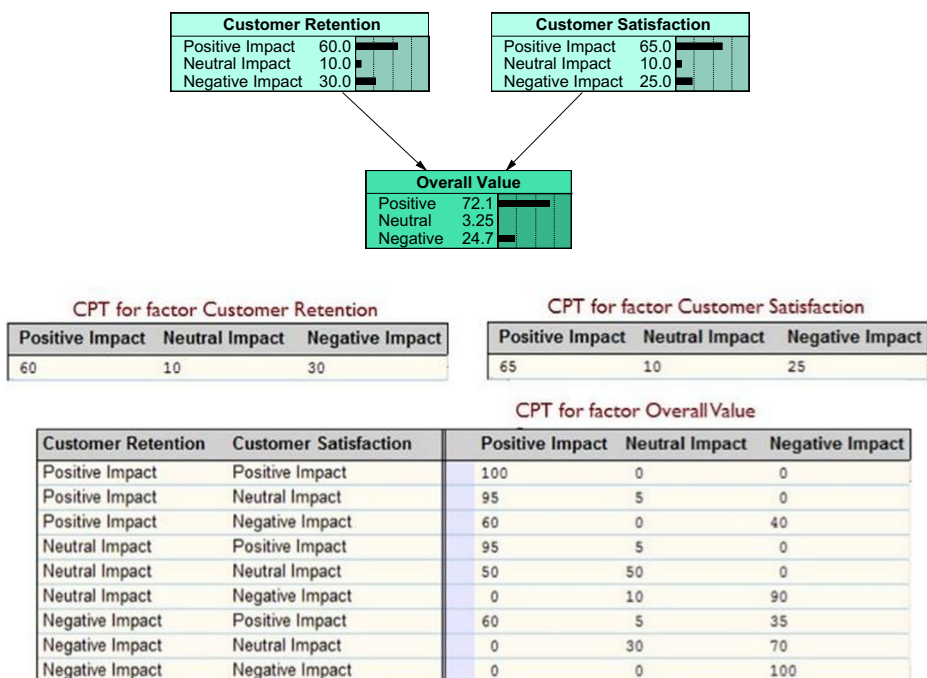

Customer Retention
Positive Impact	60.0
Neutral Impact	10.0
Negative Impact	30.0

Customer Satisfaction
Positive Impact	65.0
Neutral Impact	10.0
Negative Impact	25.0

Overall Value
Positive	72.1
Neutral	3.25
Negative	24.7

CPT for factor Customer Retention

Positive Impact	Neutral Impact	Negative Impact
60	10	30

CPT for factor Customer Satisfaction

Positive Impact	Neutral Impact	Negative Impact
65	10	25

CPT for factor Overall Value

Customer Retention	Customer Satisfaction	Positive Impact	Neutral Impact	Negative Impact
Positive Impact	Positive Impact	100	0	0
Positive Impact	Neutral Impact	95	5	0
Positive Impact	Negative Impact	60	0	40
Neutral Impact	Positive Impact	95	5	0
Neutral Impact	Neutral Impact	50	50	0
Neutral Impact	Negative Impact	0	10	90
Negative Impact	Positive Impact	60	5	35
Negative Impact	Neutral Impact	0	30	70
Negative Impact	Negative Impact	0	0	100

Fig. 2 Very basic example of a value estimation model

Formally, the posterior distribution of the Bayesian network is based on Bayes' rule (Pearl, 1988):

$$p(X|E) = \frac{p(E|X)p(X)}{p(E)} \tag{1}$$

where

- $p(X|E)$ is called the *posterior* distribution and represents the probability of X given evidence E.
- $p(X)$ is called the *prior* distribution and represents the probability of X before evidence E is given.
- $p(E|X)$ is called the *likelihood* function and denotes the probability of E assuming X is true.

Once a BN is specified, evidence (e.g. values) can be entered into any node and probabilities for the remaining nodes automatically calculated using Bayes' rule (Pearl 1988). Therefore, BNs can be used for different types of reasoning, such as predictive, diagnostic and what-if analyses to investigate the impact that changes on some nodes have on others (Neil and Fenton 1996).

2.3 Value-based decision-making

Software engineering decisions have traditionally taken place in a value-neutral setting in which short-term aspects such as project costs and schedule were predominantly considered (Boehm 2003). Thus, over the last 30 years, significant effort has been put into developing software cost estimation techniques (Jorgensen and Shepperd 2007). Numerous studies have shown that the primary critical success factor that differentiates successful products/projects from failed ones lie in the value domain (Biffl et al. 2005). Value-based software engineering (VBSE) emerged in 2003 (Boehm 2003) to stress the importance of considering value in software-related decisions and to highlight that it is not enough to merely meet schedules, budget and quality objectives but, ultimately, software products need to create value to achieve long-term growth (Biffl et al. 2005).

When thinking in terms of value, the spectrum and variety of stakeholders involved in the decision making is wider than when thinking in terms of costs, and a broader range of factors is also considered (e.g. business, marketing, technical, etc.). In addition, different value propositions from different stakeholders have also been recognised as an aspect characterising value in the software domain (Biffl et al. 2005). For example, different stakeholders might differ on which features bring the most value to a product and on reasons for arguing their views.

With regard to earlier work in VBSE, previous studies have proposed value considerations and corresponding measurement solutions needed for making decisions about software product management and development (e.g. Biffl et al. 2005; Boehm 2003). A systematic mapping study, including 364 primary studies in value-based decision-making, showed that these contributions were often isolated and with a limited perspective (Jan and Ibrar 2010). Khurum et al. (2012) proposed a large classification of ~50 value perspectives using as basis the work from Jan and Ibrar (2010) and also additional literature from economics, business and marketing research. They argued that such classification represents the views of all the different stakeholders who make decisions relating to software products. Later, this classification was extended to include a project management perspective (Wnuk and Mendes 2015);

however, both our previous industrial collaboration building numerous effort estimation models (Mendes 2012; Mendes 2014) and our current experience on value-based decision making with the four ICT partners showed that the use of a very detailed classification of factors that requires considerable training in order to be used during knowledge elicitation lead to the industry's disengagement from collaborating. In other words, industry's needs must be met specifically and to be company-specific—time spent has to be focused upon eliciting solely the value factors that are deemed important for a specific organisation in order to be able to successfully support them in their decision-making processes and to later build the company-specific value estimation model. In summary, we argue that the value factors important for an organisation should be co-created from the start with and for that organisation.

Further, there have also been several studies investigating stakeholder value propositions in relation to product management and release planning (studies closer to the scope of this research) (e.g. Wohlin and Aurum 2005; Barney et al. 2006; Barney et al. 2008; Barney et al. 2009; Fogelström 2010; Achimugu et al. 2014; Lindgren et al. 2008a, b). However, these studies differ from our approach in two distinct ways: (i) some employ mainly a top-down aspect by means of research methods such as surveys using semi-structured questionnaires and using checklists as instruments. As a consequence, the decision-making criteria employed are predefined by the researchers, instead of emerging in collaboration with the key stakeholders. Such approach can likely lead to results that miss additional important criteria due to issues such as anchoring effect (McElroy and Dowd 2007). Moreover, surveys can also suffer from a range of problems, such as non-response bias (Davern 2013), and (ii) other studies base their results solely on notes taken during interviews, which provide better means to explicitate the tacit knowledge from the interviewees; however, such approach does not guarantee that all the important knowledge has been elicited (Shadbolt and Smart 2015). In turn, the method we employed to identify value propositions from key stakeholders (detailed in Sect. 4) has used the text from transcribed interviews, in addition to Grounded theory coding principles (Strauss and Corbin 1998), in order to identify value factors. Such method enabled all interviewees to speak freely and for the two first authors, who interviewed jointly all ten stakeholders, to focus upon the questions and knowledge being provided. This is the first time such approach is employed to elicit value propositions from key stakeholders in the context of feature selection for release planning in software-intensive products.

Finally, two other studies claim to employ value estimation approaches within the context of product development, as follows: (i) Castro et al. (2012) proposed a measure to assess the value of product functions during their development by calculating for each function being implemented its estimated size and quality divided by its estimated cost. Although the authors describe their approach as an estimation model, it is in fact solely a measure of a function's value, not an estimation model. (ii) Mohamed and Wahba (2008) put forward a technique called value point measurement, which employs a waterfall-like process using as input the list of requirements to develop in a given release, weights for different stakeholders as means to identify their importance in the decision process, a classification of 14 different requirement characteristics (measured for each requirements using a four-point ordinal scale) and the type of project for which the value points are being calculated (new, enhancement). These different inputs are used to obtain a value point for each requirement, using a process very similar to function points counting; the output from this waterfall process, which is embedded in a prototype tool, is a list of requirements ranked by their value point. The authors also argue that their work details a value estimation model; however, what they did is simply to define a measure to assess each requirement. In summary, with regard to studies employing prediction

techniques to support value-based decision making within the context of software/software-intensive product management, to the best of our knowledge, no effort in this direction has been developed so far, making our VALUE framework the first to address this research challenge.

Undoubtedly, the topic of value prediction is challenging and requires a detailed understanding of value within the context of product management decision-making. The research detailed herein puts forward a framework aimed to solving such challenge, thus bringing additional benefits to companies that are already thinking value and perhaps also to others who wish to make a paradigm change from cost to value-based decision making.

2.4 Bayesian networks applied to decision-making in software engineering

Bayesian networks have been applied in many different research fields, which also include software engineering. One of the first advocates of their use in software engineering have been Fenton and Neil (Neil and Fenton 1996), who have made significant contributions in a range of areas such as quality prediction (e.g. Neil et al. 2003), defect prediction (e.g. Fenton and Neil 1999) and risk assessment (e.g. Fenton et al. 2004), and in the use of BNs applied to other fields too (e.g. healthcare). They have also proposed their own BN software that is described as a decision support and risk analysis tool. The essence of Fenton and Neil's argument towards the use of BNs to support decision-making is that BNs are models that enable the representation of the uncertainty inherent to complex domains, the important variables that should be considered when making decisions in such domains and the cause-and-effect relationship between those variables. Once a BN model is built and validated, it can be employed for predictive as well as diagnostic reasoning and can also provide the means, via performing a range of what-if scenarios, to identify possible problems, avenues for improvement, risks and potential benefits.

Since Fenton and Neil's pioneering work, many others have followed on their footsteps and employed BNs for decision-making in software engineering. We have identified four secondary studies investigating the use of Bayesian networks in software engineering; three targeted at specific research areas, such as effort estimation (Radlinski 2010), quality prediction (Tosun et al. 2015) and requirements engineering enhancement (del Aguila and del Sagrado 2015). One, by Misirli and Bener (2014), looked at the use of BNs throughout all areas in software engineering. This is the only one that we will detail further herein, as it is broader in approach, when compared to the other three secondary studies. Note that we have also identified another paper that seemed to focus on BNs applied to decision-making in software engineering (Nageswarao and Geethanjali 2016); however, it does not provide either a survey of previous work or any other sound research contribution, so it will not be included in the discussion that follows.

Misirli and Bener (2014) identified 117 papers, distributed in the following knowledge areas:

- Software quality—54 papers (46.15%)
- Software engineering management—31 papers (26.5%)
- Software design—7 papers (5.98%)
- Software testing—7 papers (5.98%)
- Software requirements—5 papers (4.27%)
- Software construction—5 papers (4.27%)

- Software maintenance—3 papers (2.56%)
- Software engineering tools and methods—2 papers (1.71%)
- Related disciplines—2 papers (1.71%)
- Software engineering process—1 papers (0.85%)
- Software configuration management—0 papers

Some of their findings, which are directly relevant to this research, were as follows:

1. Close to "72% of BN applications in software engineering are proposed in the area of software quality, i.e. predicting faults and failures, and in software engineering management, i.e. project and resource planning. The use of BNs in the areas of software testing, design and construction have become more popular after the 2000s" (Misirli and Bener 2014). None of these applications were targeting at value-based prediction, thus providing further evidence about the novelty of the proposal we are putting forward herein for value estimation.
2. A total of 55% of the primary studies included in their SLR uses expert knowledge as the primary source for defining the network structure. It has also been the first author's experience to build effort estimation BN models in constant collaboration with experts. Within the context of the value estimation BN model-building approach detailed herein, we propose using a mix between expert-based and data-based sources for model building, mainly due to the extremely busy schedules of key stakeholders in all of our four industrial partners.

As pointed out by Darwiche (2010), BNs have been now used in a wide range of fields due to its strength in representing uncertainty and cause-and-effect relationships. However, there are still several challenges to tackle when building such models, one of which applies to eliciting probabilities in child nodes' CPTs. This paper also provides a possible solution to this problem, which employs a previously used approach when building BNs from domain experts, in combination with data gathered from decision-making meetings via the Value tool.

3 The VALUE framework

The main goal of the VALUE framework is to support companies improve their value-based decision-making. It has five main parts (shown in Fig. 3 as A1 to A5), which jointly aim to (i) elicit the fundamental value factors from key stakeholders; (ii) use the fundamental value factors with a bespoke software tool—Value tool, to support and improve decision-making within the context of software/software-intensive products and (iii) also provide a value estimation model (integrated in the Value tool), built using data on previous decisions gathered using the Value tool, input from key stakeholders and a semi-automatic algorithm for probability generation, as additional support to improving decision-making.

Figure 3 provides an overview of the VALUE framework's main parts—A1 to A5. When a company first employs the framework, some of these parts are very likely to be carried out sequentially at first (e.g. A1 > A2; A2 > A3 > A4 > A5). However, we are not implying that they should always be carried out in a waterfall-like way. For example, as the use of the Value tool progresses, some parts may be carried out again in a non-sequential order (e.g. carry out A1, followed by A2 and then A1 back again; or carry out A3 to A5 and then A1 again), and so forth.
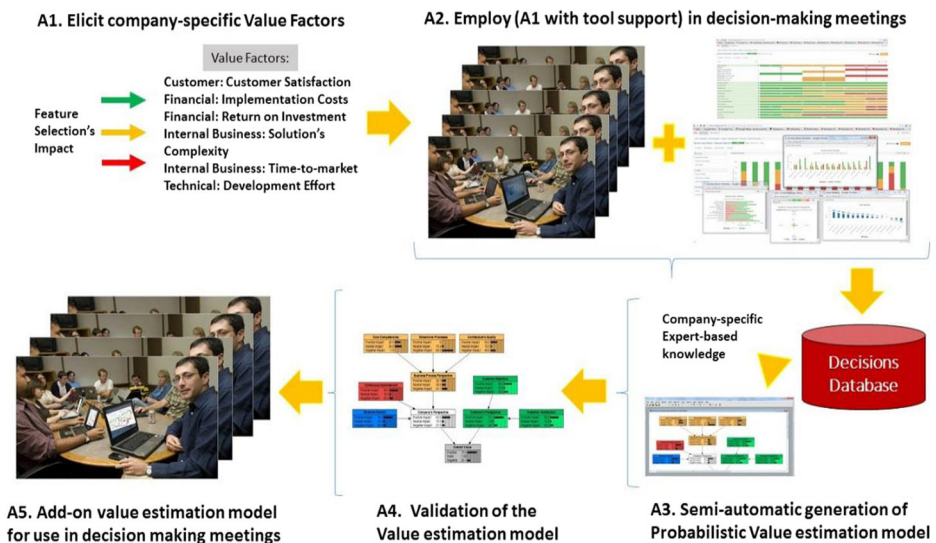
Fig. 3 The VALUE framework and its five main parts

– Elicit company-specific value factors

This part represents the elicitation of key stakeholders' value propositions, so to make them explicit and later to be aggregated and employed as part of their decision-making process. Within the context of this research, such elicitation can be done using qualitative research methods, in either of two ways:

First option:

1. Individual interviews with key stakeholders, which are later transcribed and analysed using Grounded Theory principles (Urquhart 2012); these analyses are also sent back to the interviewees for comments.
2. Focus group meeting(s) with the key stakeholders (same ones who participated in step (a)) to agree upon a combined set of value factors to use for decision making. During such meeting(s), stakeholders also define how these factors are to be measured with regard to the expected effect that the selection of features/ideas/etc. will have upon each value factor.

Second option:

3. Focus group meeting(s) with the key stakeholders to jointly elicit and agree upon a combined set of value factors to use for decision making. During such meeting(s), stakeholders also define how these factors are to be measured with regard to the expected effect that the selection of features/ideas/etc. will have upon each value factor.

Section 4 will detail how A1 has been carried out with company A.

– Employ (A1 with tool support) in the decision-making process

This part represents the use of a Web-based tool (Value tool) aimed to support decision-making. The Value tool was developed in collaboration with all four industry partners, in a co-creation approach via a design science research method (Hevner et al. 2004). Section 5 details the Value tool through the description of a pilot study carried out with company A.

Within the context of A2, there have been a number of usability measurement case studies (Freitas et al. 2016), and pilot studies with different industry partners (one of which is detailed in Sect. 5). Section 5 also presents a detailed testimonial from one of company A's key stakeholders, based on their experience using the Value tool for decision-making at company A.

– Semi-automatic generation of a probabilistic value estimation model

This part entails the use of data in the decision database, input from key stakeholders and a semi-automatic algorithm for probability generation, to build and validate a company-specific Bayesian network (BN) model to be used to estimate the overall value of a decision. Note that in order to build a value estimation BN model, a company will need to have access to experts in BNs to guide through steps A3 to A5.

The overall process we use when building BN models is called expert-based knowledge engineering of Bayesian networks process (EKEBN) (see Fig. 4) (Mendes 2012; Mendes 2014) and has been adapted from Woodberry et al. (2004) in order to provide a set of steps that are achievable and meaningful to the practitioners participating in the process. In Fig. 4, the arrows represent flows through the different sub-processes, which are depicted by rectangles. These sub-processes are executed by people—the knowledge engineer and the domain experts (orange rectangles), by automatic algorithms (white rectangles) or via a combination of people and automatic algorithms (orange rectangle with bottom left white triangle). Within the context of this research, the first author is the knowledge engineers (KEs), and a company's key stakeholders are the domain experts (DEs). The three main steps within the EKEBN process are the structure building, uncertainty quantification and model validation. This process iterates over these steps until a complete BN model is built and validated.

Note that when building BNs, they can be built solely from data, from domain expertise or using a combination of both. In our previous work (Mendes 2012; Mendes 2014), we built such models based solely on domain expertise because the companies participating in that research collaboration were able to provide us with as much of their time as needed in order to co-create the effort estimation BN models; however, within the context of the research detailed herein, all industry partners, including company A, have very clear limitations on their time availability, thus we had to look for alternative solutions to build the value estimation BN models; thus, we are using a hybrid approach (data + knowledge) to BN model building (Mendes and Mosley 2008; Misirli and Bener 2014).

Each of the three main steps part of the EKEBN process is detailed next.

3.1 Structure building

The structure building step represents the qualitative component of a BN model, which results in a graphical structure that represents the explicitation of tacit knowledge from one or more DEs. Within the context of this research, such structure represents (i) the value factors elicited from key stakeholders, representing the factors they use when making decisions relating to software/software-intensive products; (ii) factors' causal relationships and (iii) how value
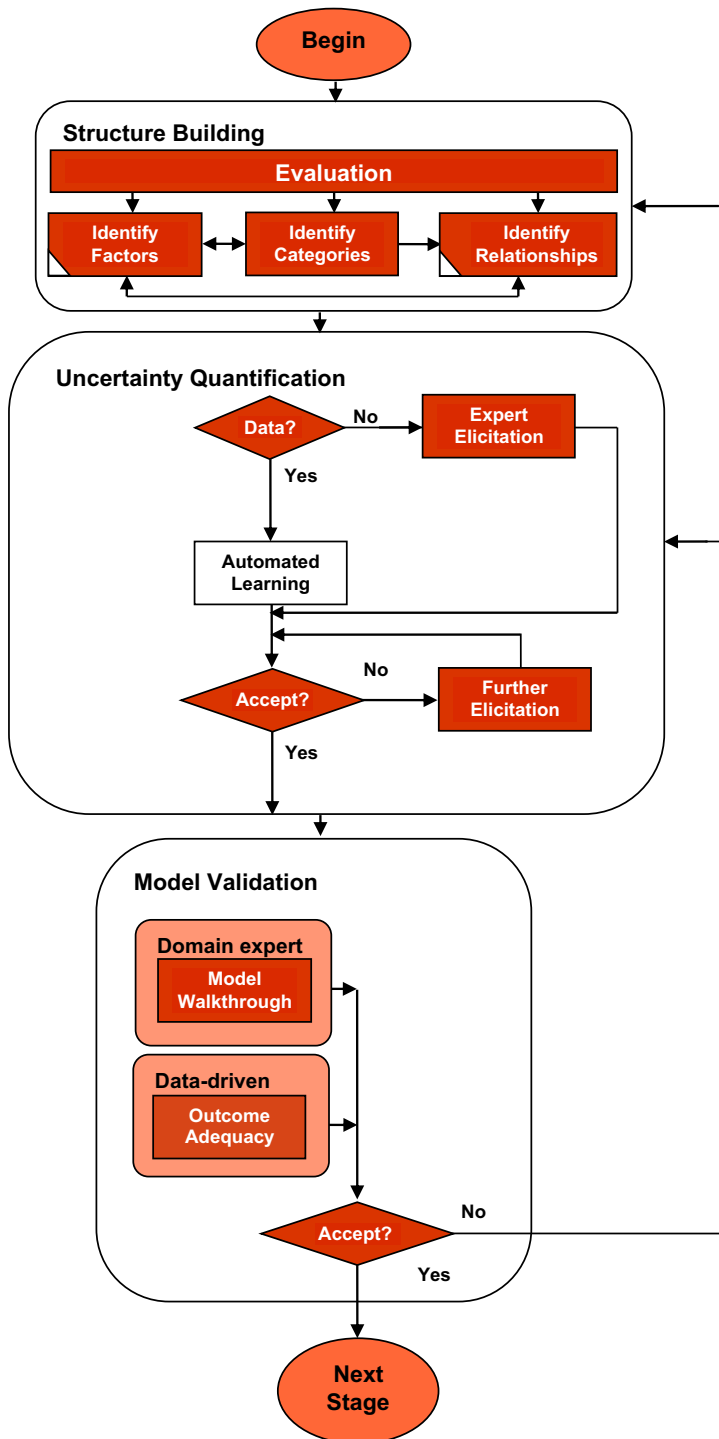
Fig. 4 Expert-based knowledge engineering of Bayesian networks

factors will be measured, i.e. the states that each factor will take (e.g. positive impact, neutral impact, negative impact).

With regard to causal relationships, they will be identified in two different ways, depending on the company's preferred choice:

1. Focus group meeting with key stakeholders, using DEs' domain expertise to decide upon causal relationships. This is the same approach used by Mendes (2012, 2014).
2. Available data from past decision-making meetings (stored in the Decisions Database—see Fig. 3), which is used to learn the relationships between value factors, via the same approach by Mendes and Mosley (2008). The Decisions Database stores the decisions made by the key stakeholders, when using the Value tool for decision-making. These data-driven relationships are then checked by the key stakeholders, who can also add/remove/modify relationships.

During this sub-process, the stakeholders (DEs) also define which value factors are to be parent nodes and which ones are to be child nodes. The end result from this sub-process is a graph containing the value factors, their states and cause-and-effect relationships. We chose to use very simple relationships in our detailed example in Sect. 6, keeping the same approach previously employed when using a very simple prepared value BN model during meetings with all the industry partners. Further details are given in Sect. 6.

3.2 Uncertainty quantification

The uncertainty quantification step represents the quantitative component of a BN, where conditional probabilities corresponding to the quantification of the relationships between factors (Jensen 1996) are obtained. Within the context of the research detailed herein, there are two sub-processes that are used in order to obtain the probabilities:

1. Obtaining directly from the data stored in the Decisions Database (see Fig. 3). This applies to all the priors for the value factors that were chosen to be parent nodes by the DEs during the structure building step.
2. Use of an additional algorithm—weight sum algorithm (detailed in Sect. 6), in combination with input from DEs to provide a solution to the semi-automatic generation of probabilities for the child nodes part of the value estimation BN model. Details on how the WSA can be applied are also given in Sect. 6.

The next step, model validation, corresponds to part A4 on the VALUE framework, detailed next.

– Validation of the Value estimation model

3.3 Model validation

This step validates the BN model that results from the two previous steps and determines whether it is necessary to re-visit any of those steps. Two different validation methods are

generally used—model walkthrough and outcome adequacy (Mendes 2012; Mendes 2014). They are briefly presented below and detailed in Sect. 6.

Model walkthrough, which is a method employed in numerous fields such as medicine, decision analytic modelling and social sciences (Kopec et al. 2010; Sargent 2013; Weinstein et al. 2003; Penaloza et al. 2015) and aims to obtain a subjective assessment on how good stakeholders "feel" a model is, at face value, with regard to providing/measuring what it is supposed to provide/measure (Drost 2011). Herein, this means that if the BN's outputs for overall value (probabilities given to each of the three states positive, neutral and negative) seem plausible to stakeholders, for all the different what-if scenarios they use, this would suggest that the BN model's output 'at face value' matches what these stakeholders 'feel' should be the case. However, note that a model that has face validity does not mean the same as a model that has been 'shown/demonstrated' to work. This is why we have an additional method also being used—outcome adequacy is explained next.

Outcome adequacy uses a validation set containing a subset of real case scenarios from past decision-making meetings not used for model building, to assess, for each case, the frequency with which the state (e.g. positive, neutral, negative) for the BN model's node overall value with the highest probability matches the state identified via vote counting for that case. Whenever the model does not comply with the vote counting result, it is manually re-calibrated.

– Add-on value estimation model for use in decision-making meetings

This part corresponds to the use of the value BN model as part of the decision-making process and also to carry out focus-group meetings with key stakeholders to obtain feedback about its use and perhaps need for modifications/recalibration.

As data is gathered in the Decisions Database, it is also important to specify with a company checkpoints in which more recent data may be used to recalibrate the BN model. Such decisions will also be part of our ongoing discussions with the industry partners and one of the topics of our future work.

This paper presents empirical results for parts A1 and A2 (see Sects. 4 and 5 respectively) and uses a detailed example partially based on the data on decisions gathered from A2 to describe parts A3 to A5 (see Sect. 6).

When carrying out each of the five parts abovementioned, one must have in mind whether it contributes to achieving the Framework's three aims. With regard to aim (i), Sect. 4 (in Fig. 3(A.1)) details the different research methods we have employed to elicit value factors and also validate the elicitation results. Our original goal was to identify the most adequate value factors used by company A's key stakeholders when making decisions relating to features selection for product B. With regard to aim (ii), Sect. 5 details a pilot study (in Fig. 3(A.2)) carried out with two of the three key stakeholders, using the Value tool, the value factors previously identified via aim (i) and two of product B's features. Section 5 also provides a detailed testimonial from one of company A's key stakeholders in relation to gained benefits from using the Value tool + embedded value factors. Finally, as for aim (iii), we have provided in Sect. 6 (Fig. 3(A.3–A.5)) an example BN for company A, based both on real data on decisions gathered by the Value tool during the pilot study (Sect. 5) and other example data. However, we will only be able to provide further details and a discussion on how this particular research aim was achieved once we have gathered more data on decisions from company A's decision-making meetings and have also carried out focus-group meetings to identify parent

nodes, relationships and to obtain input to the WSA method. This is one of the topics of our future work.

The next sections will explain in detail parts A1 to A5 of the VALUE framework.

4 Explicitation of company-specific value factors

This section details the steps we followed to explicitate the value factors used by company A's three key stakeholders when selecting features to be added to one of their most important products. Features are chunks of product functionality that deliver business value; within the context of company A, value factors represent any kind of aspect, criteria or consideration that are used by the stakeholders when deciding whether a feature should be included in a release (e.g. financial, customer related, technical, organisational or any other aspect that is relevant for the different stakeholders participating in the decision-making process).

company A is a large software company with offices in 20 countries and presence in more than 100 locations worldwide. It develops several products (product family) that offer different services to its more than 200 customers. Their focus within our research collaboration is to improve decisions relating to features' selection for one of their key products—product B henceforth. There are only three people in company A who participate in product B's strategic decision-making. They are the key stakeholders within their business context, and they were the ones who participated in knowledge elicitation activities and in the pilot study detailed in Sect. 5.

Regarding their development process, they follow an agile-based software development process characterised by iterative development. The iterative nature of the process is important because company A works in a turbulent environment in which living with/being able to adapt to continuous business changes is one of their main challenges. The feature selection process is also iterative, encompassing many discussion loops between activities.

product B's releases are delivered approximately every few months and encompass the following activities: (i) the main characteristics of the product are selected based on certain criteria; (ii) the design phase begins and feedback loops are established between product definition, product design and product development.

Materials such as power point presentations and customer segment studies as well as tools for managing feature backlogs such as JIRA are used during this process.

We employed a qualitative approach to explicitate (elicit) value factors, because it provided a detailed exploration of the phenomenon under study, gaining access to different stakeholders' views in their own context and using their own natural language.

Value factors were elicited in two phases:

Phase I: we conducted individual semi-structured interviews with each of the three key stakeholders abovementioned. Each interview lasted for about 1 h. Once transcribed, they were analysed using coding techniques and resulted in an initial list of value factors.

Phase II: we complemented the elicitation process with focus group meetings aimed to agree upon the core set of value factors as per all stakeholders' viewpoints.

Each is detailed further next.

Those three key stakeholders have a wide experience on both the business domain and conducting feature selection tasks. They use their own criteria and also centralise information needed when selecting features from other stakeholders, both inside and outside the organisation, such as customers, development teams, and maintenance and operations teams. Thus, they represent the voice of the other stakeholders during feature selection meetings and are, in

the end, the accountable persons for deciding upon the features to be included in the next product release.

– The first interviewee is product B's manager (PM). As product manager, he has a broad visibility of the entire development process. However, his duties primarily focus on strategic product decisions from business goals and product roadmaps to feature prioritisation. He is the main responsible for deciding upon product B's features and, ultimately, he is the one taking the final decision. He has worked in company A for 4 years and performing the role of product manager for the past year. He has 16 years of experience in the software domain.
– The second interviewee is responsible for product B's marketing (PMk), the role he has performed for the past 18 months. As responsible for product marketing, he mainly focuses on activities related to the product's roadmap, defining features and participating in the feature selection/prioritisation process. He is responsible for defining the product's target customer segment and for outlining key value propositions to be communicated towards that target segment. He has worked in company A for 11 years. Prior to his current position, he was product B's product manager (role taken over by the first interviewee). In addition, he is responsible for the marketing of other products that are part of the same product family. He has been working in the software domain for 15 years.
– The third interviewee is the product manager for the entire product family to which product B's belongs (PF), a role he's had for the last 4 years. His duties primarily focus on strategic decisions for the product family in order to ensure that the different products in the same product family are well-aligned and are also in accordance with the company's business strategy. He has worked for company A for 4 years and in the software domain for 15 years.

We designed an interview script containing three sections: (1) warm-up questions including demographic and context setting questions, such as a description of the interviewee's responsibilities related to feature selection; (approximately; 2) value factor elicitation questions in order to ascertain the set of value factors that the interviewee considers when deciding upon a feature. Particularly, we asked the interviewee to think about the last decision-making meetings and the information/knowledge that he used/brought to the meeting in order to determine the set of features that should be included in the next product release and (3) wrap-up questions to check for any missing relevant topic that the interviewee would like to discuss prior to concluding the interview. Although planned to last for 60 min, all three interviews lasted a bit longer (99, 77 and 63 min respectively). Interviews were voice recorded, transcribed and then analysed using the Grounded Theory principles (Urquhart 2012).

Although a pre-study of the related literature was conducted, we set aside theoretical ideas in order to make sure that we had no pre-conceived theoretical notions and let the value factors emerge from the collected data. The interview transcripts were subject to an iterative multi-step process of data analysis in a systematic manner using coding techniques.

We followed Glase's (1978) stages of data analysis and used the coding guidelines provided by Urquhart (2012).

The analysis was conducted by the second author and the set of factors identified in the analysis was discussed with the first author in order to improve the validity of the analysis. Each interviewee received a report containing a list of value factors identified in their interview together with their descriptions and was asked for feedback.

Note that a detailed explanation of the method employed herein is given in Rodriguez et al. (2016), using results from interviews carried out with another industry partner.

To illustrate the analysis, we present a single example of quotations from interview transcripts that resulted in the value factor "End-user satisfaction" (see Table 1).

End-user satisfaction:

"*The customer impact means to me for example customer satisfaction measured in the NPS (Net Promoter Score) scoring. So that's one factor […]. If the consumer fails or struggles or is displeased of installing our product or using our product they are more like to switch and turn and stop at the next billing period.*"—Product manager.

"*Clearly we want to be consumer-led, and how do we build great products for the consumers […].And then of course to the extent that we've already defined that these could be the features like if there's a mock-up available that we can show to the consumers they can click on it, then of course we can have the dialogue that okay out of these ten features, what did you find to be the most interesting ones, which ones you would not use at all, and then of course that helps us scope then the upcoming release that okay seems like out of these ten, these three are the most important ones and then that frees again resources to do something else if the remaining seven were not interesting for the target segment.*"—Product family product manager.

"*In general, I think the consumer value and the experience should have the most factors so that that area would be the kind of something that we have emphasis on.*"—Product marketing director.

We identified 13 main value factors and 26 sub-factors from the first interview (product manager), 18 main factors and 12 sub-factors from the second interview (product marketing director), and 12 main value factors and 14 sub-factors from the third interview (product family manager).

After aggregating all the value factors (and eliminating duplicates), we conducted focus group meetings with the key stakeholders, where we (i) reviewed each value factor, determining whether it was really important to the stakeholders; (ii) dropped those factors that were considered less important and (iii) discussed if the factors and their categories had a clear meaning and were conveniently named. The focus group sessions focused on eliminating ambiguity and agreeing upon a common terminology; (iv) checked for any categories that could be combined and (v) ensured that all the relevant value factors were included.

The first and second authors acted as facilitators during these sessions. Three focus group sessions were conducted until a mutual satisfactory agreement was achieved. Due to calendar restrictions, not all the three stakeholders participated in every session; however, they were as accommodating as possible, depending on their availability.

Table 1 presents the 21 value factors agreed upon, arranged into five different groups also determined by the three key stakeholders: customer satisfaction, business value, cost efficiency, market competitiveness and technology and architecture. Factors belonging to the customer satisfaction group make reference to aspects that are relevant from a customer's perspective. The business value group comprises factors that refer to the impact that the implementation of a feature would have to the wealth of the company, mainly in monetary terms, in the long run. Cost efficiency factors focus on the several costs that implementing a feature would entail. Market competitiveness is composed of value factors with regard to the competitiveness of the product in the market in terms of added value compared to similar products. Finally, the technology and architecture group centres around technical aspects such as existing technical capability to implement a feature.

Table 1 Value factors

Group	Name	Description
Customer satisfaction	1. End-user experience (ease of use)	The impact that the implementation of the feature will have upon user experience and usability (e.g. make the end-user to feel happy, feeling that they succeeded in the process of using the product and they have achieved what they wanted to do).
Customer satisfaction	2. End-user satisfaction (value delivery)	The impact that the implementation of the feature will have upon end-user satisfaction (e.g. increased perceived value of the product and/or the feature meets the functionality that the end-user is expecting, etc.).
Customer satisfaction	3. Operator's satisfaction	The impact that the implementation of the feature will have upon operator satisfaction (e.g. increased perceived value of the product, and operator appreciation/willingness to have the feature).
Customer satisfaction	4. Protection promise	The impact that the implementation of the feature will have upon security and privacy of the user and the data (e.g. improving protection against existing threats, responding to new threats, etc.).
Customer satisfaction	5. Security exciter (feature's visibility)	The impact that the implementation of the feature will have upon making security visible and tangible to the end-customer. Related to feature marketability (if the feature is implemented, level to what the feature will be visible to the end user and helps him/her to understand security, level to what the feature can be sold and is marketable).
Business value	6. Company's long-term strategy	The impact that the implementation of the feature will have to the company when considering its lifetime value and cost (looking at a 2- to 3-year investment horizon, where a long-term thinking takes into consideration the operational aspect where our production and hosting costs go up by 5% per subscriber because of the new feature).
Business value	7. Drives subscriber growth	The impact that the implementation of the feature will have upon the amount of product's users and subscribers (e.g. acquiring new subscribers or minimising the churn).
Business value	8. Financial value in direct consumer channel (OIT)	The expected business value of the feature (order intake/sales) in the direct consumer channel (both within the target segment and other segments).
Business value	9. Financial value in operator channel (OIT)	The expected business value of the feature (order intake/sales) in the operator channel.
Business value	10. ROI	The impact that the implementation of the feature will have upon the product, when we consider the estimated worth brought in by the feature against the costs needed to development the feature.
Cost efficiency	11. Delivery cost	The impact that the implementation of the feature will have upon the product's delivery costs (reduces delivery costs vs. increases delivery costs).
Cost efficiency	12. Maintenance cost	The impact that the implementation of the feature will have upon product's maintenance costs (reduces maintenance costs vs. increases maintenance costs).
Cost efficiency	13. Operational costs	The impact that the implementation of the feature will have upon the product's operational costs (reduces operational costs vs. increases operational costs).
Cost efficiency	14. Product's development effort	The impact that the implementation of the feature will have upon the effort for completing the release (e.g. feature story points).

Table 1 (continued)

Group	Name	Description
Cost efficiency	15. Support cost	The impact that the implementation of the feature will have upon the product's support costs (reduces support costs vs. increases support costs).
Market competitiveness	16. Product's competitiveness from end-user perspective	The impact that the implementation of the feature will have upon the product's competitiveness from the customer perspective (e.g. being competitive in reviews, etc.).
Market competitiveness	17. Product's competitiveness from operator channel perspective	The impact that the implementation of the feature will have upon the product's competitiveness from the operator channel perspective (e.g. position in terms of the operator channel).
Market competitiveness	18. Product's differentiation/-innovativeness	The impact that the implementation of the feature will have in supporting the product's differentiation, when compared to similar products in the market (e.g. innovation, creating things differently).
Market competitiveness	19. Time-to-market	The impact that the implementation of the feature will have upon the current release's time-to-market (is the release going to be out in the next 3 months or in the next 12 months?)
Technology and architecture	20. Technical efficiency enabler	The 'feature' improves the technology and architecture enabling, e.g. future cost saving, creation of new products/features on top of it, can be reused by other products/services, refactoring.
Technology and architecture	21. Technical feasibility	The impact that the implementation of the feature will have upon the company's existing technical capability to implement the feature. A positive impact would mean that the feature is technically feasible; a neutral impact would mean that there is a 50/50 chance that the feature is technically feasible; a negative impact would mean that the feature is not technically feasible at the moment.

Note that these factors' definitions were recently revisited by those three key stakeholders during internal meetings, so Table 1 shows the most up-to-date descriptions.

In addition, the key stakeholders also agreed upon a way to measure each value factor. They chose 'impact', measured using a three-point ordinal scale—positive, neutral or negative. This measure was defined for later use with the Value tool during a pilot study attended by two of the three key stakeholders (Sect. 5). The impact measure's interpretation is as follows:

If a given feature (e.g. *Feature A*) *is implemented in product B's next release*, *what is the impact* (*positive/neutral/negative*) *that its inclusion in product B will have upon each of the value factors* (*e.g. customer satisfaction*)?

Some value factors may seem to mean the same, when taken out of company A's and product B's context (e.g. end-use experience and end-user satisfaction). However, one relates to the experience of the end-user using product B in terms of usability and "what the feature will make the end-user to feel" (end-user experience), and the other refers to the general satisfaction that the end-user will get from using product B in terms of whether the product meets the functionality that the end-user was expecting (end-user satisfaction). Similarly, maintenance, operational and support represent different types of cost in our company case and an individual value factor was created for each of them (value factors 13, 14 and 15).

We would like to reiterate the importance of recruiting the right stakeholders, who are referred to as success-critical stakeholders in the VBSE context (Boehm 2003; Biffl et al. 2005), for taking part in both the interviews and the focus group sessions. Within the context of company A, these three key stakeholders lead very busy lives commonly occupied with company strategic meetings, travelling worldwide visiting customers and meeting development teams to make sure that they properly understand the vision of the product.

Such constraints prompted us to conduct the elicitation process in two steps, so to include some flexibility into the process. Individual interviews were easier to arrange as they do not require the synchronisation of the participants' agendas. Moreover, current video-conference systems very well support the interviewee to be in a different location. Out of our three interviews, one was conducted face-to-face (marketing director) whilst the other two were video interviews (product manager and product family manager). Further, having a preliminary list of value factors representing each individual mental model as a starting point for conducting the focus groups relaxes the requirement for having all the stakeholders in each focus group session.

5 Tool support for value-based decision making

This section describes a Web-based tool (Value tool) and the results from a pilot study carried out in company A, with the participation of two of their three key stakeholders.

The motivation for designing the Value tool within the scope of this research is twofold:

1. First, to be used by our partner companies (and potentially also other ICT companies) to improve their decision-making relating to the selection of decision items (e.g. features, ideas, requirements) by different key stakeholders (e.g. product managers, product owners) using a decision criteria (value factors). As will be shown later, this tool stores the combined set of value factors that had been previously explicitated via interviews and focus group meetings (or only focus group meetings) with the key stakeholders participating in the decisions for which the tool will be used. Further, the tool also enables the recording of individual as well as group decisions relating to the decision items being selected, where such recording also includes the rationale used along the way. The individual assessment relating to one's decisions about the best set of decision items based on the chosen decision criteria also represents the externalisation of tacit knowledge; here, such externalisation occurs at the individual level; later, using a tool functionality called 'the dashboard', the different key stakeholders can use a number of different visualisations that aggregate their individual externalised decisions. This dashboard enables the combination of the various individual decision-making externalisations and can also foster the internalisation of knowledge as stakeholders jointly discuss the best set of decision items to select during that decision-making meeting.
2. Second, to gather data throughout several decision-making meetings, using as basis the value factors jointly agreed upon by the key stakeholders and relating to the selection and ranking of decision items (e.g. features, requirements, ideas). Such data will be later on used to partially build a value estimation BN model for the company (company-specific model). The value estimation functionality will also be added to the Value tool, to be available for use by companies as an additional functionality. Details on this step are given in Sect. 6.

The Value tool (Fig. 5) was co-created with all four industry partners, using a design science approach (Hevner et al. 2004) through an iterative process involving interviews, elicitation of requirements and the progressive tool implementation and assessment. This was done deliberately in order to provide a solution that would fit into companies' current decision-making processes. Further, the co-creation process was fundamental to understanding companies' current value-based decision-making processes and to offer them with a solution that would match as closely as possible their existing processes. In addition to the feedback received from the industry partners, the Value tool was also used in three different case studies aimed at assessing its usability (Freitas et al. 2016). This was also done in order to mitigate possible barriers to its full adoption in industry. These three case studies were particularly important as they provided very useful feedback to improve its interface so to provide a better user experience.

The Value tool was designed in a way to enable its use in a wide range of decision-making scenarios (e.g. selection of ideas/features/requirements for a product, selection of bugs to fix). This means that the nomenclature used in the tool had to also be more general. For this reason, we adopted the following terms (see Fig. 5): deliverable—used to describe a product, a project, a release, a service etc., anything that represents the deliverable to which decision items relate to. A decision item represents a feature, a requirement, a use case, a bug, an idea or anything that needs to be decided upon by the key stakeholders participating in the decision-making process. A deliverable is associated with multiple decision items. The explicit knowledge from stakeholders

Fig. 5 Value-based decision making through the Value tool
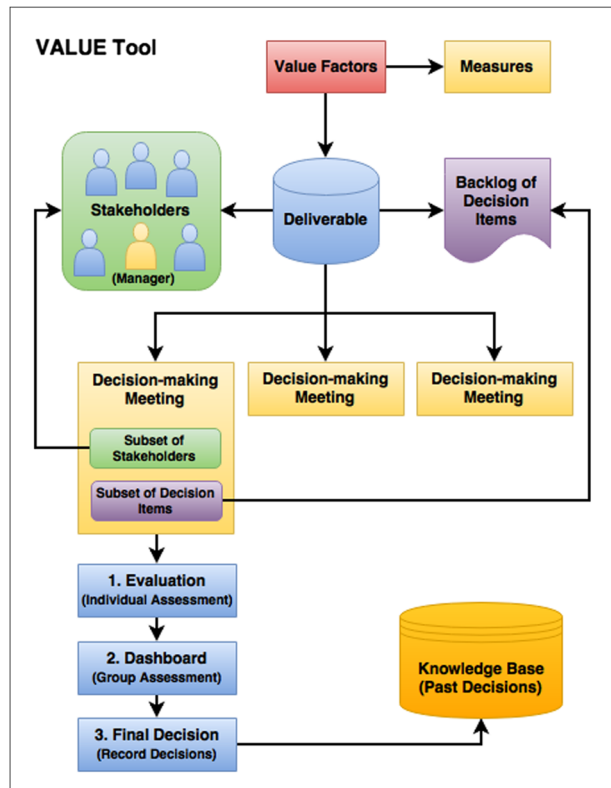
about what they understand as value is represented in the Value tool via a set of company-specific value factors. The identification of value factors is an important step prior to using the Value tool, as detailed in Sect. 4. Finally, each value factor is associated with a measure, which identifies how each decision item, if selected to be part of the deliverable under discussion, will be assessed against each value factor.

As shown in Fig. 5, at the top level, we have a repository of deliverables and each deliverable has a set of stakeholders and decision items. One of the stakeholders is the deliverable's manager and is responsible for setting up in the Value tool the data needed for the decision-making meetings (e.g. loading the information about each feature to be discussed, assigning stakeholders to a decision-making meeting). All decision-making activities are represented in the tool as meetings, i.e. decisions are stored per meetings. This means that a given deliverable may have several meetings associated with it. Each of the meetings is composed by stakeholders, decision items, value factors and measures. During a meeting, stakeholders follow three sequential steps: individual assessment, group assessment and final decision.

Each of these three types of assessment will be explained next using data from a pilot study that took place with one of the industry partners (company A) on the 1st of December 2015. This pilot study was attended by two of the three key stakeholders making decisions relating to product B (see Sect. 4 for further details on the stakeholders and also the value factors that were decided upon). These two key stakeholders were the product marketing director (PMk) and the product manager for the entire product family to which product B belongs (PF).

5.1 Individual assessment

The individual assessment is the first stage part of a decision-making process, as per the Value tool (Fig. 6). Each key stakeholder individually assesses the impact, which in this case is positive, neutral, and negative, which the selection of a given feature (feature 1 in Fig. 6) will have upon each of the value factors. Individual assessments can be done asynchronously prior to a joint decision-making meeting or during the meeting. Stakeholders can also use their laptops/tablets/smartphones to run the value tool, which also adds to the tool's flexibility. Note that the little balloons beside each value factor can also be used to enter the rationale for a given decision, i.e. rationale for choosing either positive, neutral or negative for a given decision item, in relation to a value factor (see Fig. 7). Further, the tool uses a default setting for the measure(s) being employed. Within the context of company A, the default is Neutral, which means that the neutral state is automatically selected for all the value factors, for all the features to be discussed upon in a decision-making meeting. Such functionality enables stakeholders to focus solely upon the value factors that they see fit, given their expertise.

In summary, individual assessments can be done synchronously during a joint decision-making meeting or asynchronously prior to a joint decision-making meeting. Within the context of this pilot study, the individual assessments were done synchronously, as part of the pilot meeting.

5.2 Group assessment

Once all individual assessments are completed, the stakeholder who is the meeting's manager can change the status of the meeting from 'ongoing' to 'analysing'. This means that now, all
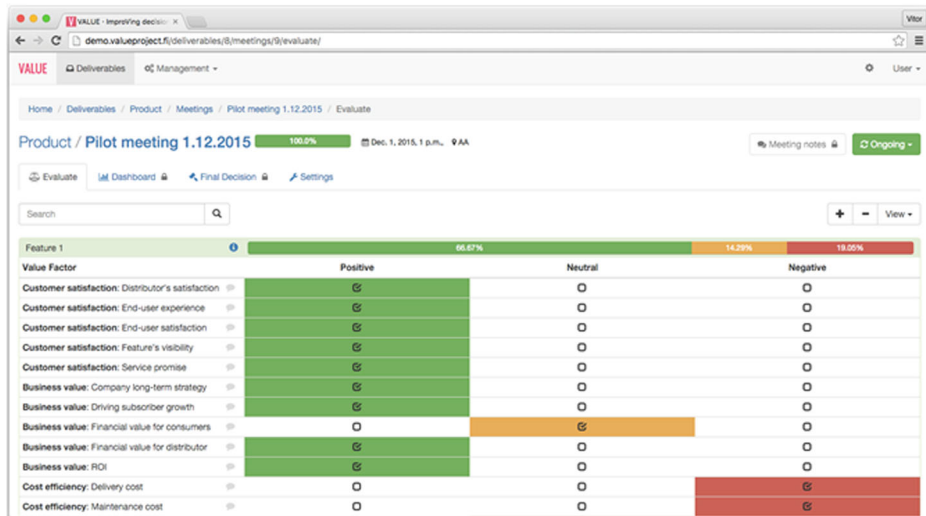
Fig. 6 Individual assessments during a decision-making meeting

stakeholders can see the group's aggregated assessments, via numerous visualisations provided in the Value tool's 'Dashboard' (Fig. 8). The explicit activation of the dashboard was one of the requests made by some of the participating companies, to enable the unbiased individual assessment by the key stakeholders, prior to a joint meeting. Figure 8 shows some of the visualisation options provided. The bar chart shows the aggregated assessments for feature 1 done by the two key stakeholders who attended the pilot study, for each of the value factors. The radar chart shows the same data however now aggregated by value factors' groups. The use of radar charts enables aggregated results to be presented using a higher level of granularity (value factors' groups rather than individual value factors) and was one of the explicit requests made by company A. This particular radar chart shows a high level of agreement on a positive impact on customer satisfaction and a polarised opinion on cost efficiency (50% negative impact, 40% neutral impact and 10% positive impact). The heat map chart, shown in the background of Fig. 8, gives an instant feedback on the value factors where the most positive, neutral and negative impact on feature 1 are; for example, in this case, the most negative impact was related to the value factor 'product's development effort'.

Further, the dashboard provides the means to visualise all the stakeholders' assessments, compare assessments between decision items (see Fig. 9), compare views from different groups of stakeholders, compare the impact of different value factors and create scenarios that group
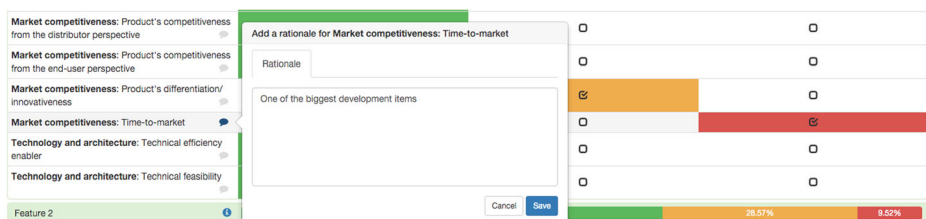
Fig. 7 Rationale of why time-to-market was evaluated as negative
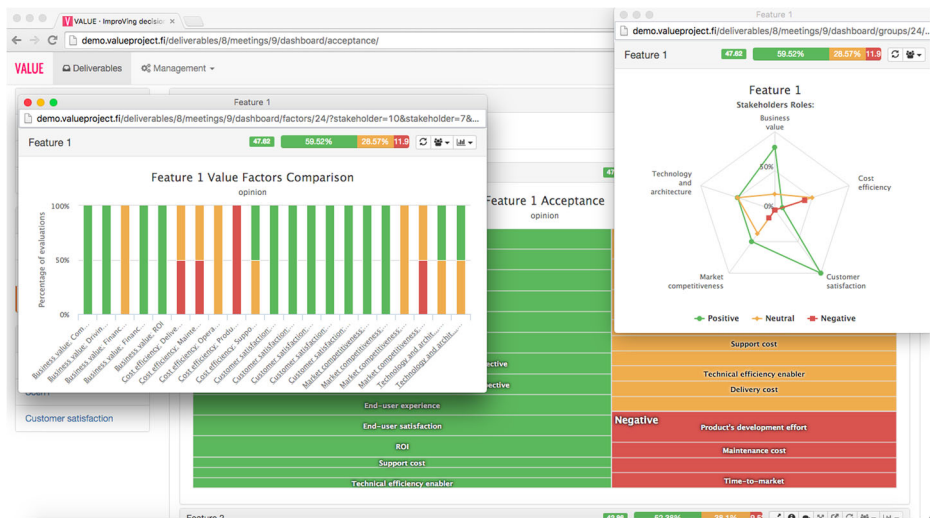
Fig. 8 Some of the data visualisation options provided by the Value tool

decision items together (e.g. a scenario can be a subset of *n* features with the best impact on the revenue growth; another scenario can be the a subset of *k* features with the best impact on the customer satisfaction). Different scenarios can be compared, based on all the visualisations provided in the dashboard. Within the context of this pilot study, we did not use scenarios (although they were also requested by company A to include in the Value tool) because there were only two features being assessed.

The dashboard is useful to guide group discussions as the Value tool can identify conflicting views (e.g. a group of stakeholders assessed a given feature with an overall positive impact on the value factor 'revenue growth' at the same time as another group of stakeholders assessed the same feature with an overall negative impact upon the same value factor revenue growth), thus saving time by providing the means for stakeholders to go straight to the decision items that deserve more attention, in order to discuss the conflicting assessments and make a decision.
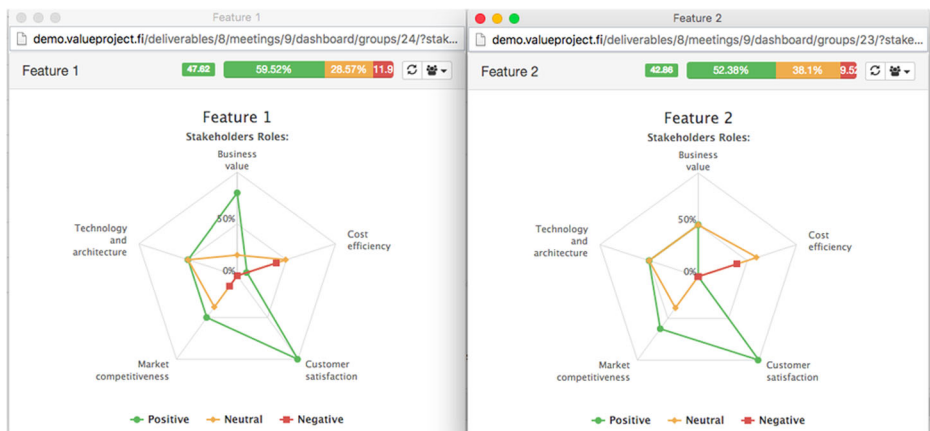
Fig. 9 Features comparison side by side

5.3 Final decision

The final step of a decision-making meeting comprises documenting the group's decision relating to the ranking and selection of the decision items discussed during a meeting (Fig. 10). The final decision screen for the pilot study presented herein includes the assessed features, summarising their overall impact and value ranking. The value ranking is calculated by subtracting the percentage of negative impact from the percentage of positive impact. For example, in relation to feature 1, it has an overall positive impact of 59.52% and a negative impact of 11.9%, giving a value ranking of 47.62 (59.52–11.9 = 47.62). The value ranking can range from −100 to 100 and is useful to compare features as a single number gives an insight about a feature's impact, by looking at the two extreme scale points.

The development of the Value tool started with a simple prototype representing our views of what would be a software solution to tackle the challenges of a value-based approach to decision-making in the software domain. Once this prototype was completed, there were a series of meetings with one of the industrial partners, followed by other meetings with the remaining industry partners. The focus of each meeting was to demonstrate the tool and to obtain feedback on any additional functionality needed, usability improvements and also changes to the existing functionality. In total, 16 meetings took place, accounting to 44 person-hours. All participants were key stakeholders in their respective companies.

After the pilot study detailed herein was carried out, company A underwent an internal re-structuring, which included revisiting their road mapping strategy. Once such re-structuring was finalised, there were meetings between the three key stakeholders (these stakeholders were introduced in Sect. 4), aimed to use the Value tool to assess additional features for the same target product—product B, to check whether the existing value factors needed revisiting and also to scrutinise the Value tool in relation to supporting their decision-making needs. These were internal meetings attended only by the three key stakeholders.

A joint feedback meeting between one of the key stakeholders (PMk) and the first and third authors took place on the 7th of December. The feedback received was very positive and, as a result of their additional meetings using the Value tool, the company has also decided to widen the Value tool's use inside its Oulu branch, with other internal meetings planned to take place soon.

Stakeholder PMk provided us with the following testimonial, via personal communication over e-mail:
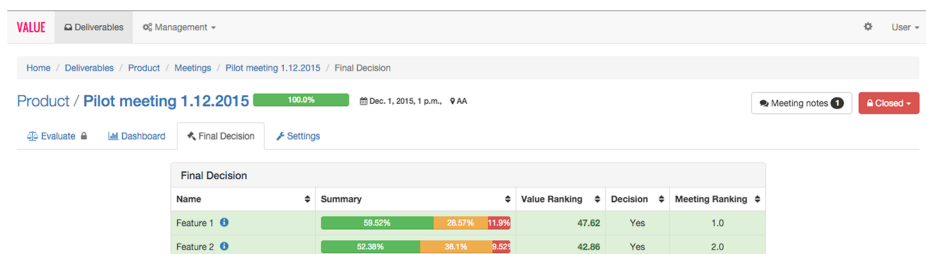
Fig. 10 Final decision screen

"The key benefits of VALUE tool in decision making are as follows:

1. More stakeholders can be involved. It is always challenging to consolidate many opinions into a proper priorities list. VALUE tool enables involving multiple stakeholders in the decision making process. Responses can be viewed by the stakeholder group.
2. Priorities and decision criteria are transparent. Every item is processed using carefully selected decision criteria. This both helps in selecting items on top of the list but also in communicating why certain features are not there.
3. Offline decision making. Stakeholders may not always be available for decision making at a given time. The tool enables collecting feedback from stakeholders at a suitable time. Everyone gets a chance to provide their feedback. This also makes the decision making faster as selecting one suitable time for processing business goals with multiple stakeholders is challenging and meetings tend to go far in the future, e.g. next free common timeslot is in 2 weeks time.
4. Previous decisions are visible. It is often forgotten over time why certain decisions and priorities were set. The tool allows revisiting of those decisions. This helps in understanding what decisions led to what outcome.
5. Learning. The bottom line is to make better decisions and learn while making them. Consolidating different views (multiple stakeholders and different stakeholder roles in the tool), analysing with a broad set of criteria and the ability to revisit the decisions or even processing items again provides a nice platform for decision making."

Several pilots have also taken place with another two industrial partners (both large ICT companies), and their results will be the focus of future publications.

6 Building a company-specific value estimation model

This section details the mechanisms to be used to build and validate company-specific value estimation models for companies participating in our research project and also any other companies. To do so, we will employ a detailed example of a value estimation model for company A, where part of this example uses the value factors detailed in Sect. 4 and real data from the individual assessments by the two key stakeholders who participated in the pilot study detailed in Sect. 5.

6.1 Structure building

As described in Sect. 3, we are employing the EKEBN process to build company-specific value estimation BNs.

The BN model's structure comprises factors and their cause-and-effect relationships. Most of these factors represent the value factors elicited from key stakeholders from the company for which the BN model is to be built. Further, every value estimation model will also introduce an additional factor called overall value, which represents the estimated overall value of a decision scenario. Other additional factors may also be optionally added to a BN's structure, as seen in the example value BN model shown in Fig. 11. This is a common solution when looking to reduce the number of probabilities
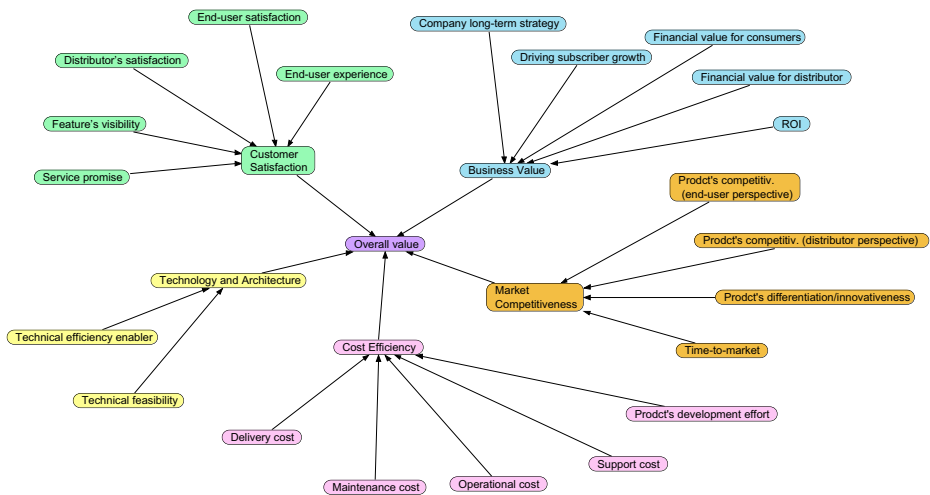
Fig. 11 Example of a value estimation BN model for company A

to be elicited and/or to keep the model's structure simpler (Mendes 2014; Mendes and Mosley 2008).

Figure 11 shows a suggested value BN model for company A, including (i) all 21 value factors already detailed in Sect. 4; (ii) a factor called overall value, which ultimately aims to provide a probability distribution associated with the overall value of a decision and (iii) five additional factors that are not part of the original 21 value factors; these additional factors were introduced for two main reasons: (a) to make the model's structure look simpler, when compared to having all the arcs from the 21 value factors pointing towards overall value (see 12) and b) to resemble the same approach that was used with a very basic BN example presented during our meetings with company A and the other industry partners to explain the use of BNs for estimating the overall value of decisions (this example is shown in Fig. 13).
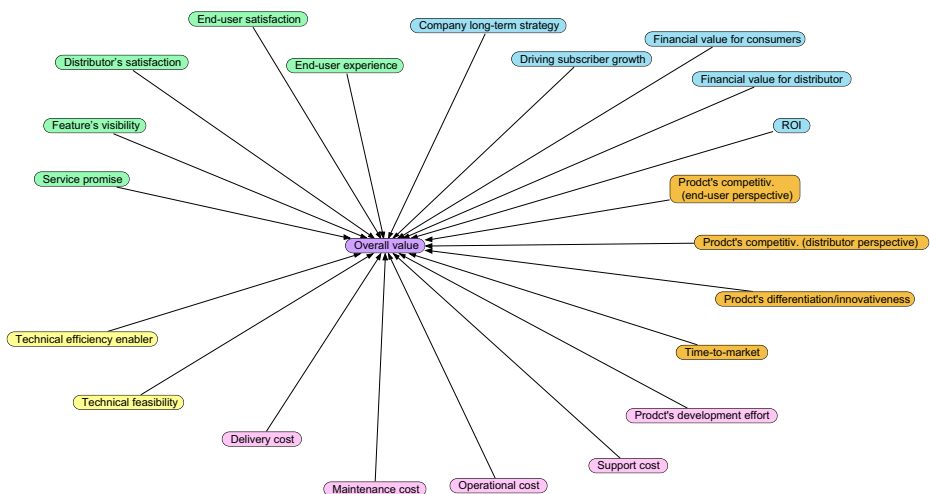
Fig. 12 A value estimation model for company A where all value factors affect directly overall value
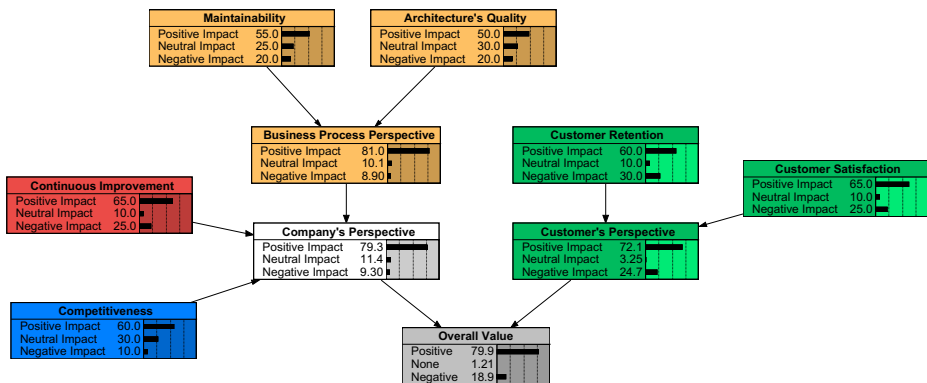
Fig. 13 Prepared value BN model used in discussions with industry partners

6.2 Steps to build the BN model's structure

First choice The BN model's structure (value factors to be parent nodes, value factors to be child nodes, other possible structure 'simplification' nodes; all relationships between nodes) is defined in meetings with the key stakeholders.

    OR

Second choice Use steps 1 to 4 below as a detailed mechanism to build the model's structure. Note that we are using herein as possible states to measure the impact of decisions the same ones that were decided upon by company A, i.e. positive, neutral and negative.

    Step 1. Arrange the data, as shown in Table 2, for the range of decision-making meetings that are to be considered for model-building. All the data comes from the Decisions Database (see Fig. 3 in Sect. 3). Such data will be saved into an excel file, for use by BN structure learning algorithms (Mendes and Mosley 2008).

    In Table 2, we have the following:

        N stakeholders, S_n, that we note: $S_1,…, S_N$.
        M features, F_m, that we note: $F_1,…,F_M$.
        L factors, V_l, that we note: $V_1,…,V_L$.

    OV represents the intermediate overall value. It will be set as positive, neutral or negative depending on the largest frequency of k: positive/neutral/negative, given a stakeholder and a feature. In case there is a tie, k' will be set as the least optimistic state (negative > neutral >

Table 2 Matrix to be used for structure and parameter learning and for model validation

Data for a single meeting					
S_n	F_m	V_1	V_2	V_L	OV
S_1	F_1	k: positive	k: positive	k: positive	k': positive
S_1	F_M	k: neutral	k: neutral	k: neutral	k': neutral
S_N	F_1	k: positive	k: positive	k: negative	k': positive
S_N	F_M	k: positive	k: negative	k: negative	k': negative

positive), so to be conservative. Note that this variable will not be used when building the BN structure but rather to validate the BN model (details given in Sect. 6.3). Also note that our assumption is that every value factor has the same importance.

Table 3 shows the same matrix introduced in Table 2 however using the data gathered from the pilot study at company A that was described in Sect. 5. Note that the data has been transposed for the sake of clarity.

Step 2. Once the matrix is ready, we will use the same solution employed in Mendes and Mosley (2008), where two BN tools—Hugin and PowerSoft—were used for structure building. The variables to use in this step are only the value factors (grey-shaded area in Table 2), and our goal is to identify the suggested relationships (arcs) that both BN tools will provide, based on the input data.

Step 3. Also similarly to what was done in Mendes and Mosley (2008), the suggested relationships will be validated with the company's DEs (key stakeholders) via focus groups meetings. Note that prior to this validation, we will also add to the BN model's structure a node called overall value, so DEs can also determine which value factors should be this node's parents.

Step 4. If the company so wishes, additional factors may also be added to the BN's structure. One example of such additional factors is shown in Fig. 11—factors customer satisfaction, business value, market competitiveness, cost efficiency and technology and architecture.

6.3 Uncertainty quantification

The uncertainty quantification step, as introduced in sub-section 3, is the step in which probabilities are used to quantify the uncertainty in the knowledge domain being modelled. It is done by quantifying the states on each parent node and also the states of each child node, conditional on

Table 3 Matrix with data gathered from the pilot study run at company A

Stakeholders Features	S1 F$_1$	S1 F2	S2 F1	S2 F2
Company long-term strategy	Neutral	Positive	Neutral	Positive
Driving subscriber growth	Positive	Positive	Positive	Positive
Financial value for consumers	Positive	Neutral	Neutral	Neutral
Financial value for distributor	Positive	Positive	Neutral	Positive
ROI	Positive	Positive	Neutral	Positive
Product's development effort	Neutral	Negative	Neutral	Negative
Delivery cost	Negative	Negative	Negative	Neutral
Maintenance cost	Negative	Negative	Neutral	Neutral
Operational cost	Neutral	Neutral	Negative	Neutral
Support cost	Neutral	Positive	Neutral	Neutral
End-user experience	Positive	Positive	Positive	Positive
End-user satisfaction	Positive	Positive	Positive	Positive
Distributor's satisfaction	Positive	Positive	Positive	Positive
Service promise	Positive	Positive	Positive	Positive
Feature's visibility	Positive	Positive	Positive	Positive
Product's competitiveness from the end-user perspective	Positive	Positive	Positive	Positive
Product's competitiveness from the distributor perspective	Positive	Positive	Positive	Positive
Product's differentiation/innovativeness	Neutral	Neutral	Positive	Neutral
Time-to-market	Neutral	Negative	Neutral	Neutral
Technical efficiency enabler	Positive	Positive	Neutral	Neutral
Technical feasibility	Positive	Positive	Neutral	Neutral
OV	Positive	Positive	Neutral	Positive

their parents' states. These probabilities can be (i) automatically learnt from data, using probablities learning algorithms (e.g. (Heckerman and Breese 1996)); (ii) elicited completely via the explicitation of domain experts' tacit knowledge (Mendes 2012b) and (iii) built using a combination of both (Mendes and Mosley 2008; Misirli and Baner 2014).

Note that the way in which we will obtain the probabilities for the parent and child nodes differs as follows:

6.3.1 Parent nodes

The value factors chosen by company A's key stakeholders (domain experts —DEs) to be parent nodes will have their probabilities calculated as follows. Let us assume that one of the value factors selected by company A's DEs to be a parent node is factor V_1 (see Table 2). Therefore, the probability of V_1's state being positive (k: positive in Table 2) is the total number of cells in column V_1 where k: positive divided by the total number of cells in column V_1. This means that all the parent nodes' probabilities correspond to the frequencies with which stakeholders have chosen the states positive, neutral or negative when selecting features and taking into account the impact of their selection upon value factor V_1. We are employing here data that corresponds to DEs' domain expertise as reflected in their decisions (positive, neutral, negative) during decision-making meetings. Figure 14 shows the outcome from this step using real data from the pilot study at company A (based on the data shown in Table 3). Note that all of these calculations are done outside the Value tool.

6.3.2 Child nodes

There are three main challenges relating to the uncertainty quantification of child nodes, as follows: (i) the data gathered in the decisions database (Fig. 3) would need to be large and quite diverse to offer a sufficient range of combinations so to learn probabilities for all the childs' different states.
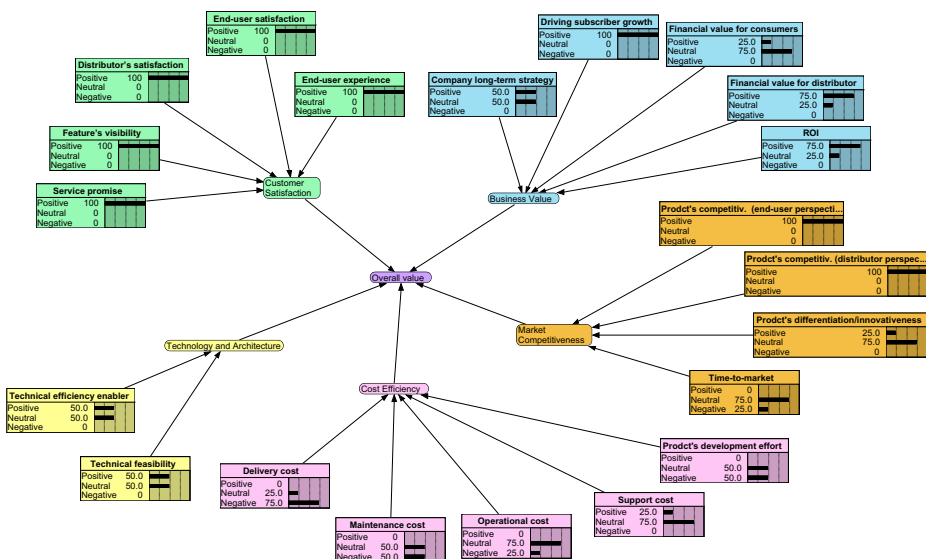
Fig. 14 Frequencies for the parent nodes for company A's example value estimation BN model using real data from the pilot study (see Table 3)

This was the problem faced by Mendes and Mosley (2008) when learning probabilities for child nodes directly from data. (ii) There may be additional factors that the DEs (key stakeholders) may decide to add to the model, which were not elicited previously, and for which there is no existing data in the decisions database (one such example is factor 'Customer Satisfaction' in Fig. 14). (iii) Within the context of this research, none of the DEs from any of the participating companies can commit their time to elicit all the probabilities for the child nodes' CPTs and for other additional nodes' CPTs, as these child CPTs may require hundreds and even thousands of probability quantifications (Jensen, 1996; van der Gaag et al. 1999).

We are using herein the same semi-automatic method (SAM) we have employed previously—the WSA—and also an existing tool that implements this method. The WSA requires very little input from DEs to generate CPTs semi-automatically. Baker (2009) and Baker and Mendes (2010) have investigated the WSA algorithm previously in three different case studies, where the results from applying the WSA to obtaining child node's CPTs were compared to child CPTs for existing BNs being used in practice. The WSA has demonstrated to significantly mitigate the CPT elicitation burden in those case studies, while yielding high accuracy levels. The WSA was also employed in Read (2010) to help with eliciting large CPTs to build an effort estimation model for a medium-size company in New Zealand.

Further, as part of Baker and Mendes' (2010) work, Baker has also implemented a prototype application with their solution, to be used in combination with an existing BN tool (Baker 2009).

This prototype application is being used herein to show how to apply the WSA method to obtain probabilities for child nodes' CPTs semi-automatically. Figure 15 shows a screenshot from the WSA prototype tool providing the following information:

Fig. 15 Applying the WSA to quantify the CPT for the child node cost efficiency

1. Listing on the left-hand side of the BN nodes that are the child nodes for the example value estimation BN model shown in Fig. 14;
2. Displaying the CPT for the node cost efficiency. In this example, a DE would be asked to provide probabilities for the combination of parents' states that are more meaningful to them (compatible), based on their expertise. Node cost efficiency has five parents ('delivery cost', 'maintenance cost', 'operational cost', 'support cost' and 'development effort'). Starting from delivery cost, and working through until reaching development effort, a DE would first choose the states (positive, neutral, negative) that seem more compatible to them and finally enter the probabilities for cost efficiency to be positive, neutral and negative, given each chosen parental configuration. An example of a parental configuration is delivery cost = positive, maintenance cost = positive, operational cost = neutral, support cost = neutral and development effort = positive). This DE also has to enter a weight for each one of the parent nodes (has to add up to 1) indicating their importance upon their child node's probabilities. Once both steps are done, the CPT is generated using the WSA Eq. 2 (see Appendix). Please also see Baker (2009) for further details on the WSA method. Finally, with the WSA, a DE would need to provide only 45 probabilities, rather than 729 probabilities.

Figure 16 shows the example value estimation BN model after exporting the generated CPT to the child node cost efficiency; finally, Fig. 17 shows the example value estimation model after the WSA tool was used to generate all the child CPTs.

As previously mentioned, the main challenge with employing the WSA method is actually the expertise of the DE(s) in being able to identify, with some level of confidence, the parental configurations that are more meaningful in practice, given their experience in value-based decision making. However, as discussed in Sects. 1 and 4, all the companies participating in this research collaboration already employ a value-based decision-making approach, thus suggesting that their level of confidence when choosing parental configurations will likely be high. Our previous experience eliciting parental configurations from experts (Baker 2009,
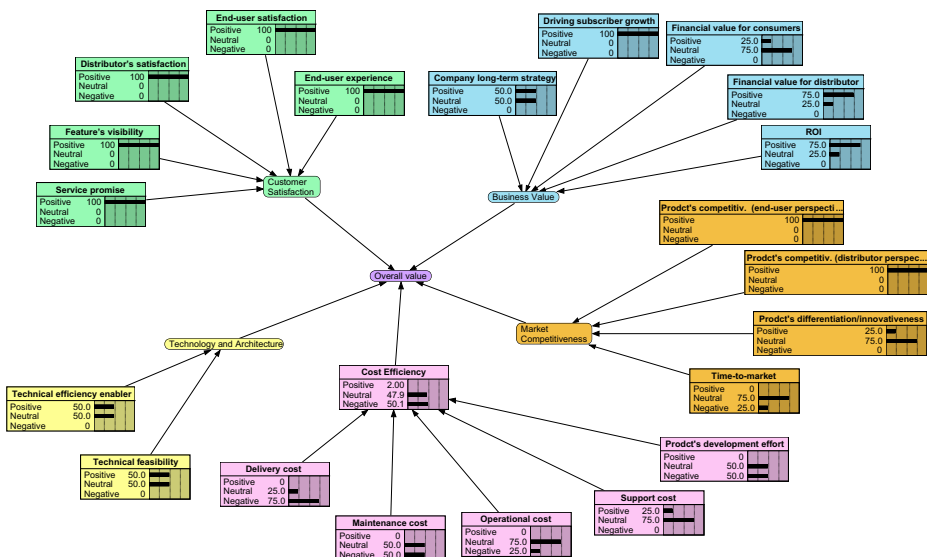

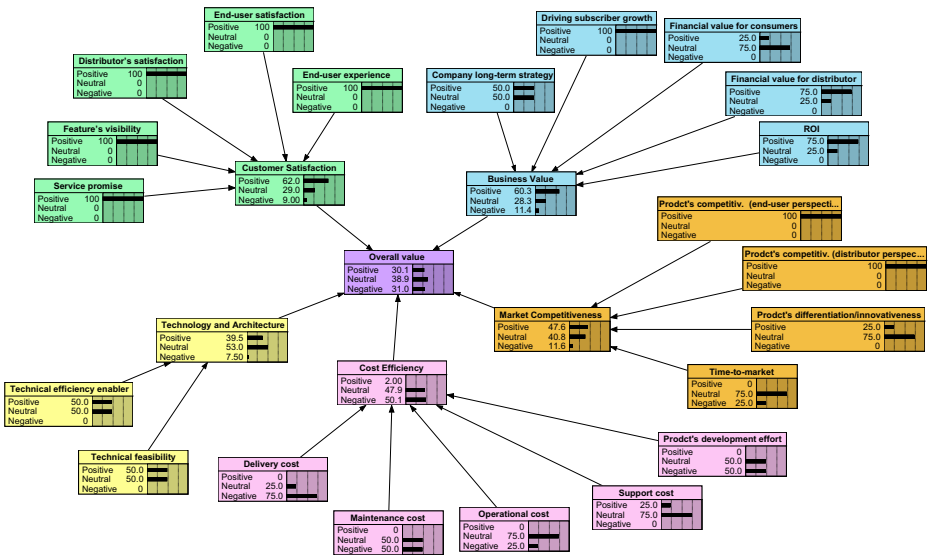Fig. 16 Exporting the generated CPT to the child node cost efficiency

Fig. 17 The example value estimation BN model for company A with all the CPTs quantified

Read 2010) showed that the task was seeing as quite straightforward by the DEs. Further, as will be explained in the next section, the validation of the BN model can also include the direct manual change of probability values as part of the model's calibration step.

6.4 Model validation

As briefly introduced in Sect. 3, the model validation step evaluates the BN model that results from the two previous steps and determines whether it is necessary to re-visit any of those
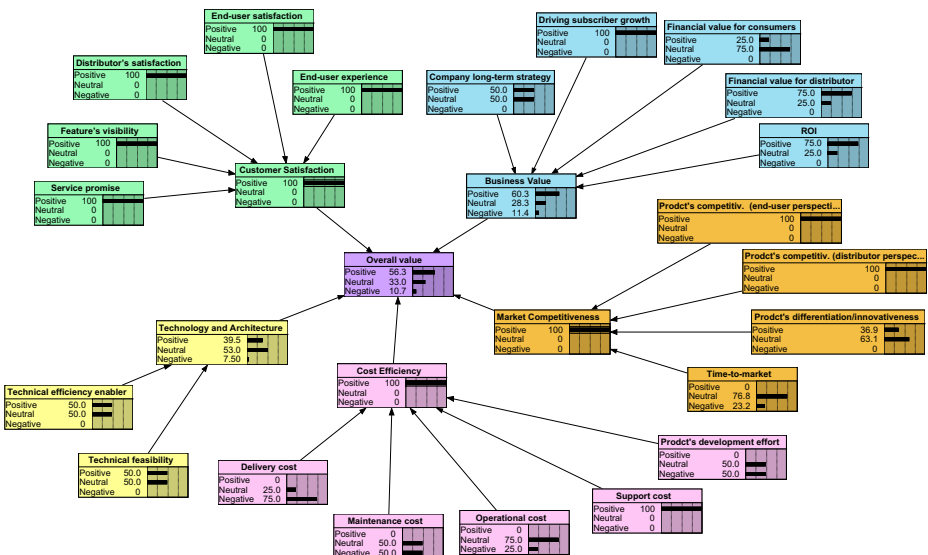

Fig. 18 "What-if" scenario used for model walkthrough where evidence is entered in child nodes

steps. Two different validation methods are generally used—model walkthrough and outcome adequacy. Both methods are to be employed within the context of this research, as follows:

Model walkthrough This is a validity measurement type, called face validity, employed in many areas already cited in Sec. 3, and also used by the first author when building expert-based effort BN prediction models (Mendes 2012; Mendes 2014). Its goal is to obtain an expert-based subjective assessment on how good they "feel" a model is, at face value, with regard to providing/measuring what it is supposed to provide/measure (Drost 2011). Within the context of company A, let us suppose that stakeholder PM, for a given feature N, believes that its implementation into product B would have a positive effect on customer satisfaction, market competitiveness and cost efficiency. This is the scenario shown in Fig. 18, where such evidence has been entered into those nodes. The BN model shows that the overall value for such choice (to include feature N in product B) would be positive with a probability of 56.3%, neutral with a probability of 33% and negative with a probability of 10.7%. If the BN's output for overall value seems plausible to stakeholder PM, this would suggest that the BN model's output at face value matches what stakeholder PM feels should be the case. If the same occurs in relation to all the other scenarios used, then one can say that the model has face validity. However, note that a model that has face validity does not mean the same as a model that has been 'shown/demonstrated' to work. This is why we have an additional method also being used, outcome adequacy, and explained next.

Outcome adequacy uses real data from past decision-making meetings, rather than hypothetical scenarios, to compare what the BN model is suggesting as the state (positive, neutral, negative) for node overall value with the highest probability to the state that is obtained for a given scenario (e.g. row in Table 2) based on data from past decisions.

With regard to company A, and also any other companies using the VALUE framework, outcome adequacy will be measured using the following steps:

1. Gathering of data from value-based decision-making meetings that were not used to build the value estimation BN model. The data uses a similar matrix format shown in Table 2.

7 For each meeting M_1 to $M_{P'}$

a. For each feature (F_1 to $F_{M'}$) discussed in that meeting

   i.   For each key stakeholder S_1 to $S_{N'}$ who participated in the meeting

      1.   Enter in the value estimation BN model the states for all value factors V_1 to V_L

a. Check whether the *highest probability* for the node overall value points to a state that matches the same state for k', for that meeting, stakeholder and feature.

i. If it does not match, then re-calibrate.

Step 2 aims to check every individual assessment made by the stakeholders against the probabilities suggested by the value BN model for node overall value. If any result does not

match, then the model is immediately re-calibrated. This is a similar approach to the one used by Mendes (2012, 2014). Re-calibration means the direct change of probability quantifications in CPTs, in order to provide a matching outcome.

7.1 Using the value estimation BN model

Once the model has been created using various intermediate tools, from structure learning to CPT generation tools, a BN model, previously saved as a file, will be imported by the Value tool, for use by company A's key stakeholders during their decision-making meetings. Note that our goal is for the value estimation model to be an additional resource to the individual assessment and dashboard aggregated data visualisation. Perhaps company A's stakeholders may initially want to do their individual assessment and to use the dashboard to see the aggregated data, and only later on use the BN model to enter evidence representing the aggregated views from the stakeholders in order to obtain the overall value for the feature being previously individually assessed. In any case, the Value tool will enable the use of different "what-if" scenarios, which can be employed jointly during a meeting attended by all the stakeholders, or also individually prior to a joint decision-making meeting. Figures 19 and 20 show possible what-if scenarios using the Value tool.

8 Implications to research and practice

Implications of this work to research This is the first time that a framework that incorporates and intertwines a conceptual framework of knowledge creation (Nonaka and Toyama 2003), Web-based tool support, and a process for the knowledge engineering of Bayesian networks (Mendes 2012), is put forward for use within the context of value-based software engineering, as means to help improve value-based decision-making, and also to estimate the value of decisions relating to software/software-intensive products. This
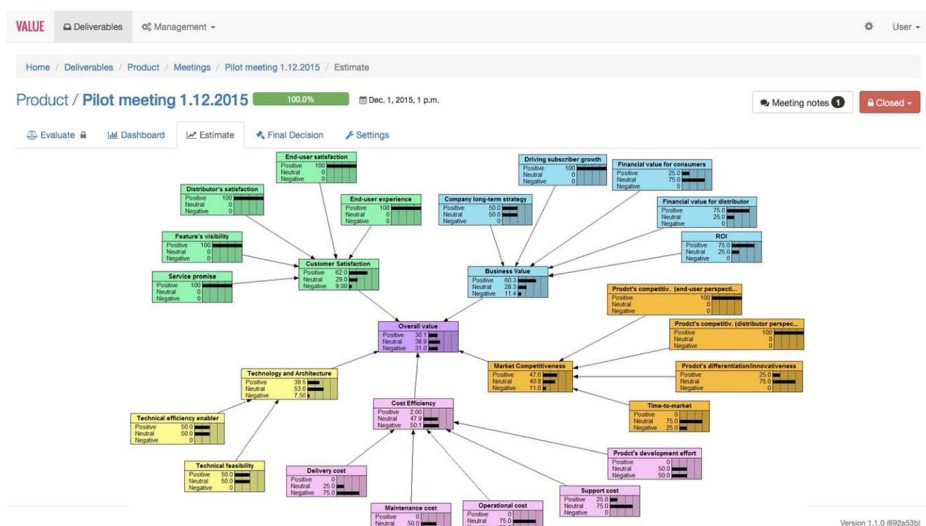
Fig. 19 Company A's example value estimation model imported into the Value tool
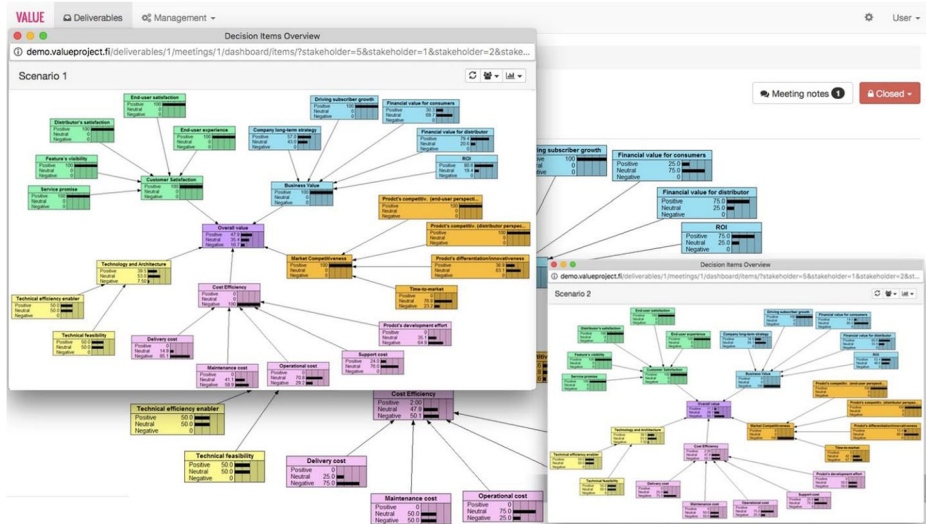
Fig. 20 Different what-if scenarios

framework, VALUE framework, includes detailed steps as to how to identify value factors, how to employ them for value-based decision making and how to use the data gathered from the decision-making meetings to further build and validate a BN model for estimating the value of decisions (Table 4).

We also put forward that the framework detailed herein can be applied to other contexts within value-based software engineering, and even outside VBSE (e.g. organisational governance). This means that additional extensions to the Value tool, and also to the way estimation models are to be built, can be proposed and validated, so providing an opportunity to widen the range of applicability of the initial framework to areas other than product management.

Implications of this work to practice The work detailed herein provides a step-by-step approach to helping companies that wish to improve their current value-based decision-making

Table 4 Data groups to be used to measure the BN model's predictive accuracy

M_p	S_n	F_m	V_1	V_2	V_L	OV
M_1	S_1	F_1	k: positive	k: positive	k: positive	k': positive
M_1						
M_1	S_1	F_{20}	k: neutral	k: neutral	k: neutral	k': neutral
M_1						
M_1	S_N	F_1	k: positive	k: positive	k: negative	k': positive
M_1						
M_1	S_N	F_{20}	k: positive	k: negative	k: negative	k': negative
M_2	S_1	F_{21}	k: positive	k: positive	k: positive	k': positive
M_2						
M_2	S_1	F_{35}	k: neutral	k: neutral	k: neutral	k': neutral
M_2						
M_2	S_N	F_{21}	k: positive	k: positive	k: negative	k': positive
M_2						
M_2	S_N	F_{35}	k: positive	k: negative	k: negative	k': negative
$M_{P'}$	$S_{N'}$	$F_{M'}$	k: positive	k: negative	k: negative	k': negative

in the context of VBSE or to make a paradigm shift from cost- to value-based decision-making. Concretely, the paper describes the following:

1. A way to elicit and aggregate key stakeholders' tacit knowledge, which can later be incorporated into a Web-based tool—Value tool. Such elicitation can be done directly via a focus-group meeting with the key stakeholders, or using a combination of interviews and analysis + focus group meeting to validate the aggregated results from the interviews' analyses.

2. A pilot study carried out with a large software company, describing how the Value tool was used in practice for decision-making. This tool provides the following functionality:

   - Representation of value considerations by different stakeholders
   - Explicit representation of value
   - Support for decision-making meetings
   - Individual assessment of stakeholders' decisions that can be done synchronously during a meeting or asynchronously before the decision-making meeting actually starts
   - A rich dashboard with consolidated data from all stakeholders' assessments that can be used to compare, discuss and support group decisions
   - Historical view of all decisions items selected and not selected for a set range of meetings
   - Means to record stakeholders' decisions and document past decisions
   - Promote transparency on decisions
   - Promote organisational learning
   - In the future, to also enable the estimation of the overall value relating to each decision item being discussed in a meeting

   3. A detailed testimonial from one of company A's key stakeholders clearly stating a number of benefits from using the VALUE framework (value factors' elicitation + Value tool's use)

4. It also provides an example of a value estimation model using a technique, Bayesian network, which enables the representation of knowledge, how such knowledge is inter-related and also the quantification of the uncertainty that is inherent to such knowledge. This technique has been used in a wide range of fields outside (e.g. Darwiche 2010) and inside (e.g. Misirli and Bener 2014) software engineering, which also include software effort estimation, where the first author had first-hand experience building several prediction models in collaboration with companies in New Zealand and Brazil. Further, we also detail another technique, weight sum algorithm, and its extension, which enable the building of CPTs semi-automatically. The WSA and its extension (proposed by one of the co-authors in this paper—Baker) have also been previously employed in probability elicitation with companies in New Zealand (Baker 2009; Read 2010).

How easy is it for ICT companies to employ the VALUE framework? The elicitation of value factors may be completed in a single meeting attended by all key stakeholders who will use the Value tool for decision-making. It only needs someone who is a good

communicator to act as a facilitator. We have recently carried out one such meeting at one of our industry partners, with the participation of six key stakeholders. It lasted for 3 h and led to 27 value factors, which were later on reduced to 19 factors over e-mail exchange with the stakeholders.

The Value tool is straightforward to deploy (~2 h max) and the individual assessment and dashboard features are easy to use, as per usability evaluations carried out (Freitas et al. 2016), and also based on an industry workshop, demonstrations and pilot studies at our industry partners and demonstrations at other companies.

As for the building of a value estimation model using a Bayesian network, this would require a company to have access to experts in BNs, to help with structure learning, using the WSA tool and a BN tool, and model validation, prior to importing the BN model into the Value tool and validating the model with stakeholders.

We estimate that between 16 and 24 h of work should be sufficient to deploy a value BN model in a company. This estimation also assumes that one would be able to have meetings with the key stakeholders during the building and validation processes.

9 Threats to validity

This section discusses threats to the validity of our results using as basis three types of validity threats that are commonly discussed in empirical studies in software engineering (Perry et al. 2000):

Construct validity: represents to what extent the independent and dependent variables precisely measure the concepts they claim to measure.

External validity: represents the extent to which we can generalise the findings to a wider population.

Internal validity: relates to whether changes that occur in the dependent variable can without a doubt be attributed to changes that have occurred in the independent variables rather than be attributed to changes in other unknown factors, e.g. confounding factors.

9.1 Construct validity

With regard to the extent to which one can rely on the value factors that were elicited as truthfully representing how value is used by company A when making decisions relating to product B, our comments are as follows:

- Interviews were carried out with stakeholders who are already thinking value-based. We had the participation of only three stakeholders; however, they are the very key people responsible for all the main decisions relating to company A's product B. In addition, they have spent close to 20 h in internal discussions about the value factors, which were used as input to reducing the original number of factors from around 60 to 21.
- The application of GT-related principles by an experienced research in the use of such principles in several previous research contexts. Further, the first and second authors also attended a 2-day workshop on using GT and GT-related proceedings by a leading expert in this field.
- We held several focus group meetings with the key stakeholders who make decisions relating to product B, who jointly agreed upon the best subset of value factors to use in

their decision-making meetings. Use of these factors in a pilot meeting attended by two of the three key stakeholders.

With regard to the measure chosen to measure value, it was not possible to use a numerical scale; therefore, the choice was between a nominal or ordinal scale. Company A chose to measure value as 'value impact', measured as a three-point ordinal scale, and both the measure and the scale points were decided upon by the key stakeholders.

A large part of the process that was put forward to building company A's value estimation BN model has been previously employed by the first author to successfully build seven effort estimation BN models (five in New Zealand and two in Brazil). In addition, the WSA algorithm has also been assessed by the first and fourth authors in this paper using real expert-based BN models and by an additional researcher in collaboration with a medium-size company in New Zealand (Read 2010) and showed to be a very quick and straightforward way to gather the most important probabilities for child node CPTs. Further, we have also detailed how the probabilities obtained via the WSA algorithm will be validated, and such validation is to be based not only on direct input from domain experts but also using data from past decision-making meetings. We believe that the challenge here does not relate to the proposed solution to building a value estimation model, rather, it relates to the availability of key stakeholders to attend meetings aimed to decide upon parent nodes, relationships, WSA's inputs and so on. Our key stakeholders are responsible for strategic decisions in the organisation and as such have very busy schedules.

9.2 External validity

From its onset, the set of value factors and value estimation BN model were meant to be company-specific solutions. Therefore, the results are bound by the context of the company for which they were done. However, at a later stage, when value factors and value estimation BN models from different companies are elicited, there may be value in aggregating some of them whenever there is a similar industrial context of use, so to understand further the value-based decision making phenomenon. This is one of the topics of our future work.

9.3 Internal validity

The Value tool was co-created from the start with the participating companies, using a design science paradigm (Hevner et al. 2004). This was done purposefully so to provide a tool that represented very closely the existing decision-making processes already used by the participating companies. Thus, we believe that the way in which the Value tool enables the individual as well as the group assessment of decision items is not hindering the quality of the decisions made. As for the value estimation BN model, the example detailed herein was partially created using real data from the pilot study described in Sect. 5, and once we have enough data gathered for decision-making meetings at company A, we will initiate the set of tasks (e.g. focus group meetings to choose the parent nodes, etc.) necessary to build, validate and calibrate the model for use as one of the functionalities provided by the Value tool. This will be one of the topics of our future work.

The choice of estimation technique to be employed for value estimation was discussed with all the participating ICT companies prior to starting the project collaboration and welcomed by all.

10 Conclusions and future work

This paper has proposed and detailed the VALUE framework and has presented some empirical findings from applying it to an ICT company in Finland.

As shown in Sect. 3, the objective of our framework is therefore threefold:

1. To elicit in collaboration with key stakeholders in a decision-making activity, their individual and combined understanding of value
2. To employ a bespoke software solution to support companies improve their value-based decision making
3. To provide additional support to decision-making via a value estimation model that can be used to predict the value of their decisions

The VALUE framework also includes as its basis the conceptual framework of knowledge creation, an adaptation of the knowledge engineering of Bayesian networks process (EKEBN). Both have been used in combination in previous work in the field of effort estimation, in collaboration with numerous companies in New Zealand and Brazil.

As part of our framework, we also include, in combination with the conceptual framework of knowledge creation, and the EKEBN process, a wider range of research methods, both qualitative and quantitative. Some of them were detailed herein using empirical results. These methods are the following:

1. Interviews with key stakeholders in the decision-making process. Interviews were recorded and transcribed in order to be later on analysed using Grounded theory principles. This method can in some cases be replaced by focus group meeting(s) (see next).
2. Focus groups meetings to:

   a. Discuss the set of value factors identified from the interviews, so to agree on the set of factors to use with tool support for value-based decision-making
   b. Elicit and agree upon a set of value factors to use with tool support for value-based decision-making
   c. Identification of the value factors to be parent/child nodes, their relationships, additional factors and which ones affect directly the node overall value
   d. Discuss lessons learnt from employing the value tool for decision making and estimation
3. Case studies in which the Value tool is used by the key stakeholders to make decisions relating to the selection of decision items. We have described herein a pilot study carried out at company A, with the participation of two of the three key stakeholders, where the focus was to use the 21 value factors they previously decided upon in the selection of features. One of company A's key stakeholders also provided a detailed testimonial about the benefits given by the VALUE framework to company A.

One of our goals for future work is to build and validate a value estimation model for company A, as detailed in Sect. 6. Further, we also plan to investigate further possible improvements to the WSA algorithm (semi-automatic generation of CPTs detailed herein), and its comparison to other proposed solutions.

Our future work also includes carrying out focus groups meetings and the building of value estimation models for the other companies participating in our research project. These will be the focus of future research papers.

APPENDIX—Semi-automatic methods for CPT generation and WSA calculations

Semi-automatic methods for CPT generation

Pearl's Noisy-OR Gate (Kincaid and Cheney 2002) is perhaps one of the more established methods, followed by its generalisation (Noisy-MAX Gate) (Kincaid and Cheney 2002). However, these methods often make fundamental assumptions about the BN (e.g. the existence of at most one parent to each existing node; independence between parent nodes, the need for a domain ontology that each node must have an "absent" state). In many domains, however, these assumptions cannot be satisfied. To remedy many of these constraints, other methods were proposed and discussed next. Note that our focus here is on semi-automatic methods that are based on the assumptions that nodes can have more than one parent and that many domains lack a clearly defined ontology. Das (2004) proposed a method known as the weighted sum algorithm (WSA). The WSA is based on two heuristics originally proposed by Tversky and Kahneman (1982). The first states that the more cognitively accessible an event is, the more likely it is perceived to occur (known as the availability heuristic). The second heuristic is to mentally simulate a scenario to assess the ease with which different results are produced given an initial set of parameters and operating constraints (known as the simulation heuristic). What makes this method unique, when compared to others, is that it focuses on asking experts questions that are easy to visualise and simulate because they relate to more realistic probabilities. Once this is done, the method generates the remaining probabilities that are likely harder to estimate. Later on, Fenton and Neil (2005) and Fenton et al. (2007) proposed the ranked nodes method, which focuses on eliciting the parent nodes' CPTs and then generating the CPTs for the child nodes, as opposed to partially eliciting the probability of the child node's CPT and generating the remaining probabilities. Ranked nodes are discrete ordinal variables that can be mapped to a numerical scale that is both continuous and monotonically ordered. The ranked nodes method can be applied only when all the following requirements are satisfied: all nodes involved in the application of this method (i.e. a child node and all of its parent nodes) must satisfy the requirements to be ranked nodes as per definition. Next, Zhong and Brenda (2007) proposed two interpolation methods—linear and piecewise interpolation methods. Interpolation is a method that generates new data points from a given set of discrete data points within a certain range, and it can be used as a heuristic to complete missing probabilities in a CPT. Linear interpolation sets an equal interval between the generated data points given that there are at least two elicited probability values in the table in order to establish a range (Heckerman and Breese 1996; Kincaid and Cheney 2002). Piecewise interpolation predicts missing data points from available data points that share some form of similarity (Kincaid and Cheney 2002). Zhong and Brenda (2007) also showed via empirical studies that linear interpolation is significantly inferior to piecewise interpolation in

terms of estimation accuracy. Chin et al. (2009) recently presented a semi-automatic method (SAM), which was described as a systematic approach for generating probabilities for BNs. To generate a child's CPT, Chin et al. generate separately intermediate CPTs considering only one of the child's parent at a time, and then aggregate all these intermediate CPTs using an algorithm proposed in Pearl (1988). Other methods have also been proposed (Zhou 2015), where some seek to incorporate qualitative expert's knowledge while learning the BN's probabilities from scarce data. They are looking for an optimal estimation of the parameters respecting parameter constraints (e.g. Maximum-likelihood estimation, Monte-Carlo and inference in auxiliary BN). With the exception of some recent work (Baker and Mendes 2010), most of the existing empirical studies on SAMs focus on a singular BN model (from a selected domain) and consider only one factor (e.g., effectiveness measured using a single similarity measure).

Therefore, what makes the WSA algorithm unique, compared to other techniques, is that it focuses on asking DEs questions that are easy to visualise and simulate because they relate to more realistic probabilities. Once this is done, then the algorithm generates the remaining probabilities that are likely harder to estimate. However, an important assumption must be made when applying this algorithm: the DE must be capable of identifying realistic compatible parental configurations of each state for each parent. If the DE fails to do so, the algorithm will generate less trustworthy probabilities.

WSA calculations

To describe the weighted sum algorithm, we will refer to the BN in Fig. 21.

We represent a node's individual states in superscript, so $\{x^1, x^2, \ldots x^n\}$ are all the states of the child node x; for the parent nodes, we will use numbers in subscript to distinguish between different parents. For example $\{p_1^1, p_1^2, \ldots p_1^n\}$ are all the states of the first parent P_1.

To formally define the notion of parental compatibility, let the parent P_i be assigned an arbitrary state p_i^v i.e. $P_i = p_i^v$, and let P_j be another parent, such that P_j is considered compatible with $P_i = p_i^v$ only when P_j is in some state (p_j^w) that is most likely, according to the domain expert's knowledge, to coexist with $(P_i = p_i^v)$. Therefore, we will use the notation $\mathrm{Comp}\left[P_i = p_i^v\right]$ to represent the set of states that are compatible with $P_i = p_i^v$ for all parents.

$$\mathrm{Comp}\left[P_i = p_i^v\right] = \left\{p_j^w, \forall j \neq i \,\middle|\, \max_{w=1\ldots|P_j|} \mathrm{P}\left(p_j^w \,\middle|\, p_i^v\right)\right\} \qquad (1)$$

Fig. 21 Generic naive Bayesian network
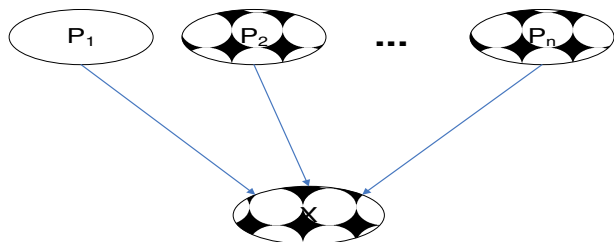
The algorithm takes as input:

1. The conditional probabilities corresponding to the compatible parental configurations for every parental state.
2. A relative weight value (between zero and one) for each parent node, denoting the degree of influence a parent has on a child node.

Those inputs are used as weighted sum corresponding to each state configuration in the child node's CPT, as follows:

$$P\left(X = x^i | p_1^a, p_2^b, p_3^c, \ldots p_n^z\right) =$$

$$w_1\, P\left(X = x^i | \mathrm{Comp}\left[P_1 = p_1^a\right]\right) + w_2\, P\left(X = x^i | \mathrm{Comp}\left[P_2 = p_2^{\,b}\right]\right)$$

$$+ w_3\, P\left(X = x^i | \mathrm{Comp}\left[P_3 = p_3^c\right]\right) + \ldots + w_n\, P\left(X = x^i | \mathrm{Comp}\left[P_n = p_n^z\right]\right)$$

$$for\ i = 1 \ldots \left|x\right| \tag{2}$$

where w is the relative weight value for a parent, e.g. w_1 is the relative weight value for P_1. To illustrate its usage, we refer back to our example in Fig. 2 (Sect. 2), and we intended to calculate an arbitrary cell in the CPT that has not been elicited, for instance, the probability that the overall value is negative given that the customer retention is positive and customer satisfaction is neutral. Assuming that both parents are equally influential (i.e. both have an equal relative weight value of 0.5), then the weighted sum calculation for this cell is as follows:

$$P(\text{Overall value} = \text{negative} \mid \text{customer retention} = \text{positive}, \text{customer satisfaction} = \text{neutra}; al)$$

$$= 0.5 \times P(\text{overall value} = \text{negative} \mid \mathrm{Comp}[\text{customer retention} = \text{positive}]) \quad + 0.5$$

$$\times P(\text{overall value} = \text{negative} \mid \mathrm{Comp}[\text{customer satisfaction} = \text{neutral}])$$

$$\tag{3}$$

References

Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. R. A. (2014). Systematic literature review of software requirements prioritization research. *Information and Software Technology, 56*(6), 568–585.

del Águila, I.M., and del Sagrado, J. (2015) Bayesian networks for enhancement of requirements engineering: a literature review, in Requirements Engineering Journal, pp:1–20.

Alégroth, E. (2013) Random visual GUI testing: proof of concept, Proceedings of SEKE 2013, pp: 1–6.

Arent, J., and Nørbjerg, J. (2000) Software process improvement as organizational knowledge creation: a multiple case analysis, Proceedings of the 33rd HICSS Conference, 1–11.

Aurum, A., Jeffery, R., Wohlin, C., and Handzic, M. (Eds.) (2003) Managing Software Engineering Knowledge, Springer.

Baker, S. (2009) Towards the construction of large Bayesian networks for Web cost estimation, in Department of Computer Science. 2009, University of Auckland: Auckland.

Baker, S., and Mendes, E., (2010). Evaluating the weighted sum algorithm for estimating conditional probabilities in Bayesian networks. SEKE 2010: 319–324.

Barney, S., Aurum, A., and Wohlin, C., (2006) Quest for a silver bullet: creating software product value through requirements selection, software engineering and advanced applications. SEAA'06. 32nd EUROMICRO Conference on. IEEE.

Barney, S., Aurum, A., and Wohlin, C., A (2008) Product management challenge: creating software product value through requirements selection", Journal of Systems Architecture, 54(6), 576–593.

Barney, S., Wohlin, C., Hu, G., and Aurum, A., (2009) Creating software product value in China, software, IEEE, 26(4), 84–90.

Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., and Grünbacher, P., (2005). Value-based software engineering, Springer New York, Inc.

Boehm, B. (2003), Value-based software engineering: reinventing "earned value" monitoring and control. SIGSOFT Software Engineering Notes 2003; 28:3T.J

Castro, O., Espinoza, A., & Martínez-Martínez, A. (2012). Estimating the software product value during the development process. *International Conference on Product Focused Software Process Improvement*, 74–88.

Chermack, T. J. (2003). Mental models in decision making and implications for human development. *Advances in Developing Human Resources, 5*, 408–422.

Chin, Kwai-Sang Tang, Da-Wei et al. (2009). Assessing new product development project risk by Bayesian network with a systematic probability generation methodology. Expert Syst. Appl. 36, 6 (August 2009), 9879–9890.

Darwiche, A., (2010), Bayesian networks, communications of the ACM, vol. 53, pp. 80–90, 2010.

Das, B. (2004), Generating conditional probabilities for Bayesian networks: easing the knowledge acquisition problem. CoRR, 2004. cs.AI/0411034.

Davenport, T. (2013) How P&G presents data to decision-makers, Harvard Business Review, (https://hbr.org/2013/04/how-p-and-g-presents-data).

Davern, M. (2013). Nonresponse rates are a problematic indicator of nonresponse bias in survey research. *Health Services Research, 48*(3), 905–912.

Drost, E. (2011). Validity and reliability in social science research. *Education Research and Perspectives, 38*(1), 105–123.

Dyba, T. (2003), A dynamic model for software engineering knowledge creation, In: Aurum, A., Jeffery, R., Wohlin, C., Handzic, M., (Eds.) (2003) Managing Software Engineering Knowledge, Springer, 95–114.

Easterbrook, S. M., Singer, J., Storey, M, and Damian, D. (2007) Selecting empirical methods for software engineering research. Appears in F. Shull and J. Singer (eds) *"Guide to Advanced Empirical Software Engineering"*, Springer.

Fabri, J.A., L'Erario, A., do Santos Domingues, A.L., Presende Trindade, A.L., de Paula Pessoa, M.S. (2011), Knowledge management and concept maps applied to software process improvement, Proceedings of the 6th Iberian Conference on Information Systems and Technologies, pp. 1–4.

Fenton, N. E., & Neil, M. (1999). A critique of software defect prediction models. *IEEE Transactions on Software Engineering, 25*(5), 675–689.

Fenton, N. and M. Neil (2005) Ranked nodes: a simple and effective way to model qualitative judgements in large-scale Bayesian networks, Research Report, Queen Mary The University of London, Computer Science.

Fenton, N.E., Marsh, W., Neil, M., Cates, P., Forey, S., and Tailor, T., (2004) Making resource decisions for software projects, 26th International Conference on Software Engineering (ICSE 2004), May 2004, Edinburgh, United Kingdom. IEEE Computer Society 2004, ISBN 0–7695–2163-0, pp: 397–406

Fenton, N., Neil, M., & Caballero, J. G. (2007). Using ranked nodes to model qualitative judgments in Bayesian networks. *IEEE Transactions on Knowledge and Data Engineering, 19*(10), 1420–1432.

Fogelström, N. D., (2010) Understanding and supporting requirements engineering decisions in market-driven software product development, PhD Thesis, School of Computing, Blekinge Institute of Technology,.

Foos, T., Schum, G., & Rothenberg, S. (2006). Tacit knowledge transfer and the knowledge disconnect. *Journal of Knowledge Management, 10*(1), 6–18.

Fouad, A., Phalp, K., Mathenge Kanyaru, J., & Jeary, S. (2011). Embedding requirements within model-driven architecture. *Software Qual J, 19*, 411–430.

Freitas, V., Mendes, E., Turhan, B., (2016) Providing tool-support for value-based decision-making: a usability assessment, Proceedings Euromicro 2016 (submitted for assessment)

van der Gaag, L.C., et al. (1999) How to elicit many probabilities, in Uncertainty in Artificial Intelligence. 1999, Morgan Kaufmann. p. 647–654.

Garvin, D.A. (1998) Building a learning organization, In: Harvard Business Rev. on Knowledge Management.

Ghazinoory, S., Daneshmand-Mehr, M., & Reza Arasti, M. (2014). Developing a model for integrating decisions in technology road mapping by fuzzy PROMETHEE. *J. of Intelligent & Fuzzy Systems, 26*, 625–645.

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory*. Mill Valley: Sociology Press.

Hamzah, M., Sobey, A., Koronios, A., (2010) Supporting decision making process with information visualisation: a theoretical framework, Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp: 267–271.

Hayes, J. H., & Offutt, J. (2006). Input validation analysis and testing. *Journal of Empirical Software Engineering, 11*(4), 493–522.

Heckerman, D. (1995), A tutorial on learning with Bayesian networks, Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR- 95–06.

Heckerman, D., & Breese, J. S. (1996). Causal independence for probability assessment and inference using Bayesian networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A, 26*(6), 826–831.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) Design science in information systems research, MIS Q. Manag. Inf. Syst., vol. 28, no. 1, pp. 75–105, 2004.

Holland, S., & Dawson, R. (2011). Classification and selection of tools for quality knowledge management. *Software Quality Journal, 19*(2), 393–409.

Jan, N., and Ibrar, M., (2010) Systematic mapping of value-based software engineering—a systematic review of value-based requirements engineering, Masters thesis Software Engineering, thesis number: MSE-2010:40, 2010, Blekinge Institute of Technology, Sweden.

Jensen, F.V., (1996) An introduction to Bayesian networks. 1996, London: UCL Press.

Jorgensen, M., and Shepperd, M. (2007), A systematic review of software development cost estimation studies, IEEE TSE, Vol 33(1), pp: 33–53.

Khurum, M., Gorschek, T., & Wilson, M. (2012). The software value map—an exhaustive collection of value aspects for the development of software intensive products. *J. of Software: Evolution and Process, 25*(7), 711–741.

Kincaid, D., and W. Cheney (2002) Numerical analysis: mathematics of scientific computing, 2002, American Mathematical Society.

Kopec, J. A., Finès, P., Manuel, D. G. et al., (2010) Validation of population-based disease simulation models: a review of concepts and methods, BMC Public Health, 10, article 710.

Lempert, R., Nakicenovic, N., Sarewitz, D., & Schlesinger, M. (2004). Characterizing climate-change uncertainties for decision-makers. *An editorial essay, Climatic Change, 65*, 1–9.

Lindgren, M., Land, R., Norström, C., & Wall A. (2008a). Key aspects of software release planning in industry. *Proceedings of the 19th Australian Conference on Software Engineering* (pp. 320–329). March 26-28, Perth, Western Australia, 2008. IEEE Computer Society Washington, DC, USA. ISBN: 978-0-7695-3100-7.

Lindgren, M., Land, R., Norström, C., & Wall A. (2008b). Towards a capability model for the software release planning process – based on a multiple industrial case study. *Proceedings of the 9th international conference on Product-Focused Software Process Improvement*. Italy: Monte Porzio Catone.

McElroy, T., & Dowd, K. (2007). Susceptibility to anchoring effects: how openness-to-experience influences responses to anchoring cues. *Judgment and Decision making, 2*(1), 48–53.

Mendes, E., (2012), Using knowledge elicitation to improve Web effort estimation: lessons from six industrial case studies, Proceedings of ACM/IEEE ICSE, pp. 1112–1121.

Mendes, E., (2014), Practitioner's knowledge representation: a pathway to improve software effort estimation, Springer, 2014.

Mendes, E., and Mosley, N. (2008) Bayesian network models for Web effort prediction: a comparative study, IEEE TSE, Vol. 34(6), pp: 723–737.

Mendes, E., Turhan, B., Rodriguez, P., and Freitas, V. (2015) Estimating the value of decisions relating to managing and developing software-intensive Products and Projects, Proceedings of PROMISE'15.

Misirli, A. T., & Bener. (2014). Bayesian networks for evidence-based decision-making in software engineering, IEEE TSE. *Vol., 40*(6), 533–554.

Mohamed, S. I., & Wahba, A. M. (2008). Value estimation for software product management. *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 2196–2200). IEEE Computer Society. doi:10.1109/IEEM.2008.4738261.

Nageswarao M., and Geethanjali, N. (2016) A survey of Bayesian network models for decision making system in software engineering, International Journal of Computer Applications (0975–8887) Volume 134 – No.8, January 2016.

Neil M and Fenton NE, (1996) Predicting software quality using Bayesian belief networks, Proc 21st Annual Software Eng Workshop, NASA Goddard Space Flight Centre, 217–230, Dec.

Neil, M., Krause, P., Fenton, N.E., (2003) Software quality prediction using Bayesian networks, in software engineering with computational intelligence, (Ed Khoshgoftaar TM), Kluwer, ISBN 1–4020–7427-1, Ch. 6.

Nonaka, I., & Toyama, R. (2003). The knowledge-creating theory revisited: knowledge creation as a synthesizing process. *Knowledge Management Research and Practice, 1*, 2–10.

Pearl, J. (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Peñaloza, R. M. C., et al. (2015). A systematic review of research guidelines in decision-analytic modeling. *Value in Health, 18*(4), 512–529.

Perry, D. E., Porter, A. A. & Votta, L. G. (2000) Empirical studies of software engineering: A roadmap. *ICSE - Future of Software Engineering Track*, 345–355.

Radlinski, L. (2010). A survey of Bayesian net models for software development effort prediction. *International Journal of Software Engineering and Computing, 2*(2), 95–109.

Read, J., (2010), Comparing Bayesian network and Web-CoBRA techniques, MSc. Thesis, The University of Auckland, Auckland, New Zealand.

Rodriguez, P., Mendes, E., and Turhan, B. (2016) Identifying key stakeholders' value propositions for feature selection in software-intensive products: An Industrial Case Study, submitted to TSE.

Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation, 7*(1), 12–24.

Schneider, K. (2009) Experience and knowledge management in software engineering, Springer.

Shadbolt, N.R., and Smart, P.R. (2015) Knowledge elicitation, In J. R. Wilson & S. Sharples (Eds.), Evaluation of Human Work (4th Ed.), CRC Press, Boca Raton, Florida, USA.

Shaw, M. (2003) Writing good software engineering research papers, Proceedings ICSE 2003.

Shen-Hsieh, A., and Schindl, M. (2002) Data visualization for strategic decision making, Proceedings of CHI 2002: Case Studies of the CHI2002|AIGA Experience Design FORUM, pp: 1–11.

Steiger, D. M. (2010). Decision support as knowledge creation: a business intelligence design theory. *International Journal of Business Intelligence Research (IJBIR), 1*, 29–47.

Steiger, D.M, and Steiger, N. M. (2007). Decision support as knowledge creation: an information system design theory, in System Sciences, HICSS 2007. 40th Annual Hawaii International Conference on, pp. 204a-204a.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Thousand Oaks: Sage.

Tosun, A., Bener, A. B., & Akbarinasaji, S. (2015). A systematic literature review on the applications of Bayesian networks to predict software quality. *Software Quality Journal*, 1–33.

Tversky, A. and Kahneman, D. (1982). Availability: a heuristic for judging frequency and probability. Judgment under Uncertainty: Hureistics and Biases, 1982.

Urquhart, C. (2012) Grounded theory for qualitative research: a practical guide, Sage.

Weinstein, M.C., B. O'Brien, J. Hornberger, et al. Principles of good practice of decision analytic modeling in health care evaluation: Report of the ISPOR Task Force on Good Research Practices-Modeling Studies (2003). Value Health 2003; 6:9–17

Wieringa, R.J., and Heerkens, J.M.G., (2006) The methodological soundness of requirements engineering papers: a conceptual framework and two case studies, Procs of the Requirements Engineering Conference, pp: 295–307.

Wnuk, K., and Mendes, E. (2015), The project management perspective on software value: a literature review, Proceedings of KKIO 2015.

Wohlin, C., and Aurum, A., (2005) Criteria for selecting software requirements to create product value: an industrial empirical study. Value-based Software Engineering. Springer Berlin Heidelberg, 179–200.

Woodberry, O., Nicholson, A., Korb, K., and Pollino, C. (2004). Parameterising Bayesian networks. Proceedings of the Australian Conference on Artificial Intelligence (pp. 1101–1107).

Zhong, T., & Brenda, M. (2007). Developing complete conditional probability tables from fractional data for Bayesian belief networks. *Journal of Computing in Civil Engineering, 21*(4), 265–276.

Zhou, Yun. (2015) New techniques for learning parameters in Bayesian networks., PhD thesis, Queen Mary University of London, (http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.667460).

Further reading

Chin, Kwai-Sang Da-Wei et al. (2009). Assessing new product development project risk by Bayesian network with a systematic probability generation methodology. Expert Syst Appl. 36, 6 August (2009), 9879-9890.

Das, B. (2004), Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem. CoRR, 2004. cs.AI/0411034.

Heckerman, D. and J.S. Breese (1996) Causal independence for probability assessment and inference using Bayesian networks. IEEE Transactions on Systems, Man and Cybernetics, Part A, 26(6): p. 826-831.

Heckerman, D. (1995), A tutorial on learning with Bayesian networks, Microsoft Research, Redmond, Washington, Tech Rep MSR-TR- 95–06.

Tversky, A. and D. Kahneman (1982a), Judgment under uncertainty: Heuristics and biases. Judgment under Uncertainty: Hureistics and Biases, 1982: p. 3-20.
Tversky, A. and D. Kahneman (1982b), Availability: A heuristic for judging frequency and probability. Judgment under Uncertainty: Hureistics and Biases, 1982.
Zhong, T. and M. Brenda (2007). Developing Complete Conditional Probability Tables from Fractional Data for Bayesian Belief Networks. Journal of Computing in Civil Engineering, 21(4): p. 265-276.
Zhou, Yun. (2015). New Techniques for Learning Parameters in Bayesian Networks., PhD thesis, Queen Mary University of London, (http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.667460)

Emilia Mendes is Full Professor in Computer Science at the Blekinge Institute of Technology (Sweden), and also a Tekes-funded Finnish Distinguished Professor at the University of Oulu (Finland). Her areas of research interest are mainly within the context of empirical software engineering, value-based software engineering, 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 several journals, which include TSE and the SQJ.

Pilar Rodríguez (PhD) is a postdoctoral researcher at the Department of Information Processing Science, University of Oulu, Finland. She earned her PhD in Computer Science from University of Oulu and her M.Sc. in Information Technologies from Technical University of Madrid. Before taking her current position, she was a research assistant at Technical University of Madrid. She has worked in several research projects related to software engineering including EU-ITEA2 FLEXI, Cloud Software Program, N4S and VALUE (Finland). Her research interests include empirical software engineering, value creation and measurement in the software domain, agile, lean and cognitive biases in software engineering.
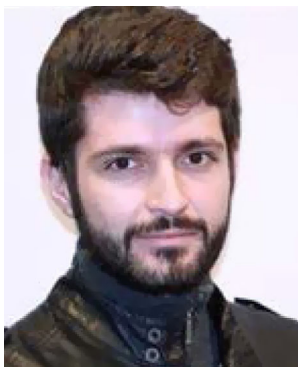
Vitor Freitas is a PhD Student in M3S research team in the Department of Information Processing Science at the University of Oulu, Finland. He received his B.Sc. degree in Information Systems and his M.Sc. degree in Computer Science. Prior to starting his doctoral studies Vitor worked 6 years in software industry as systems analyst and Web developer.
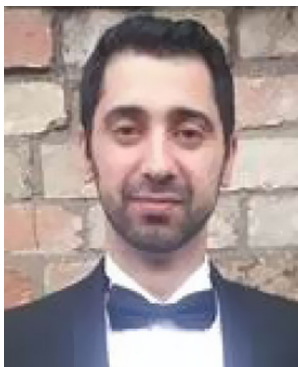
Simon Baker is a Doctoral Commonwealth Scholar and a Cambridge Trust Scholar at Cambridge University's Computer Laboratory. He is also affiliated with the Cambridge Language Technology Laboratory (LTL). His research interests focus on applications of machine learning in Natural Language Processing, Text Mining, and Software Engineering.

Mohamed Amine Atoui received the Ph.D. degree in engineering science from Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS), University of Angers, France, in 2015. His research interests concern Bayesian networks, machine learning and decision making. He is currently a Post-doctoral research fellow in the Department of Information Processing Science at the University of Oulu, Finland.