Bus Inf Syst Eng 62(4):347–360 (2020)
https://doi.org/10.1007/s12599-019-00612-4

RESEARCH PAPER

Exploring the Relations Between Net Benefits of IT Projects and CIOs' Perception of Quality of Software Development Disciplines

Damjan Vavpotič · Marko Robnik-Šikonja · Tomaž Hovelja

Received: 13 March 2018 / Accepted: 13 May 2019 / Published online: 9 August 2019

Abstract Software development enterprises are under consistent pressure to improve their management techniques and development processes. These are comprised of several software development methodology (SDM) disciplines such as requirements acquisition, design, coding, testing, etc. that must be continuously improved and individually tailored to suit specific software development projects. The paper proposes a methodology that enables the identification of SDM discipline quality categories and the evaluation of SDM disciplines' net benefits. It advances the evaluation of software process quality from single quality category evaluation to multiple quality categories evaluation as proposed by the Kano model. An exploratory study was conducted to test the proposed methodology. The exploratory study results show that different types of Kano quality are present in individual SDM disciplines and that applications of individual SDM disciplines vary considerably in their relation to net benefits of IT projects. Consequently, software process quality evaluation models should start evaluating multiple categories of quality instead of just one and should not assume that the application of every individual SDM discipline has the same effect on the enterprise's net benefits.

Accepted after two revisions by Matthias Jarke.

D. Vavpotič (✉) · M. Robnik-Šikonja · T. Hovelja
Faculty of Computer and Information Science, University of Ljubljana, Večna Pot 113, 1000 Ljubljana, Slovenia
e-mail: damjan.vavpotic@fri.uni-lj.si

M. Robnik-Šikonja
e-mail: marko.robnik@fri.uni-lj.si

T. Hovelja
e-mail: tomaz.hovelja@fri.uni-lj.si

Keywords IT project management · Software development process · Kano model · Net benefits · Attribute evaluation · OrdEval algorithm

1 Introduction

Software development enterprises are under increasing pressure to compete in the global market. For this reason, their software development processes need to facilitate the development of complex software products for demanding customers. Under these conditions, the software development process has to produce and support software products that satisfy many consumers, while at the same time achieve economic and design sustainability (Zdravkovic et al. 2015). Unfortunately, according to the CHAOS report (StandishGroup 2015), only 29% of software development projects meet the above-stated demands in that they are completed on-time and on-budget, with all functions as initially specified. Another 52% of the projects are completed and operational but over-budget, exceed the time estimate, and offer fewer functions than originally specified, which can severely affect the quality of the developed software. This situation makes it essential for software development enterprises to continuously improve their software development processes (Bass 2016).

Software development processes comprise several disciplines that can be individually adapted and tailored to suit specific software development projects (Vavpotič and Bajec 2009; Vavpotič and Hovelja 2012), such as requirements acquisition, design, coding and integration, testing, deployment, IT project management, etc. Improving the quality of these disciplines importantly affects the success of software development enterprises (Hovelja et al. 2015).

Several process maturity reference frameworks address the question of software development quality and assume that increased maturity of software development methodology (SDM) results in its increased quality (Loon 2007; Kneuper 2009; Laporte et al. 2016). Unfortunately, these frameworks do not evaluate the actual impact of software development quality on enterprise net benefits, i.e., the extent to which SDM disciplines contribute to the success of the enterprise (Urbach and Müller 2011; Vavpotič and Hovelja 2012; Hovelja et al. 2015). However, even frameworks that attempt to establish such a link (Järvinen et al. 2000; Basili et al. 2007) consider quality as a one-dimensional concept, meaning that satisfaction (i.e., the measure of how SDM disciplines meet or surpass expectations of their users) is considered to be proportional to the quality level: the higher the quality, the higher the enterprise's (as a user of the SDM) satisfaction, and the lower the quality, the lower the enterprise's satisfaction (Chen 2012). Such one-dimensional quality approaches have been proven to limit the understanding of the product design quality and in consequence the ability of enterprises to improve customer satisfaction (Chen and Chuang 2008).

The purpose of this paper is to develop and test a methodology that can identify different quality categories of SDM in order to increase the satisfaction of software development enterprises with the application of their SDM. Specifically, our objective is to identify SDM quality categories (linear and non-linear relations between the quality of an SDM discipline and satisfaction). Such information can help software development enterprises to select SDM improvements (techniques, tools, etc.) that fit the identified quality category/s of each SDM discipline, thus improving their software development process. In order to achieve this objective, we need to answer the following research questions:

RQ1 Can the proposed methodology rank the impacts of different SDM disciplines on the enterprise's net benefits?

RQ2 Can the proposed methodology identify different quality categories of SDM disciplines?

To identify the quality categories used in the proposed methodology, we lean on the quality categories of Kano's et al. (1984) model, one of the most popular multi-category models for quality evaluation (Witell et al. 2013). Kano's model was primarily developed to establish a connection between different quality categories of a product and customer satisfaction. Our proposed methodology attempts to establish a connection between different quality categories of SDM disciplines and SDM user satisfaction. The proposed methodology relies on attribute evaluation algorithms from machine learning with which we assess the contributions of individual attributes (i.e., SDM disciplines) to the overall satisfaction and study their characteristics to identify their quality categories. We used two algorithms, ReliefF (Robnik-Šikonja and Kononenko 2003) and OrdEval (Robnik-Šikonja and Vanhoof 2007). Both algorithms are context sensitive and can detect nonlinear effects. While ReliefF provides results on the level of attributes, OrdEval is adapted to specifics of ordered data, typical for surveys, and allows analysis of individual attribute values, e.g., individual quality or maturity levels of SDM disciplines. OrdEval has so far been used primarily in marketing, and its use in software quality analysis is novel. This algorithm enabled us to analyze different categories of SDM quality as defined by Kano et al. (1984).

The remainder of the paper is organized as follows. In Sect. 2, we present the background and related work. In the first subsection, we present existing conceptual approaches to software process quality evaluation. In the second subsection, we present the Kano model for multi-category quality evaluation. In the third subsection, we present the ReliefF and OrdEval algorithms that are particularly suitable for measuring the influence of SDM quality on net benefits and for identifying multiple categories of SDM quality. In Sect. 3, we present the proposed methodology for improving the quality of individual SDM discipline applications. We introduce our exploratory study in Sect. 4. In Sect. 5, we discuss the implications of the results, and we conclude the paper in Sect. 6.

2 Background and Related Work

Our work builds on three key research areas which we integrate into a novel methodology that enables us to identify different quality categories of SDM disciplines and evaluate their net benefits. To achieve this, we first present the field of software process quality evaluation models to show the need for models that can identify multiple categories of quality. Second, we present one of the most established models for the evaluation of multiple categories of quality, i.e., the Kano model. Finally, we present two advanced feature evaluation measures that the proposed methodology uses to measure the influence of SDM quality on net benefits and to identify multiple categories of SDM quality.

2.1 Existing Software Process Quality Evaluation Models

Several process maturity reference frameworks address the question of software process quality. The focus on measuring quality in evaluation models of software processes can be explained by the quality's impact on user (enterprise) satisfaction (Witell et al. 2013) which is an important component of the relative advantage of an innovation (Rogers 2003). Process maturity reference frameworks include frameworks such as ISO/IEC 15504 (Loon 2007), the Capability Maturity Model Integration (CMMI) (Kneuper 2009), and ISO/IEC 29110 for small organizations (Laporte et al. 2016). These frameworks recognize that a software development process can have various degrees of maturity, where increased process maturity results in increased predictability regarding quality levels (Clarke and O'Connor 2012). Furthermore, they follow the process management premise that the quality of a system or product is highly influenced by the quality of the process used to develop and maintain it. Thus, they define maturity levels that embody this premise (Forrester et al. 2011). A maturity level is evaluated for each part of a development process based on the reference framework.

Such assessments provide a benchmark against a set of goals but do not evaluate the actual impact of process changes on enterprises net benefits (Unterkalmsteiner et al. 2014). This concern was acknowledged in the literature linking software process quality measurements with net benefits (Vavpotič and Hovelja 2012; Hovelja et al. 2015) and also addressed by frameworks such as PROFES (Järvinen et al. 2000) and GQM + Strategies which attempt to link business strategies with measurement goals (Basili et al. 2007). However, all the above presented frameworks and research in the field of software development process perceive quality as a single category, with a linear-like relation between quality and user satisfaction (so-called one-dimensional quality category) and do not take into consideration that other fields recognize different quality categories with non-linear relations between quality and user satisfaction (Lin et al. 2010; Witell et al. 2013). For instance, Ishikawa (1990) divided quality into two categories, backward-looking quality and forward-looking quality (Kondo 2001). A similar proposal was made by Kano et al. (1984) who, inspired by Herzberg et al.'s (1959) H–M theory, introduced multiple quality categories including an attractive (provides satisfaction when achieved fully but does not cause dissatisfaction when not fulfilled since it is not normally expected), a must-be (expected requirements that are taken for granted), and a one-dimensional one (results in satisfaction when fulfilled and dissatisfaction when not fulfilled). The introduction of the model with multiple categories enabled Kano (for the detailed explanation see Sect. 2.2) to better define the relations between quality and satisfaction (Kondo 2001; Witell et al. 2013). For instance, it was demonstrated that a TV remote control was an attractive quality category in 1983, a one-dimensional quality category in 1989, and a must-be quality category in 1998 (Kano 2001).

In the field of software development processes and SDM, we can similarly observe different quality categories. The two particularly interesting eras of SDM development as defined by Avison and Fitzgerald (2006a) are the so-called methodology and post-methodology era. In the methodology era, it was the prevalent view that the many problems of IS development can be successfully addressed by the adoption of SDM of some kind (Avison and Fitzgerald 2006b). Such a view indicates that SDM developed in the methodology era were perceived by many companies either as a one-dimensional or an attractive quality category since higher SDM quality typically meant greater satisfaction. However, later in the post-methodology era, it became clear that it was unlikely that SDM would ever achieve the exaggerated claims made by some vendors and consultants (Avison and Fitzgerald 2006a). In consequence, agile SDM emerged and one of their key principles states that excess SDM weight (defined as the conceptual product of size and ceremony of SDM) is costly (Cockburn 2002). Many companies, previously striving to achieve higher levels of SDM quality, transitioned to the agile SDM. In contrast to the traditional SDM, the agile SDM mostly focus on certain parts of the development process while for other parts they rely on discipline, skills, and the understanding of people involved in the development (Cockburn 2002). This indicates that in the post-methodology area, SDM are perceived by many companies as a must-be quality category where only basic implementation of SDM is required.

Similar perceptions can be observed at the level of SDM disciplines. In the early post-methodology era, Miller (2001) defined three types of methodologies: front-loaded, back-loaded and balanced. Front-loaded SDM stress the necessity of well-defined and systematic SDM disciplines in the fields of requirements acquisition, analysis, and design. Back-loaded SDM stress the importance of well-defined coding, integration, testing, and deployment while requirements acquisition, analysis, and design are performed less formally in the process of creating the system itself. Balanced SDM adapt to the needs of a project at hand by applying more or less SDM in different disciplines. This indicates that companies using different SDM types perceive SDM disciplines as different quality categories. On the one hand, it is likely that companies using front-loaded SDM perceive the use of SDM in requirements acquisition, analysis, and design disciplines as a one-dimensional quality category since the higher the quality, the greater the satisfaction. On the other hand, they typically consider new techniques and approaches that substantially contribute to the existing SDM as an attractive quality category. In contrast, the companies using back-loaded SDM take requirements acquisition, analysis, and design disciplines for granted since they are perceived as a must-be quality category. Thus only the basic quality is needed to achieve most of the satisfaction. This demonstrates that a single quality category evaluation is insufficient for an in-depth evaluation and understanding of the software development process.

2.2 Kano Model

Since its introduction in the 1980s, Kano's model (Kano et al. 1984) has become one of the most popular multi-category models for evaluation of quality in multiple research fields (Witell et al. 2013). The Kano model was developed to categorize the product/service features' quality based on their ability to satisfy customers' needs. The five Kano categories are must-be quality, one-dimensional quality, attractive quality, indifferent quality, and reverse quality. The must-be quality category has a log-like function shaped relation between quality and satisfaction (Fig. 1). Attributes from this category cause significant dissatisfaction when expectations are not fulfilled. However, if their value is adequate they do not contribute to the satisfaction of customers. The one-dimensional quality category shows a positive linear-like relation between quality and satisfaction. Attributes from this category cause dissatisfaction when their value is low and cause satisfaction when their value is high. The attractive quality category is characterized by an exponential-like relation between quality and satisfaction. Attributes from this category do not cause dissatisfaction when their value is low but they strongly contribute to the satisfaction of customers when their value is high. The indifferent quality category does not show any significant relationship between attribute value and satisfaction. The reverse quality category has a negative linear-like relation between attribute value and satisfaction. Attributes from this category cause satisfaction when their value is low and dissatisfaction when their value is high.

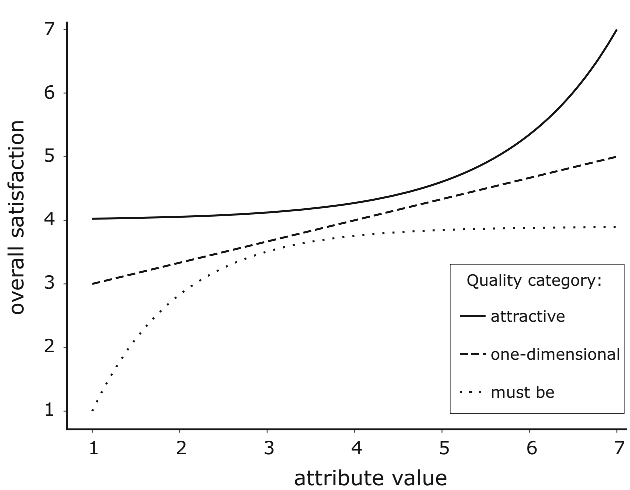

Fig. 1 Graphs of idealized Kano quality categories

Figure 1 shows the impact of attribute values on the overall satisfaction for the must-be category (dotted line), one-dimensional category (dashed line), and attractive category (solid line) according to the Kano model. As scales for the attribute values and the overall satisfaction we choose 7-point Likert scales, so the satisfaction score of four is considered neutral. The curves in Fig. 1 show idealized quality categories. In reality, we might encounter different thresholds for satisfaction and dissatisfaction regarding attractive and must-be categories, as well as different coefficients of linear growth for one-dimensional categories. In fact, different groups of users might have different perceptions of an attribute, e.g., some may consider it as belonging to a one-dimensional and others to a must-be category.

Lofgren and Witell (2008) reviewed the research on the application of Kano's model over the last two decades and found that 21 of the 28 studies used the original Kano questionnaire to classify attributes of quality. Kano's approach requires the compilation of a questionnaire with a list of functional and dysfunctional questions for each attribute to observe the distribution of customer views. For example, customers are first asked how they would feel if a particular attribute were present or fulfilled [possible answers are (a) satisfied, (b) it should be that way, (c) I am indifferent, (d) I can live with it, or (e) dissatisfied] and then how they would feel if that attribute were not present or unfulfilled (with the same possible answers). The response to both questions determines the nature of the attribute according to the Kano's model.

2.3 Selected Approach for Classifying Kano Quality Categories

Existing empirical studies have found the application of Kano's original questionnaire complex and difficult to implement in real-world situations (Lofgren and Witell 2008). Besides the original Kano questionnaire, several other approaches for classifying quality attributes were proposed (Chen 2012), such as the penalty-reward contrast analysis (PRCA), the importance grid analysis (IGA), the direct classification method, and the moderated regression analysis. Practically most useful are regression methods such as PRCA, which introduce dummy variables to model nonlinear relationships between different variables and quality (Lin et al. 2010). While these methods can provide some information about the attributes, they are not theoretically justified and the resulting coefficients of introduced dummy variables are not easy to comprehend, especially in the realistic circumstances with a presence of noise in answers (Mikulic and Prebezac 2011). Another difficulty is the requirements for normalization of variables in this method, which distorts present nonlinearities. A practical advantage of the regression methods is that they directly analyze attribute-level user satisfaction, which is easier to collect in surveys than the list of functional and dysfunctional questions proposed by Kano.

In machine learning and data mining, the quality of variables (also called attributes, features, independent variables, predictors, or input variables) is an important research question tackled in tasks such as variable evaluation, variable subset selection, and variable ranking. Several measures have been proposed, which mostly estimate the quality of variables through their predictive power concerning the response variable (also called dependent variable or class variable, in our case this is the satisfaction). Guyon and Elisseeff (2003) provide an overview of classical attribute selection approaches. Recently, the research in this area has been focused on specialized measures, for example for specifics of bioinformatics (Bolon-Canedo et al. 2014) or big data (Zhao et al. 2018).

Simple feature evaluation measures like the Gini index (Breiman et al. 1984) take only the dependence between one variable and satisfaction into account. More advanced measures take into account conditional dependencies between a response variable and several variables, the best known such measure being ReliefF (Robnik-Šikonja and Kononenko 2003). These quality evaluation measures are concerned with the predictive power of variables and, similarly to regression approaches, can detect important variables. However, they do not take into account the specifics of ordered variables (e.g., ordered attributes/ questions in surveys) and cannot provide useful evaluation for each individual value of a variable.

OrdEval (Robnik-Šikonja and Vanhoof 2007) is a feature evaluation measure building upon ReliefF. It was initially developed for customer satisfaction research in marketing. The algorithm evaluates ordinal variables, i.e., survey questions, based on their relation to the expected outcome. Different from ReliefF and other feature evaluation measures, it analyzes each variable's value separately and takes into account the asymmetric effect an increase or decrease of its value may have on the response. This allows a kind of what-if analysis. As a result, the algorithm returns conditional probabilities of the expected satisfaction upon changes in attribute values. For example, in our study, we obtain probabilities of higher overall satisfaction due to higher satisfaction with an individual SDM discipline. Previous usages of OrdEval encompass data analysis in marketing research and pharmacology. In marketing, researchers analyzed customer satisfaction (Robnik-Šikonja and Vanhoof 2007), the country of origin (Robnik-Šikonja et al. 2009), the impact of adaptive collaboration on demand forecasting accuracy of different product categories throughout the product lifecycle (Nagashima et al. 2015), and the impact of performance history on supplier selection in the French public sector (Mamavi et al. 2015). The OrdEval algorithm allows categorization of product/ service features according to the Kano model. Čufar et al. (2015) showed that using OrdEval one can construct simpler and substantially shorter questionnaires (one question per attribute) and still apply the Kano model. The authors applied the OrdEval algorithm to the management of hospital clinical pharmacy services, while we apply and adapt it to the area of software quality analysis. In the proposed methodology, we use ReliefF to identify important attributes and OrdEval to characterize them according to the Kano model, which allows us to analyze attribute-level customer satisfaction data and obtain insights not available with other methods. Both evaluation measures, ReliefF and OrdEval, are implemented in the R package CORElearn (Robnik-Šikonja and Savicky 2016).

The output of OrdEval are probabilities that an increase/ decrease in the individual attribute's value will have an impact on the response variable. The intuition behind this algorithm is to approximate the mental decision process, taking place in each individual respondent, which forms a relationship between the attribute and the response. Namely, by statistically measuring a causal effect the change of an attribute's value has on the response value, we can perform probabilistic reasoning about the importance of the attribute's values, the type of the attribute, and determine which values are thresholds for a change of behavior. For each respondent, OrdEval selects the most similar respondents and makes an inference based on the differences between them. For example, to evaluate the effect an increase of a certain attribute value would have on the overall satisfaction, the algorithm computes the probability for such an effect from similar respondents with a larger value of that attribute. The overall process is repeated for a large enough number of respondents to obtain statistically valid results.

The methodology returns conditional probabilities called 'reinforcement factors'. These factors approximate the upward and downward reinforcement effect the particular attribute's value has on the satisfaction and are depicted on the right and left-hand side of each graph, respectively (see Figs. 2, 5, 6, 7, 8, 9, 10). For each value of the attribute, we obtain estimates of two conditional probabilities: the probability that the satisfaction value increases given the observed increase in the attribute's value (upward reinforcement), and the probability that the satisfaction value decreases given the observed decrease of the attribute's value (downward reinforcement). To take the context of other attributes into account, the probabilities are computed in a local context, from the most similar instances. The visualization of these factors with box-plots provides information about the role of each attribute, the importance of each value, and the threshold values. To

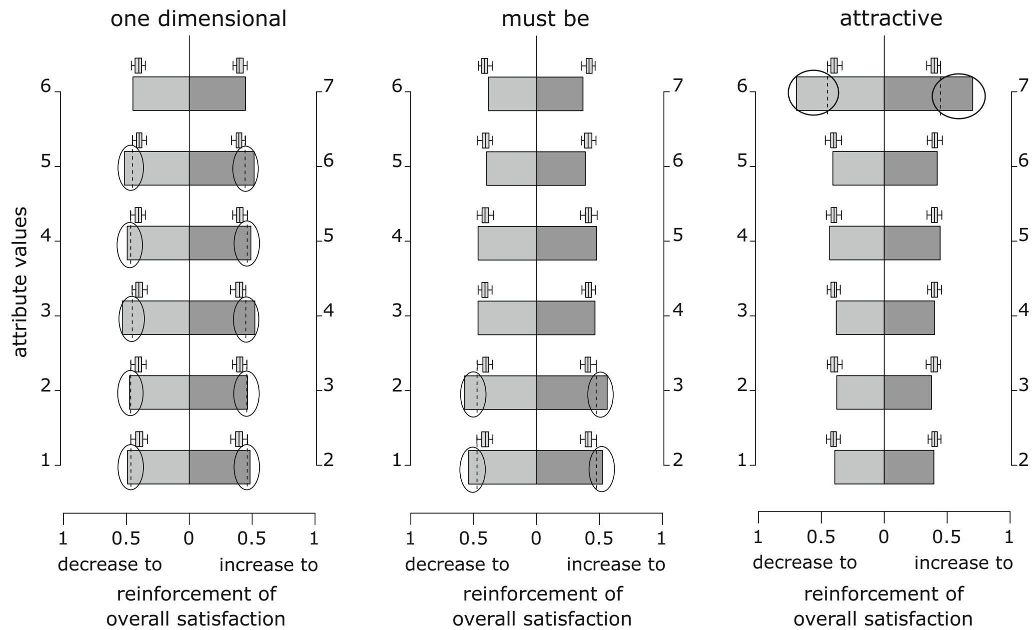

Fig. 2 Visualization provided by the OrdEval algorithm for three idealized attributes constructed to demonstrate the quality categories: one-dimensional (left), must-be (middle), and attractive (right)

understand the idea of the OrdEval algorithm, the attribute should not be treated as a whole. Rather, we shall observe the effect a single attribute's value may have.

For each reinforcement factor, OrdEval computes confidence intervals which are plotted as box-and-whiskers plots above the obtained reinforcement factors. Statistically, reinforcement values outside the confidence intervals are significantly different from random effects at the 0.05 level. Missing values of survey questions are estimated from their class-conditional probabilities.

Figure 2 shows an illustration of the three quality categories according to the Kano model: one-dimensional, must be, and attractive. The graphs are generated with the OrdEval algorithm from a data set with three hypothetical product/service features (i.e., attributes) clearly showing these three quality categories. The values of attributes are generated on a 7-point ordinal scale with a uniform distribution of values, where 1 denotes very low satisfaction and 7 denotes very high satisfaction. The overall satisfaction in this toy example is defined as a sum of contributions from all three attributes with a small amount of noise (to make it more realistic) and normalized to the 1–7 scale. The bars on the graphs' left-hand side show downward reinforcements, i.e., the probability that the overall satisfaction will decrease if the value of the attribute decreases (e.g., in the top left-hand side row of the three graphs, we can observe the impact on the overall satisfaction if attribute values change from 7 to 6). The bars on the right-hand side of the three graphs show upward reinforcements, i.e., the probability that the overall satisfaction will increase if the value of attributes will increase (e.g., in the top right-hand row of the three graphs, we see the impact on satisfaction if the attribute value changes from 6 to 7). For the attribute demonstrating the one-dimensional category, almost all values show considerable reinforcements, both for increase and decrease of values (denoted with ellipses). For the must-be category attribute, we can observe a strong jump in impact when the value changes between 1 and 2 as well as for changes between 2 and 3. For the attractive category attribute, we notice a jump when the value changes from 6 to 7 or inversely. Note that these are idealized attributes and in reality the effects may have different thresholds, they may be mixed within the same attribute (e.g., due to different perception of the same attribute by different subgroups of users), or may not be significant enough (e.g., due to a low impact of attribute or an insufficient number of users expressing certain score). To quantify the significance of the impact, the box-and-whiskers plots above each reinforcement bar show the distribution of reinforcement scores under the condition that the impact of the attribute is random but with the same value distribution. The reinforcement bars stretching beyond the whiskers (95% confidence interval estimated with bootstrapping) are therefore statistically significant in the sense that it is highly probable that their effect is not random. For example, due to added noise, in the attribute from the one-dimensional category we see that the changes from 7 to 6 and 6 to 7 are not statistically significant. For real-world examples, the box-and-whiskers are typically wider than the ones shown for idealized attributes in Fig. 2.

All reinforcement factors that express this statistical significance obtained with permutation test are marked with ellipses on the graphs.

The information the OrdEval algorithm provides has no parallel in standard approaches like multiple regression analysis. First, there is substantial context sensitivity. Typically, the attributes are highly conditionally dependent upon the response and have to be evaluated in the context of other attributes. OrdEval is intrinsically contextual and assumes neither independence nor any fixed distribution of the attributes. Second, there is the ability to handle ordered attributes and ordered response and to use the information the ordering contains. The order of attribute's values contains information which is comparable but not the same as values of numerical attributes, e.g., poor, good, very good and excellent values are ordered when expressing a certain attitude but this ordering is not necessarily linear. Third, OrdEval is aware of the ordered nature of answers and of the positive (negative) correlation between changes of attribute values and the response (e.g., if the value of the attribute increases from poor to good, we have to be able to detect both positive and negative correlation to the change of the overall response value). Fourth, OrdEval has the ability to handle each attribute value separately, e.g., for some attributes, the good and very good values have an identical neutral impact on the response, the poor value may have a strong negative impact, and the excellent value has a highly positive impact. We are able to observe and quantify each attribute's values separately and thereby identify important thresholds. Fifth, the permutation test generating a multitude of random attributes with the same distribution of attribute values as the original attribute provides confidence intervals for what a random effect would be and the reinforcement values beyond that interval are highly unlikely to be produced by chance. Sixth, the visualization of the output allows experts to use it as a powerful exploratory data analysis tool, e.g., to identify the type of attributes in the Kano model. Seventh, the output takes the form of probabilities which are comprehensible and interpretable by a large audience and can also be used operationally. Finally, the algorithm is fast and robust concerning noise and missing values.

3 The Proposed Methodology

Based on the reviewed literature we propose a novel methodology that evaluates the impact of individual SDM disciplines on enterprise net benefits and also considers quality as a multiple-category construct. The methodology consists of four main phases as presented in Fig. 3.

In the first phase, we measure the CIOs' satisfaction with the individual SDM disciplines application and net benefits of IT projects. The satisfaction with SDM application is measured on the level of disciplines that are defined based on the well-established Rational unified process (Kruchten 2000) and include requirements acquisition, system design and architecture, coding and integration, testing, and deployment. One questionnaire item was used for each SDM discipline. The net benefits questionnaire items are defined in accordance with DeLone-Mclean model of IS success (Urbach and Müller 2011). All 7 questionnaire items use 7-point Likert scales ranging from 1 (strongly disagree) to 7 (strongly agree). It is important to note that it is not necessary to use Rational Unified Process disciplines. The disciplines or other process parts from different SDM can be used as the basic unit of software development process evaluation. Similarly, it is not necessary to focus on the CIO perspective as other relevant stakeholder perspectives can be used as a substitute, e.g., project managers, process managers, or product managers.

In the second phase, we use ReliefF (Robnik-Šikonja and Kononenko 2003) to analyze the associations between CIOs' satisfaction with the application of SDM in individual development disciplines and net benefits of IT projects, taking into account possible attribute interdependencies. The ReliefF score near 0 or below is typical for disciplines with irrelevant net benefits, while positive values reveal disciplines with relevant associations with net benefits. The exact values of ReliefF scores are problem-related, therefore we avoid direct interpretation of numerical values and use ReliefF scores to rank the associations according to their importance.

In the third phase, we use the OrdEval algorithm (Robnik-Šikonja and Vanhoof 2007) to analyze the impact of the attribute values, i.e., CIOs' satisfaction scores for individual disciplines which allows us an inference about attribute characteristics according to the Kano model. The visualization of OrdEval results can indicate the thresholds where the attribute's values start having a strong positive or negative impact on the overall CIOs' satisfaction (see examples for typical attributes in Fig. 2).

In the fourth phase, we use the information gathered in the previous phases to prepare improvements that focus on the SDM disciplines with the highest impact on enterprise net benefits and fit the quality categories of individual SDM disciplines. This enables enterprises to focus on the improvement of the most beneficial SDM disciplines and aligns the improvements with the identified quality categories in specific SDM disciplines.

4 Exploratory Study

Our exploratory study aims to prove the capability of the proposed methodology to rank the impact of SDM disciplines on enterprises' net benefits (RQ1) and identify

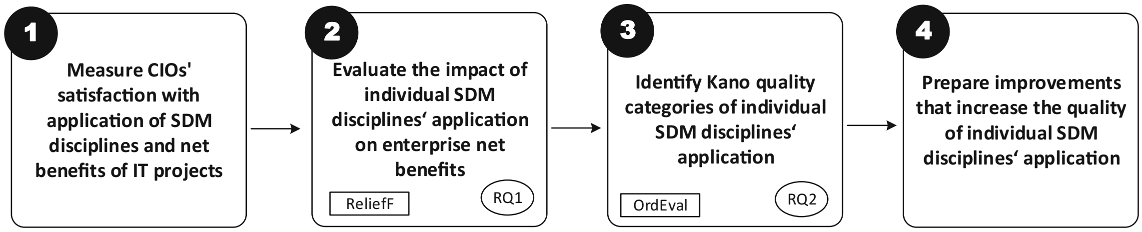

Fig. 3 The proposed methodology for improving the quality of individual SDM discipline application

different quality categories in specific SDM disciplines (RQ2). The exploratory study was selected as an appropriate approach that allows us to address the studied phenomena in a new light (Robson 2002). It is especially suitable to address the "how many" line of inquiry (Yin 2009). In our specific case, we can obtain an answer to the question of how many different quality categories individual SDM disciplines exhibit. Survey or archival methods are better suited to address this line of questions than other methods (Yin 2009).

4.1 Study Description

In line with the majority of other IT deployment studies in the literature (Brynjolfsson et al. 2002; Bresnahan et al. 2002; Mittal and Nault 2009; Hovelja et al. 2013), this study focused on the largest non-financial enterprises in a country. The financial enterprises were not included, as they are exposed to stricter regulatory laws which do not give them the same freedoms concerning organizational processes and structures as the non-financial enterprises have (Hovelja 2008), and because of problems in defining and quantifying their output (Brynjolfsson and Hitt 1996). The starting population included the top 1000 enterprises in Slovenia based on the 2014 added value creation. The included enterprises are not a random sample as the top thousand enterprises in Slovenia were surveyed. However, this group of enterprises presents a relevant study group due to its importance to the national economy. We sent the surveys to the CIOs of the studied enterprises and received 113 appropriately completed responses. The survey was conducted from March until May 2016. Based on personal and phone communication with CIOs involved in the study, we found that the relatively low 11.3% response rate was mainly caused by a lack of time to fill out the questionnaire. The questionnaire asked participants for information about the characteristics and outcome of a recently completed important software project they had been involved in, regardless of its size and success. The key characteristics of our sample are as follows.

Fifty-eight percent of the projects had a budget of less than 100,000 EUR, 29% between 100,000 EUR and 500,000 EUR and 12% over 500,000 EUR, while there was no response from the remaining 1%. On average the reported projects involved 13.3 people of which 4.8 were external contractors. Fifteen percent of the projects lasted less than 3 months, 27% lasted between 3 and 6 months, 29% lasted between 6 months and a year, 21% lasted more than a year while there was no response from the remaining 8%. Twenty-eight percent of the deployed software products were custom solutions, 35% were customized local pre-packaged solutions, while the remaining 37% were customized pre-packaged solutions offered by international vendors. Eleven percent of the projects achieved all anticipated net benefits of the deployed solution, 64% partially achieved anticipated net benefits of the deployed solution, while the remaining 25% failed to achieve most of the anticipated net benefits.

4.2 Study Results

In Fig. 4 we present the association between CIOs' satisfaction with SDM application in individual disciplines and net benefits of IT projects taking into account possible interdependencies (using algorithm ReliefF as a step 2 in the proposed methodology, described in Fig. 3).

The results show that the application of SDM is positively associated with the net benefits of IT projects in all disciplines (with the exception of project management). Based on our survey, the *Testing* discipline has the strongest association with the net benefits of IT projects, while *Coding and integration* discipline is ranked second and *Deployment* close third. The benefits of all three disciplines probably stem from the use of agile SDM that may result in shorter time-to-market, continuous feedback, improved release reliability, increased customer satisfaction, and improved developer productivity (Rodriguez et al. 2017). Interestingly, the CIOs' satisfaction with the application of SDM in the *Project management* discipline is not positively associated with the net benefits of the IT projects.

As step 3 of in the proposed methodology, the OrdEval method was used to classify SDM disciplines into five Kano quality categories. Figures 5, 6, 7, 8, 9 and 10 show visualizations of the OrdEval results for each discipline.

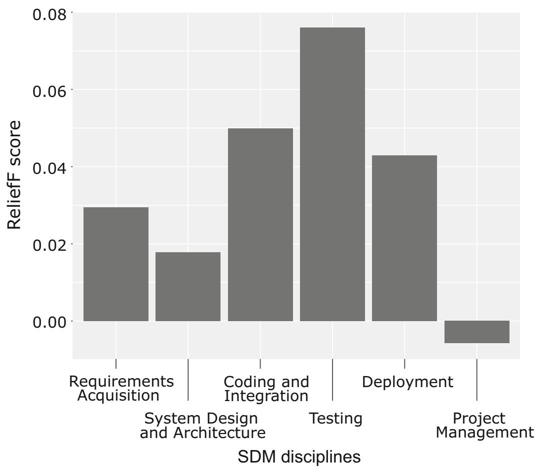

Fig. 4 Association between CIOs' satisfaction with SDM application in disciplines and net benefits of IT projects (ReliefF)

When interpreting the results, we focus primarily on statistically significant outcomes. Summarized results of how disciplines map to different Kano quality categories are provided in Table 1.

Figure 5 shows the statistically significant positive influence of SDM application in the *Requirements acquisition* discipline on CIOs' overall satisfaction when attribute values (satisfaction with *Requirements acquisition*) increase from 3 to 4, 4 to 5, and from 6 to 7. There is also a statistically significant negative influence when the satisfaction with *Requirements acquisition* decreases from 4 to 3 and from 7 to 6. Note that statistically significant impacts

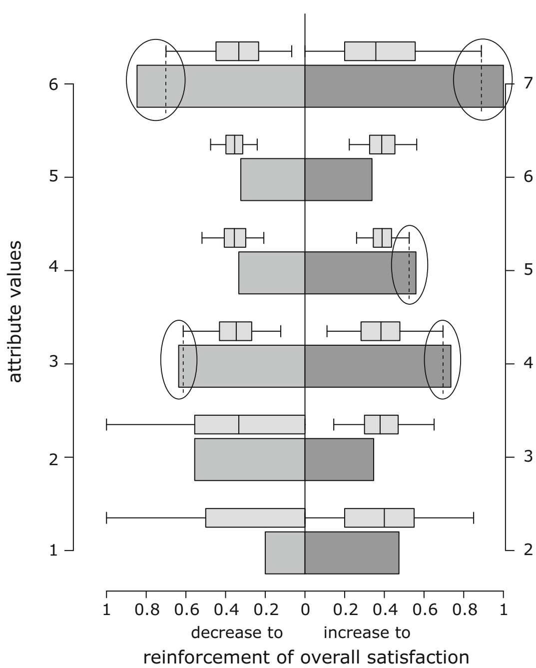

Fig. 5 OrdEval results for *Requirements acquisition*

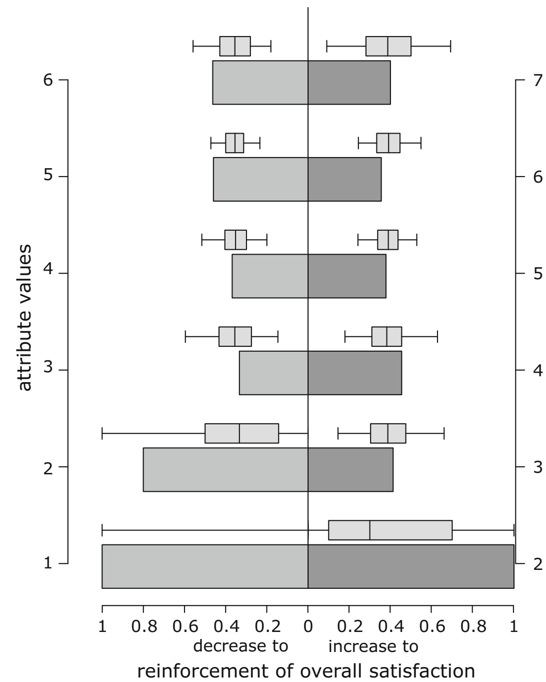

Fig. 6 OrdEval results for *System design and architecture*

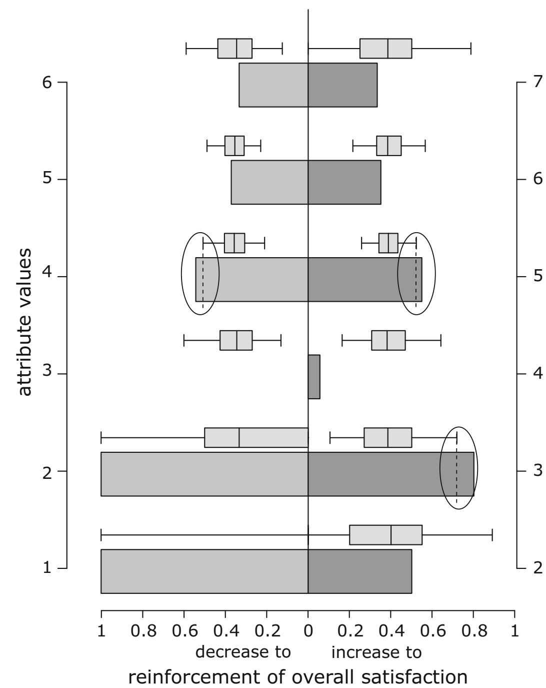

Fig. 7 OrdEval results for *Coding and integration*

are circled and can be recognized in the graph as those whose reinforcement bars stretch beyond the box-and-whiskers plots above them (box-and-whiskers plots show the distribution of random effects obtained with permutation test). The reinforcements of overall satisfaction outside these confidence intervals are significantly different from

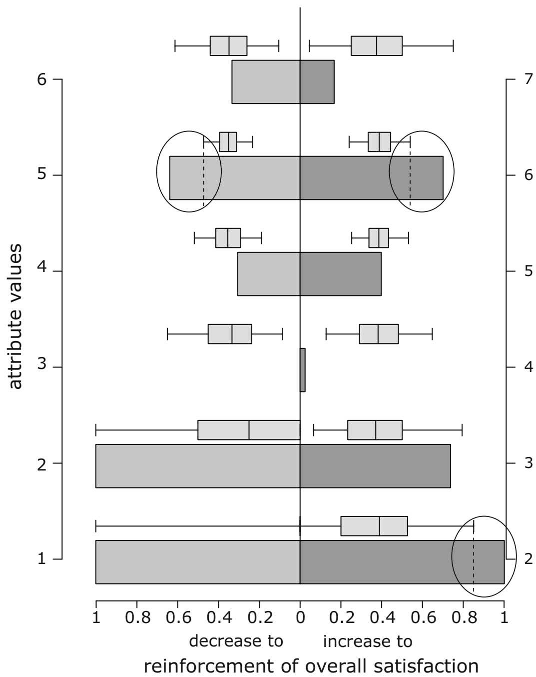

Fig. 8 OrdEval results for *Testing*

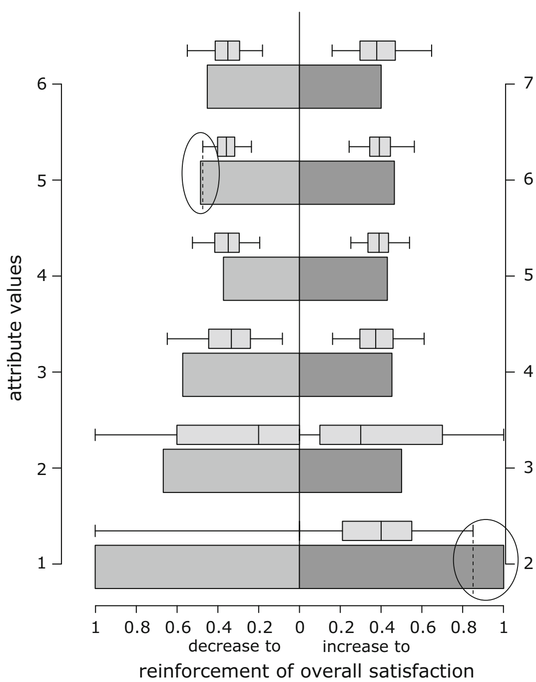

Fig. 9 OrdEval results for *Deployment*

random effects. In Fig. 5, the top right and left bars show the strongest effects. This indicates that the attractive quality category of SDM application in the *Requirements acquisition* discipline has a strong statistically significant influence on CIO satisfaction with the discipline, however, one-dimensional quality category influence is also present.

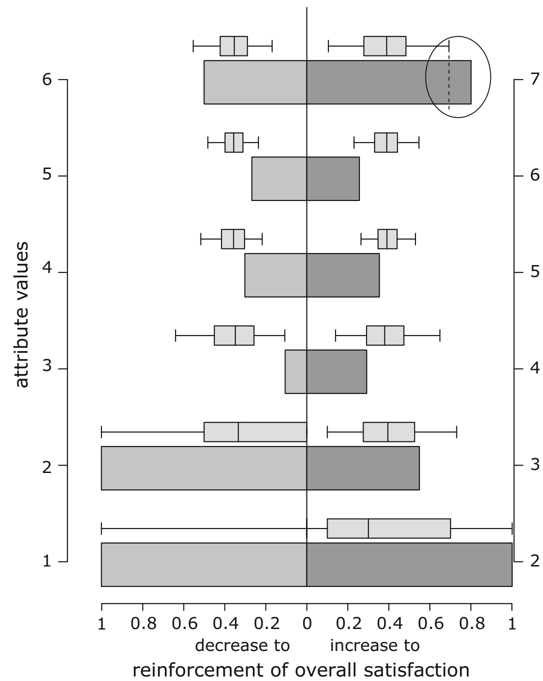

Fig. 10 OrdEval results for *Project management*

These OrdEval results might be surprising since *Requirements acquisition* is considered one of the basic building blocks of the software development (Kruchten 2000; Maglyas et al. 2017) with high importance for software project success (Fernandez et al. 2017). However, in an environment of ever-changing customer requirements and technological changes, there is a need for continuous reflection to decide on the best course of action (Kakar 2017). Thus, the results can be explained with the fact that recently many new requirement acquisition techniques and approaches have been developed (Lauesen and Kuhail 2012; Raspotnig and Opdahl 2013; Ernst et al. 2014; Lucassen et al. 2016) which are perceived to be an attractive quality category by the majority of CIOs. They enable the development of a better common understanding of a problem domain by significantly improving the quality of communication between customers and development teams.

Figure 6 shows that no statistically significant reinforcements of overall satisfaction were detected on the level of attribute values for *System design and architecture*. Thus we cannot identify a specific quality category for this discipline. Nevertheless, ReliefF (Fig. 4) clearly shows that CIO satisfaction with the discipline is positively associated with the net benefits of IT projects.

Figure 7 shows the statistically significant positive influence of SDM application in the *Coding and integration* discipline on CIO satisfaction with the discipline as attribute values increase from 2 to 3 and from 4 to 5, as

Table 1 OrdEval classification of SDM disciplines based on CIOs' perceptions

SDM discipline	Kano quality category
Requirements acquisition	Attractive quality category and for a certain group of CIOs one-dimensional quality category
System design and architecture	Inconclusive
Coding and integration	One-dimensional quality category
Testing	Must-be quality category and for a certain group of CIOs one-dimensional quality category
Deployment	Must-be quality category
Project management	Generally inconclusive and for a certain group of CIOs attractive quality category

well as statistically significant negative influences as the attribute value decreases from 5 to 4. This indicates the presence of a one-dimensional quality category. The *coding and integration* discipline is largely defined by SDM approaches that systemize key software development processes, which positively influence net benefits of IT projects. For instance, a development lag that is a consequence of a continuous evolution of IS in large organizations (Neumann et al. 2014) can be considerably reduced by introducing the practice of continuous integration. Such approaches lead to a higher level of efficiency in the coding and integration discipline (Karvonen et al. 2017). This matches our findings regarding the stable positive contribution of SDM application to CIO overall satisfaction in the *Coding and integration* discipline.

Figure 8 shows the statistically significant influence of SDM application in the *Testing* discipline on CIO overall satisfaction with the discipline as attribute values increase from 1 to 2 and from 5 to 6 or decrease from 6 to 5. The strongest effect shown in the bottom right bars indicates the must-be quality category of SDM in *Testing*, while the effect shown by the bars in the second row from the top indicates that a certain group of CIOs perceive SDM in testing as the one-dimensional quality category. We can conclude that basic levels of SDM application are required in *Testing* to enable project success, while higher levels of SDM application in *Testing* bring additional benefits. Such findings are consistent with findings of (Anand et al. 2013; Barr et al. 2015; Soetens et al. 2016) showing that additional benefits are related to SDM defining higher levels of testing automation.

SDM application in the *Deployment* discipline is generally perceived as the must-be quality category as indicated by the bottom right bar showing a statistically significant and strong effect on CIO overall satisfaction with the discipline in Fig. 9. We conclude that the SDM application to the *Deployment* discipline has to be established at least at the basic level since higher levels of SDM application in the *Deployment* discipline do not appear to be beneficial.

Figure 10 shows the statistically significant positive influence of SDM application in the *Project management* discipline on CIO overall satisfaction with the discipline as attribute value increases from 6 to 7. OrdEval results thus indicate the presence of the attractive quality category. Since the RelieF score does not show that the impact of this attribute is relevant, we can conclude that the benefits of *Project management* are on average inconclusive, but a certain group of CIOs perceives *Project management* to belong to the attractive quality category. Such findings are consistent with findings of Wells (2012) who suggests that project management methodologies help very experienced managers who understand the value of the promotion of standardization and uniformity of processes and procedures in multiple projects. However, according to Wells (2012), the majority of project managers with medium experience do not perceive the project management methodology as helpful. Their perception of the benefits of the project management methodology is directly undermined by their tacit knowledge that they use to intuitively steer project management decisions while ignoring the formal methodology directives.

5 Discussion

Based on the results of our exploratory study presented in the previous section we can answer the research questions posed in the introduction to this paper. The ReliefF algorithm enabled us to evaluate and rank the impacts of the application of different SDM disciplines on enterprise net benefits. Using the proposed methodology, we detected important differences between different SDM disciplines. This clearly shows the need for such an evaluation since it enables enterprises to focus on those SDM disciplines that deliver the highest net benefits (RQ1). The results obtained by the OrdEval algorithm, namely statistically significant reinforcement factors for different attribute values of specific SDM disciplines showed the existence of different quality categories. While some disciplines exhibited a single quality category (i.e., coding and integration, deployment, and project management) others (i.e., requirements acquisition and testing) exhibited the influences of two quality categories. Only one specific SDM discipline (i.e., system design and architecture) did not exhibit any specific quality type. The ability of OrdEval to identify specific quality categories in five of six SDM disciplines clearly indicates that existing software process quality evaluation models inadequately measure quality as a single category, while the proposed methodology proved able to distinguish several quality categories (RQ2).

Existing research in the fields of product, service and workspace quality (Matzler and Hinterhuber 1998; Kim and de Dear 2012) and the field of IT product quality (Lee et al. 2008, 2015; Mayer 2012; Ilbahar and Cebi 2017) shows that product characteristics exhibiting different quality categories require different actions to improve customer satisfaction. Our application of the Kano model in the field of software development processes shows that different parts of software development processes (like disciplines) also exhibit different quality categories (from must-be to attractive quality) and that it is not always the case that benefits increase linearly with the increase in the quality level, specifically for must-be quality category disciplines. However, existing software process quality evaluation models such as CMMI and ISO/IEC assume that benefits increase linearly with an increase in quality of the software development process. Considering existing research in IT product development, our results imply that the proposed methodology should be used complementary to the existing software process quality evaluation models.

Alternatively, the existing models could be expanded to incorporate the key phases of the proposed methodology presented in Fig. 3. Specifically, researchers in the field of software process quality evaluation models should extend the established models like CMMI and ISO/IEC with the key ability of the proposed methodology, i.e., to identify quality categories of specific software development disciplines or process parts. This would assist with the decision for or against the improvement of certain software development process parts, depending on their identified quality categories and individual attribute values (e.g., maturity levels, quality levels). According to the Kano model, existing research in the field of IT product development and our study, the highest benefits can be expected by improving the software development process parts that either belong to a must-be quality category with a low attribute value or to an attractive quality category with medium attribute value. Medium benefits can be expected by improving process parts that belong to the performance quality category. Small or no benefits can be expected by improving process parts that are either belong to a must-be quality category with a medium attribute value or to an attractive quality category with a low attribute value. Such tailoring of SDM improvements cannot be replicated by existing models in the field of the software development process since they do not perceive quality as a concept with multiple categories. This tailoring phase would help practitioners identify the process parts whose improvements will bring the most benefits. In the post-methodology era where the agile software development dominates and reducing excess SDM weight is the management's priority, such tailoring could also improve SDM acceptance in software development teams. Therefore practitioners should start to evaluate quality categories and benefits of individual software development process parts.

Although our exploratory study has certain limitations, it clearly demonstrates that the proposed methodology was able to detect different quality categories of SDM disciplines and their relation to net benefits. One limitation was that the respondents already knew their project's outcome when they participated in our survey. This may have caused response bias especially if the projects were highly successful or highly unsuccessful. In line with similar studies (Jørgensen 2016), we tried to request mainly objective information about the project characteristics. Another limitation of the exploratory study is the low response rate. However, it does not impact the validity of answers to our research questions since the data served mainly to prove that it is possible to identify specific quality categories and evaluate SDM disciplines impact on net benefits without the need to generalize results from the sample to a population. Additionally, low response rates are typical for mail surveys conducted in enterprises (not only in Slovenia). Previous research showed that response rates around 10% still allow researchers to treat the sample as a random sample which assumes that probability of non-response is equal for all units of the studied population (Hovelja 2008; Hovelja et al. 2010). Larger data sets from multiple countries would enable us to generalize the results of our exploratory study.

6 Conclusion

We proposed a novel methodology to identify different quality categories of SDM discipline (parts) with the aim to increase the satisfaction of software development enterprises with the application of their SDM. We categorized the quality of SDM disciplines according to the Kano model. This information can help software development enterprises to identify SDM disciplines with high improvement benefits.

The conducted exploratory study clearly showed the value of the proposed methodology. It demonstrated the need to move from the evaluation of a single quality category to the evaluation of multiple quality categories. Additionally, we found that the impact of different SDM disciplines on enterprise net benefits varies considerably and should be evaluated individually. Thus, we can recommend that future software process quality evaluation models start evaluating multiple categories of quality instead of just one. By improving the understanding of the concept of quality in the field of software process quality evaluation models, practitioners could achieve similar benefits as were achieved in the fields that already employ the Kano model for quality evaluation.

The proposed methodology employs OrdEval and ReliefF algorithms that were used for the first time in the software quality management context. In future work, we intend to improve the OrdEval algorithm to automatically merge values that do not have enough representatives for more reliable estimations of reinforcement factors. Another possible improvement that would increase the objectiveness of perception for different Kano qualities is to survey not only CIOs but also all other relevant stakeholders like project managers, developers, and users. An additional avenue for improving the proposed methodology is to integrate it with the quantitative Kano models that have been developed recently. In order to address the subjective classification present in the Kano model, several quantitative Kano models can be used such as Fuzzy Kano model, Continuous Fuzzy Kano model, Analytical Kano model, etc. (Violante and Vezzetti 2017).

Acknowledgements The authors acknowledge the financial support from the Slovenian Research Agency (research core Funding No P6-0411 and No P2-0359) and from the European Union's Horizon 2020 research and innovation program under Grant Agreements No 825153 and 777204.

References

Anand S, Burke EK, Chen TY et al (2013) An orchestrated survey of methodologies for automated software test case generation. J Syst Softw 86(8):1978–2001

Avison D, Fitzgerald G (2006a) Methodologies for developing information systems: a historical perspective. In: Avison D, Elliot S, Krogstie J, Pries-Heje J (eds) The past and future of information systems: 1976–2006 and beyond. Springer, Boston, pp 27–38

Avison D, Fitzgerald G (2006b) Information systems development: methodologies, techniques and tools, 4th edn. McGraw-Hill Higher Education, New York

Barr ET, Harman M, McMinn P et al (2015) The Oracle problem in software testing: a survey. IEEE Trans Softw Eng 41(5):507–525

Basili V, Heidrich J, Lindvall M, et al (2007) GQM^+ strategies—aligning business strategies with software measurement. In: First international symposium on empirical software engineering and measurement (ESEM 2007), pp 488–490

Bass JM (2016) Artefacts and agile method tailoring in large-scale offshore software development programmes. Inf Softw Technol 75:1–16

Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A et al (2014) A review of microarray datasets and applied feature selection methods. Inf Sci 282:111–135

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth Internacional, California

Bresnahan T, Brynjolfsson E, Hitt L (2002) Information technology, workplace organization, and the demand for skilled labor: firm-level evidence. Q J Econ 117(1):339–376

Brynjolfsson E, Hitt L (1996) Paradox lost? firm-level evidence on the returns to information systems spending. Manag Sci 42(4):541–558

Brynjolfsson E, Hitt L, Yang S (2002) Intangible assets: computers and organizational capital. Brook Pap Econ Act 1:137–198

Chen L-F (2012) A novel approach to regression analysis for the classification of quality attributes in the Kano model: an empirical test in the food and beverage industry. OMEGA Int J Manag Sci 40(5):651–659

Chen C-C, Chuang M-C (2008) Integrating the Kano model into a robust design approach to enhance customer satisfaction with product design. Int J Prod Econ 114(2):667–681

Clarke P, O'Connor RV (2012) The situational factors that affect the software development process: towards a comprehensive reference framework. Inf Softw Technol 54(5):433–447

Cockburn A (2002) Agile software development. Addison-Wesley, Boston

Čufar A, Mrhar A, Robnik-Šikonja M (2015) Assessment of surveys for the management of hospital clinical pharmacy services. Artif Intell Med 64(2):147–158

Ernst NA, Borgida A, Jureta IJ, Mylopoulos J (2014) Agile requirements engineering via paraconsistent reasoning. Inf Syst 43:100–116

Fernandez DM, Wagner S, Kalinowski M et al (2017) Naming the pain in requirements engineering. Empir Soft Eng 22:2298–2338

Forrester EC, Buteau BL, Shrum S (2011) CMMI for services: guidelines for superior service. Addison-Wesley, Upper Saddle River, NJ

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182

Herzberg F, Mausner B, Snyderman BB (1959) The motivation to work. Wiley, Hoboken

Hovelja T (2008) Organisational effects on information technology productivity in enterprises: the case of Slovenia. Econ Bus Rev 10(3):243–261

Hovelja T, Rožanec A, Rupnik R (2010) Measuring the success of the strategic information systems planning in enterprises in Slovenia. Management 15(2):25–46

Hovelja T, Vasilecas O, Rupnik R (2013) A model of influences of environmental stakeholders on strategic information systems planning success in an enterprise. Technol Econ Dev Econ 19(3):465–488

Hovelja T, Vasilecas O, Vavpotič D (2015) Exploring the influences of the use of elements comprising information system development methodologies on strategic business goals. Technol Econ Dev Econ 21(6):885–898

Ilbahar E, Cebi S (2017) Classification of design parameters for E-commerce websites: a novel fuzzy Kano approach. Telemat Inform 34(8):1814–1825

Ishikawa K (1990) Introduction to quality control. Chapman and Hall, London

Järvinen J, Komi-Sirviö S, Ruhe G (2000) The PROFES improvement methodology—enabling technologies and methodology design. PROFES. Springer, Heidelberg, pp 257–270

Jørgensen M (2016) A survey on the characteristics of projects with success in delivering client benefits. Inf Softw Technol 78:83–94

Kakar AK (2017) Do reflexive software development teams perform better? Bus Inf Syst Eng 59(5):347–359

Kano N (2001) Life cycle and creation of attractive quality. In: 4th international QMOD conference quality management and

organisational development proceedings, Linköpings Universitet, Sweden
Kano N, Seraku N, Takahashi F, Tsuji S (1984) Attractive quality and must-be quality. J Jpn Soc Qual Control 14(2):147–156
Karvonen T, Behutiye W, Oivo M, Kuvaja P (2017) Systematic literature review on the impacts of agile release engineering practices. Inf Softw Technol 86:87–100
Kim J, de Dear R (2012) Nonlinear relationships between individual IEQ factors and overall workspace satisfaction. Build Environ 49:33–40
Kneuper R (2009) CMMI improving software and systems development processes using capability maturity model integration (CMMI-DEV). Rocky Nook, Santa Barbara
Kondo Y (2001) Customer satisfaction: how can I measure it? Total Qual Manag 12(7–8):867–872
Kruchten P (2000) The rational unified process: an introduction, 2nd edn. Addison-Wesley, Boston
Laporte C, O'Connor R, García Paucar LH (2016) The implementation of ISO/IEC 29110 software engineering standards and guides in very small entities. Evaluation of Novel approaches to software engineering. Springer, Heidelberg, pp 162–179
Lauesen S, Kuhail MA (2012) Task descriptions versus use cases. Requir Eng 17(1):3–18
Lee Y-C, Sheu L-C, Tsou Y-G (2008) Quality function deployment implementation based on Fuzzy Kano model: an application in PLM system. Comput Ind Eng 55(1):48–63
Lee S-G, Yang C-G, Lee S-B, Lee J-B (2015) A study on the antecedents and consequences of satisfaction and dissatisfaction in web portal usage. Serv Bus 9(3):567–586
Lin S-P, Yang C-L, Chan Y, Sheu C (2010) Refining Kano's "quality attributes-satisfaction" model: a moderated regression approach. Int J Prod Econ 126(2):255–263
Lofgren M, Witell L (2008) Two decades of using Kano's theory of attractive quality: a literature review. Qual Manag J 15(1):59–75
Loon HV (2007) Process assessment and ISO/IEC 15504: a reference book. Springer, New York
Lucassen G, Dalpiaz F, van der Werf JMEM, Brinkkemper S (2016) Improving agile requirements: the quality user story framework and tool. Requir Eng 21(3):383–403
Maglyas A, Nikula U, Smolander K, Fricker SA (2017) Core software product management activities. J Adv Manag Res 14(1):23–45
Matzler K, Hinterhuber H (1998) How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. Technovation 18(1):25–38
Mayer JH (2012) Using the Kano model to identify attractive user-interface software components. In: Digital innovation in the service economy. 33th international conference on information systems (ICIS). Red Hook, Curran, Orlando, USA, pp 1470–1486
Mamavi O, Nagati H, Pache G, Wehrle FT (2015) How does performance history impact supplier selection in public sector? Ind Manag Data Syst 115(1):107–128
Mikulic J, Prebezac D (2011) A critical review of techniques for classifying quality attributes in the Kano model. Manag Serv Qual 21(1):46–66
Miller G (2001) Sizing up today's lightweight software processes. IT Prof 3(3):46–49
Mittal N, Nault BR (2009) Investments in information technology: indirect effects and information technology intensity. Inf Syst Res 20(1):140–154
Nagashima M, Wehrle F, Kerbache L, Lassagne M (2015) Impacts of adaptive collaboration on demand forecasting accuracy of different product categories throughout the product life cycle. Supply Chain Manag 20:415–433
Neumann G, Sobernig S, Aram M (2014) Evolutionary business information systems perspectives and challenges of an emerging class of information systems. Bus Inf Syst Eng 6(1):33–38
Raspotnig C, Opdahl A (2013) Comparing risk identification techniques for safety and security requirements. J Syst Softw 86(4):1124–1151
Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn 53(1–2):23–69
Robnik-Šikonja M, Savicky P (2016) CORElearn: classification, regression and feature evaluation R package. http://CRAN.R-project.org/package=CORElearn. Accessed 1 Feb 2019
Robnik-Šikonja M, Vanhoof K (2007) Evaluation of ordinal attributes at value level. Data Min Knowl Disc 14(2):225–243
Robnik-Šikonja M, Brijs K, Vanhoof K (2009) Ordinal evaluation: a new perspective on country images. In: Perner P (ed) Advances in data mining. Applications and theoretical aspects. Springer, Berlin, Heidelberg, pp 261–275
Robson C (2002) Real world research: a resource for social scientists and practitioner-researchers, 2nd edn. Blackwell, Oxford
Rodriguez P, Haghighatkhah A, Lwakatare LE et al (2017) Continuous deployment of software intensive products and services: a systematic mapping study. J Syst Softw 123:263–291
Rogers EM (2003) Diffusion of innovations, 5th edn. Free Press, New York
Soetens QD, Demeyer S, Zaidman A, Perez J (2016) Change-based test selection: an empirical evaluation. Empir Softw Eng 21(5):1990–2032
Standish-Group (2015) CHAOS report. https://www.standishgroup.com/. Accessed 29 June 2016
Unterkalmsteiner M, Gorschek T, Islam AKMM et al (2014) A conceptual framework for SPI evaluation. J Softw Evolut Process 26(2):251–279
Urbach N, Müller B (2011) The updated DeLone and McLean model of information systems success. In: Dwivedi Y, Wade M, Schneberger S (eds) Information systems theory. Springer, New York, pp 1–18
Vavpotič D, Bajec M (2009) An approach for concurrent evaluation of technical and social aspects of software development methodologies. Inf Softw Technol 51(2):528–545
Vavpotič D, Hovelja T (2012) Improving the evaluation of software development methodology adoption and its impact on enterprise performance. Comput Sci Inf Syst 9(1):165–187
Violante MG, Vezzetti E (2017) Kano qualitative vs quantitative approaches: an assessment framework for products attributes analysis. Comput Ind 86:15–25
Wells H (2012) How effective are project management methodologies? an explorative evaluation of their benefits in practice. Proj Manag J 43(6):43–58
Witell L, Löfgren M, Dahlgaard J (2013) Theory of attractive quality and the Kano methodology—the past, the present, and the future. Tot Qual Manag Bus Excell 24(11–12):1241–1252
Yin RK (2009) Case study research: design and methods, 4th edn. Sage, Thousand Oaks
Zdravkovic J, Svee E-O, Giannoulis C (2015) Capturing consumer preferences as requirements for software product lines. Requir Eng 20(1):71–90
Zhao L, Chen Z, Hu Y et al (2018) Distributed feature selection for efficient economic big data analysis. IEEE Trans Big Data 4(2):164–176