DOI 10.2298/CSIS110503072V

Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance

Damjan Vavpotič[1] and Tomaž Hovelja[1]

[1]University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, 1000 Ljubljana, Slovenia
{damjan.vavpotic, tomaz.hovelja}@fri.uni-lj.si

Abstract. Although the literature studying software development methodologies (SDMs) lists several significant positive effects of the deployment of SDMs, investments into SDMs by the enterprises remain relatively limited. Strategic investments decisions, such as SDMs investments, are mostly taken with the goal of improving enterprise performance. In this paper a model for evaluation of the adoption of SDMs that focuses on the abovementioned SDMs impact on enterprise performance is proposed. The model was empirically tested in four case studies in software development small and medium enterprises (SMEs) in Slovenia. The case studies confirmed that the use of the proposed model enabled SMEs to improve SDMs related investment and adoption decisions and enabled SMEs to invest their limited resources in the most productive and competitive way. The case study experience with the proposed model suggests that its use would also bring similar benefits to larger software development enterprises.

Keywords: Software Development Methodologies, Enterprise Performance, SDM Adoption, Evaluation Approach.

1. Introduction

A SDM can be defined as a collection of procedures, techniques, tools, and documentation aids which will help the system developers in their efforts to implement a new information system [4]. In the past decades various formal SDMs emerged that were based on different underlying philosophies and were developed in both academic and commercial environments. The main motive for their creation was to provide efficient software development procedures that would produce better software for computer supported information systems at an acceptable cost. However, many enterprises dealing with software development do not use formal SDMs and rely mainly on ad hoc development procedures. Fitzgerald [21], for instance reports that 60 per cent of enterprises do not use any SDM and that only 14 per cent use a formalised commercial SDM. Different reasons for this situation have been identified [33, 44, 59] and considerable efforts have been invested into improvement of SDM adoption in software development enterprises [20, 51, 52]. Most of these efforts considered SDM adoption as an IT decision that technicians should decide upon and often neglected the role of business management. Moreover, the assumption that SDM adoption is an IT decision is frequently encouraged by the rest of the enterprise [4]. However, application of SDM typically involves a considerable investment in time, effort and money. Therefore, it is argued that business managers in general should participate more actively in such decisions [4]. To increase business manager's involvement in SDM investments it is important to better understand the impact of SDM on enterprise performance. However, knowing only the potential impact of a SDM procedure on enterprise performance does not enable managers to enact proper SDM related improvement actions. They also need to understand the level of adoption of a specific SDM procedure, since only sufficiently adopted SDM procedures deliver actual enterprise performance benefits. Therefore only the combined understanding of adoption of SDM procedures and their impact on enterprise performance can form the appropriate basis for managers' investment decisions concerning SDM in the enterprises.

In order to significantly improve the managerial organising vision of what for, how and how much should SDM be used the following research questions need to be posed:

1. How can the evaluation of SDM procedure adoption and impact on enterprise performance be modelled? (RQ1)
2. Can such model identify SDM procedures with important enterprise performance benefits? (RQ2)
3. Can such model provide nontrivial information about SDM procedures that was previously unknown to managers responsible for SDM management? (RQ3)
4. Can such model improve SDM related investment and adoption decisions? (RQ4)

To answer these questions the paper starts with the review of the relevant literature in section 2. Based on this review a model is proposed that evaluates the adoption of SDM procedures and their influences on enterprise performance in section 3. Next, in section 4, the paper addresses all the methodological issues that had to be resolved before the conceptual model could be empirically tested. Finally, in section 5, the results of the multiple case study are presented and the paper concludes with a discussion about the technological and managerial implications of the results.

2. Literature review

The research on adoption of SDM and other software process innovations (SPI) in organisations dealing with software development [24, 26, 28] is often based on Rogers' diffusion of innovations theory (DOI) [45]. This universal theory attempts to explain why certain innovations spread among their potential users while others remain unused. Researchers in the field of SPI and SDM consider a SDM or its parts as an innovation and try to predict and explain target adopter attitudes and their innovation-related behaviour [23]. Beside DOI other innovation diffusion models and theories like the Theory of Planned Behaviour (TPB) [2], Technology Acceptance Model (TAM) [15, 54], Perceived Characteristics of Innovating (PCI) [37], and the Theory of Reasoned Action (TRA) [19] can be used to predict/explain adoption of innovations in the field of SDMs. These models focus on SDM adoption mostly from sociological, psychological or cultural aspects of target adopters. However, they typically do not consider business aspects of adoption that are of key importance for management decisions. On the other hand, general claims regarding the positive influence of SDM on enterprise performance have been made not only by SDM vendors (e.g. [30]) but also by different researchers (e.g. [4]). Thus this gap remains to be bridged.

The research on SDM adoption typically observes a SDM as a whole or observes only a single procedure of a SDM (e.g. procedure of unit testing), but does not consider a SDM as a composition of interrelated procedures. For the purpose of this study a SDM procedure is defined as a comprehensive unit that comprises different SDM activities, tasks, tools, documentation templates, design techniques etc. that are used to perform a specific part of software development like functional testing procedure, software documentation procedure, requirements acquisition procedure, etc. Due to the dynamic environment and constantly changing requirements it is hard to adopt only one SDM with strictly defined SDM procedures [34]. Therefore it is important to observe a SDM as a composition of interrelated procedures. This improves the understanding of the adoption of specific SDM procedures and their impact on enterprise performance. In this way the differences in adoption levels and performance impacts between different SDM procedures are not overlooked. The importance of considering a SDM as a composition of different parts is recognised also by the research in the field of situational method engineering [9, 32, 43] that attempts to construct a SDM suitable for a certain situation from procedures and other parts of different existing SDMs. As it is probable that a SDM is constructed from procedures of different SDMs, the performance and adoption of a SDM can be better studied on the level of its individual parts [51] or in our case procedures.

To determine the level of individual SDM procedures' adoption two dimensions were measured:

1. Frequency of SDM procedure use in a case of a given opportunity (FrqUse).
2. Frequency of opportunities for SDM procedure use (FrqOpp).

Even though existing research in the field of SDM adoption commonly uses only a single frequency measure of SDM use (e.g. [27]) new research shows that such approach is inadequate when measuring the frequency of use of individual SDM procedures [51]. The inadequacy stands in the difference of opportunities for use (FrqOpp) and actual use (FrqUse) of different SDM procedures. FrqUse measures how often software developers apply a certain SDM procedure in a case that an opportunity for its use arises during the development. On the other hand FrqOpp measures how frequently an opportunity to use a certain SDM procedure arises during the software development disregarding whether the SDM procedure is actually used or not. Used alone, frequency of SDM use is of limited value as a measure because opportunities for use can vary widely. It needs to be combined with a measure of the number of opportunities for its use.

After appropriate measures of SDM adoption were developed for the proposed model adequate enterprise performance measures were selected to complete the proposed model. The most appropriate enterprise performance measures were selected on the basis of the empirical models from the literature that studies the impact of IT on enterprise performance, theories of the enterprise (firm) and the literature on (IT) project success criteria. In standard neo-classic microeconomics the enterprise is the economy's basic unit of production that combines inputs into outputs at the lowest cost possible in order to maximize its profits [11, 12, 42]. Thus in standard micro-economic models variables of productivity (output/input) and/or profitability (profit/input) are the most widely used as measures of enterprise performance.

Therefore it cannot come as a surprise that the most established measure in the empirical literature that studies the economic impact of IT is added value per employee and its growth [16, 17]. The reason why this measure of productivity established itself over the different measures of profitability can probably be found in the fact that it allows the researchers to avoid the long lasting discussion if enterprises really behave as profit maximizers [14, 47]. When it comes to Slovenian enterprises there is ample evidence that the majority of enterprises do not behave as profit maximizers [5, 8, 41] thus the profitability measures of enterprise performance were not used. To find the appropriate enterprise performance measures a pre-case focus group of 6 SME managers was formed. The focus group found it difficult to objectively evaluate the additional added value (output) generated by a specific software part developed by using certain SDM procedures, despite the fact that the literature posits additional added value as the best enterprise performance criteria. Instead of added value managers saw costs as a more tangible measure that they routinely used to asses SDM performance. Therefore, as the managers were unable to confidently evaluate differences in outputs, the proposed model had to use the differences in costs (input) as the main productivity measure of performance (Cost).

Even though the above developed measure of enterprise performance is in accordance with the majority of empirical models studying the impact of IT on enterprise performance and neo-classic theory the findings of other important organizational, behavioural, managerial and strategic theories of the enterprise should not be ignored. These theories do not see enterprises as one-dimensional profit-maximizers but complex entities with many different and sometimes conflicting goals [6, 14, 22, 25, 38, 42, 46, 58]. Thus grounding the proposed model in different theories of the enterprise requires the measurement of other dimensions of enterprise performance in addition to measurement of enterprise productivity. One of these additional measures of enterprise performance that was included in the proposed model is how a SDM affects enterprises ability to reach their goals (Goal), since reaching the goals is stressed as a key measure of enterprise performance by several theories of enterprises.

Unfortunately many times the goals enterprises set themselves are far from being in line with the interests of the key environmental stakeholders (customers, business partners) [40, 53]. For this reason in addition to measuring the impact of SDM on enterprise productivity and on enterprise ability to achieve their key goals an additional dimension of SDM influence on enterprise performance was introduced. This third dimension measures how SDM benefits the environmental stakeholders through its impact on the improvement of enterprise's products and services (Prod).

Our three goals differ from the classic iron triangle (cost, time, quality) traditionally used as (IT) project success criteria in that they incorporate all the recent findings in the relevant literature [1, 3, 31, 36, 49, 56, 57] about the iron triangle limits. Thus the success criteria were broadened to include measures of organisational strategic success and environmental stakeholder success as follows:

1. The time criterion was broadened to include the organisational (strategic) goals (Goal).
2. Instead of just scope or quality the environmental product value for the customers and business partners of the produced software [3, 7, 31] was measured (Prod).
3. From the original classic iron triangle only costs remained unmodified in the proposed model, since the managers of the pre-case focus group preferred costs as a measure of productivity to additional added value (Cost).
4. The three above developed measures of SDM impact (Cost, Goal, Prod) together with the two SDM adoption measures (FrqOpp, FrqUse) form the core of the proposed model for evaluation of SDM procedure adoption and their impact on enterprise performance. The proposed model can be seen in Figure 1.

3. The proposed model for evaluation

Figure 1 shows the three steps that were employed to study the impact of SDM procedures on enterprise performance. The first step was to catalogue SDM procedures used in the studied enterprises. In this step a focus group comprised of technical managers, SDM users and external experts made a list of all SDM procedures available to the enterprise. Technical managers were employees that had a comprehensive overview of both technical and business aspects of studied SDM procedures, while SDM users were directly involved in the use of these SDM procedures. The cataloguing process was guided by external experts that assured that a comprehensive and complete list of SDM procedures was generated. In the second step each catalogued SDM procedure was evaluated.

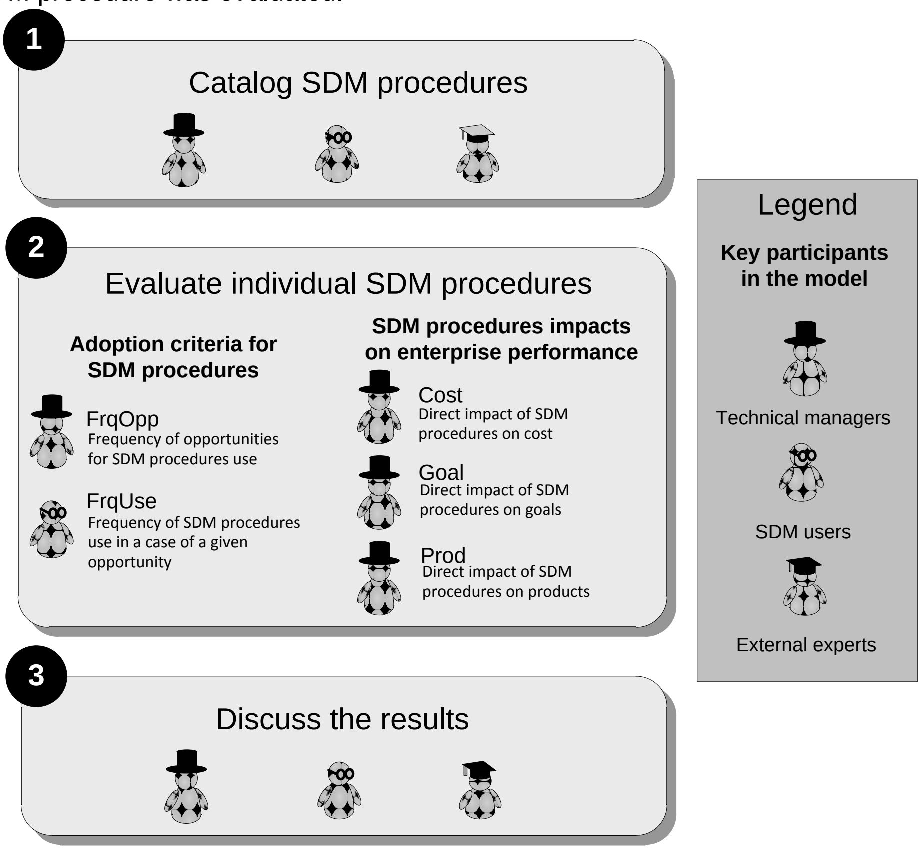

Fig. 1. The proposed model for evaluation of SDM procedure adoption and their impact on enterprise performance

The evaluation was performed by individual SDM users and technical managers. SDM users evaluated only frequency of use in case of given opportunity (FrqUse) of the SDM procedures that were related to their everyday work. While frequency of opportunities for SDM procedures use (FrqOpp) was evaluated by technical managers as only they possessed sufficient knowledge about these procedures to evaluate their FrqOpp objectively. For the same reason they also evaluated the direct impact of these SDM procedures on costs (Cost), goals (Goal) and products (Prod). The last step was an in-depth discussion of results from the second step with evaluation participants. With the development of the proposed model the first research question (RQ1) posed in this paper is addressed.

There are no theoretical reasons that would limit the scalability of the proposed model concerning the number of SDM procedures evaluated and the number of evaluation participants. For this reason the usefulness of the model should not be affected by the size of the enterprises. However, due to the fact that SMEs software development enterprises are the most widespread type of software development enterprise in Slovenia their study was a priority. Therefore the paper presents case studies performed in SMEs. Nevertheless, further testing of the proposed model in larger enterprises should be performed to confirm this assumption of scalability.

4. Methodology

The research methodology is based on embedded multiple case design with replicability analysis as defined by Yin [60]. Adoption of SDM procedures and their impact on enterprise performance in different software development contexts that are common in SMEs in Slovenia were studied. SDMs were studied on level of their procedures as stated in the literature review. The study had to take into account that adoption of a SDM is a long-term process with impacts which can be observed only over longer time periods. Thus it focused on perceptions of primary characteristics of SDM procedures (adoption levels, economic impact) of key employees that have a good understanding of their SDM. Researchers in the field of IT innovation adoption often use perceived characteristics instead of directly measuring primary characteristics. According to Downs and Mohr [18] the findings of many studies, which have examined the primary characteristics of innovations, have been inconsistent, as primary attributes are intrinsic to an innovation independent of their perception by potential adopters. The behaviour of individuals, however, is predicated by how they perceive these primary attributes. Since different adopters might perceive primary characteristics in different ways, their eventual behaviours might differ. Similar approach was used also in other SDM studies [29, 51].

The proposed model was used in four different cases: a software development enterprise dealing with development of web-based applications (case A), two different software development enterprises that produce their own pre-packaged business solutions software (cases B and C), and an enterprise producing financial software (case D). The basis for the evaluation of SDM procedure adoption (FrqUse and FrqOpp) and their impact on enterprise performance (Goal, Cost and Prod) were questionnaires that were used in each single case. They included the 2 dimensions of SDM adoption and the 3 dimensions of enterprise performance and were filled out partially by technical managers and partially by SDM users of the studied enterprises. The questions about FrqUse (how often is a SDM procedure actually used in a case of a given opportunity) and FrqOpp (how often opportunities for use of SDM procedure arise) were evaluated with 7-point ordinal scales (never=1, very seldom=2, seldom=3, sometimes=4, often=5, very often=6, always=7). The questions about SDM procedure impact on enterprise performance were also close-ended questions, however they used a seven-point Likert scale between 7 (strongly agree that a specific SDM procedures affects a specific dimension of enterprise performance) and 1 (strongly disagree that a specific SDM procedures affects a specific dimension of enterprise performance), where 0 meant neither agreement or disagreement with the statement.

The results are presented on a scatter chart comprising FrqUse as a vertical dimension and FrqOpp as a horizontal dimension. To help direct management efforts in improvement of individual SDM procedures four quadrants were formed in the scatter chart by using the medians of FrqUse and FrqOpp. Medians were used instead of means of scale because of the positive perception bias encountered during case studies which is discussed in detail in section 6. The four quadrants follow the logic suggested by Vavpotic and Bajec [51]: the first quadrant contains inefficient and unadopted SDM procedures, the second quadrant contains inefficient but adopted procedures, the third quadrant contains unadopted but efficient procedures and the fourth quarter contains adopted and efficient procedures. Efficient SDM procedures are those that have a lot of opportunities for use during the software development process (FrqOpp higher than median) while adopted SDM procedures are those that are actually used in the software development process (FrqUse higher than median).

For procedures in the first quadrant managers should investigate whether it makes sense to continue investing in them. An example of such procedure could be a procedure that is less appropriate for the needs of an organisation and has never been popular between SDM users. A possible course of action would be to discard such SDM procedures (e.g. an old procedure for system design based on structured methods in an enterprise that now uses object oriented development environment). For procedures in the second quadrant managers need to focus on creating more opportunities for their use as these procedures are already well used when an opportunity for their use arises. For instance, these can be SDM procedures that had a lot of opportunities for use in the past and are still accepted by SDM users, but are technically less appropriate for development of new systems. A recommended course of action to improve such procedure might be to replace accepted procedures with technically more appropriate procedures that do not require of SDM users to drastically change the way they work (e.g. procedures that use Enterprise generation language [39] might be introduced instead of procedures that employ Java to replace procedures based on an old programming language that developers are familiar with). The third quadrant shows procedures that have a lot of opportunities for use but are not used. An example could be a new procedure that is very suitable for development of new systems but are not yet accepted by SDM users. In such case managers should try to improve acceptance of such SDM procedures by additionally training and educating SDM users (e.g. procedures for test automation might be introduced, but not accepted by testers that are accustomed to manual test procedures therefore the testers should be additionally trained in the field of test automation). The fourth quadrant shows procedures that have a lot of opportunities for use and are well used by SDM users. Such procedures should be monitored so that when they fall from the fourth quadrant management can deploy an appropriate course of action (e.g. procedure for data modelling that is technically efficient and also accepted by SDM users should be monitored to detect technological advances that could make this procedure obsolete).

Although the suggested separation of SDM procedures into the four quadrants enables managers to select a general course of action [51] it does not help them to understand the value of individual SDM procedures for the enterprise which would in our opinion significantly improve managerial SDM related investment and adoption decisions (RQ4). For this reason each SDM procedure is presented by a point in one of the four quadrants of the scatter chart that is additionally described by its average evaluated impact on cost (marked as C), goal (marked as G) and product (marked as P).

Finally, the results of the evaluation and possible courses of action are discussed with the evaluation participants. In this way the evaluation participants are involved not only in the collection of facts, but also in the co-construction and interpretation of the case narrative [10].

The four cases are described in the following subsections. The cross-case discussion is included in the last subsection.

5. Case studies

5.1. Case A

Case A is a small enterprise focusing on development of web-based applications. Their target customers are enterprises that require web presence from simple web pages to more advanced customer oriented web applications like web-shops, customer web support etc. They use their own SDM that was mainly developed inside the enterprise and is based on structured SDMs [4]. It comprises 21 procedures describing activities that have to be performed and artefacts produced, for instance: testing the user interface, introducing new code into configuration management system, documenting program code, changing the data model etc. These procedures also define various programming, testing and documenting standards. The SDM is regularly updated and modified to facilitate the application of new IT.

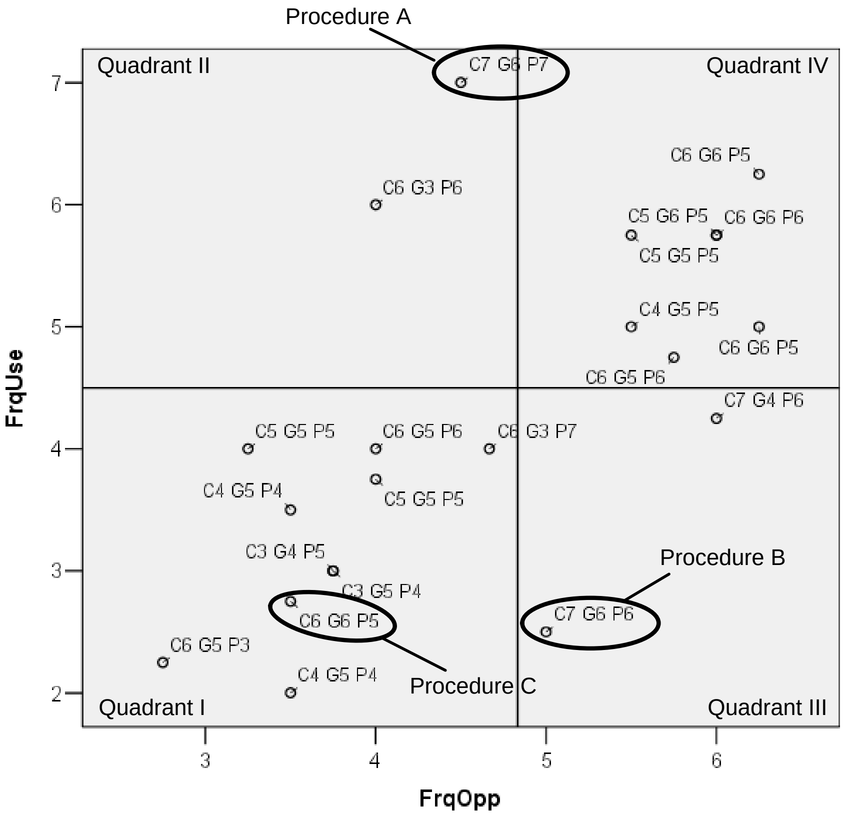

Fig. 2. Scatter chart showing evaluation of SDM procedures in Case A

Four technical managers having good knowledge about enterprise's SDM procedures and their impact on enterprise's work and four SDM users participated in the evaluation. With their help the procedures of their SDM were catalogued. Next, they individually evaluated the SDM procedures related to their work. The results of their combined evaluations are presented in a scatter chart in Figure 2 as described in the methodology section (for the purpose of clearer presentation of evaluations on the scatter chart only the integer parts of cost, goal and product evaluations are shown).

After closely examining their answers considerable differences between the 21 procedures were detected. Based on the discussion of the evaluation results the management focused on improving SDM procedures with the highest impact on cost, goals and products that were not in the fourth quadrant of the scatter chart. To illustrate the differences and the possibilities for different courses of SDM related managerial actions three individual procedures marked as A, B and C in Figure 2 that management selected for improvement are shown.

Procedure A is describing the use of a database schema generation tool only for generation of a new database schema. In such cases Procedure A is used every time. However, when only modifications to existing database schemas are required these modifications are preformed manually and therefore Procedure A is not used. The procedure was evaluated as having high impact on all three performance criteria since it speeded up the database schema development and improved its quality. Based on the discussion of the individual evaluations of Procedure A with the evaluation participants the management decided to adapt Procedure A to also facilitate its use for modifications of existing database schemas.

Procedure B is describing the use of project management and bug-tracking tool that was newly introduced in the development team at the time. Procedure B enabled the project participants to monitor their current work tasks schedule and to report progress. It also enabled project managers to assign work tasks more efficiently and to detect work tasks that have fallen behind the schedule. The evaluation of procedure showed that it had relatively high opportunities for use, but was not used on all projects. The management expected that the procedure would be used on larger projects while on smaller projects the work tasks would be still managed manually. However, evaluation of actual use showed that despite management expectations it was not used even on most large projects as there was significant resistance among developers. The procedure was evaluated as having high impact on all three performance criteria since it improved productivity of developers and quality of the end product by increasing the efficiency of work task allocation. It also reduced the time that developers spent at work place without actually working through better developer's reports of performed work. Based on the discussion of the individual evaluations of Procedure B with the evaluation participants the management decided to invest in additional training of developers and to make the use of the procedure mandatory for every large project.

Procedure C is describing the preparation and use of formal project development time plan. The formal project development time plan was prepared only for larger projects while on smaller projects the development time plan was presented in an informal manner. Furthermore, in both cases the plan was not kept up to date over the course of the project which resulted in its low use. The procedure was evaluated as having high impact on all three performance criteria since it enabled the management to improve the coordination and organization of the projects. Based on the results of the evaluations the management decided to require project managers to apply Procedure C in regular intervals over the project lifecycle.

As with Procedures A, B and C management used the proposed model to manage other procedures shown in Figure 2. They especially focused on procedures that were identified as having high performance benefits. This supports our second research question (RQ2). They also confirmed that the proposed model provided them with nontivial information about SDM procedures management (RQ3) which enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.2. Case B

Case B is an enterprise that builds and supports its own ERP solution for small businesses. It uses object-oriented development environment and a self-developed SDM partially grounded on simplified Rational Unified Process [30] and partially on agile methodologies [13]. It comprises 21 procedures (coincidentally the same number as in case A). These procedures are best described as well-defined comprehensive activities that also include short descriptions of artefacts produced by the activities and roles that perform the activities. In addition the SDM prescribes the use of supportive tools for requirements acquisition, task assignment and test automation. Interestingly, the majority of these tools were developed inside the enterprise and are therefore highly adapted to its needs and SDM. The SDM is only used to develop a single product, so there is not much need for SDM tailoring for different projects; however there is need to fulfil individual clients' requirements. It is updated on a regular basis to support new trends in IT. Three technical managers and six SDM users each experienced in the use of different procedures of the enterprise's SDM took part in the evaluation.

Similarly as in case A the procedures were catalogued first, next the evaluation participants answered the given questionnaires and finally discussed the evaluated results. After the discussion of the evaluation results the management focused on improving specific SDM procedures with the highest impact on cost, goals and products. Three individual procedures marked as A, B and C that show the course of managerial actions undertaken are presented in Figure 3. Management expected the greatest enterprise performance benefits from the improvements of these three procedures.

Procedure A is describing the use and development of common component libraries that can be reused in creation of different system functions. There are a lot of opportunities to apply this procedure except when development of custom code is required for the needs of subsystems and prototypes that use specific information technologies (e.g. technologies for integration with specific legacy systems). Although the Procedure A is often used the position of the procedure on the scatter chart shows that there is still a lot of room for improvements. Discussion of the evaluation results showed that developers do not apply all component libraries consistently since they are not familiar with some of them. Furthermore management pointed out that consistent application of all component libraries would improve product stability and maintainability and reduce the cost and time of the development, thereby allowing the company to take on additional projects. Based on these findings the management decided to strengthen the control over the produced code by introducing formal code reviews that would insure that proper component libraries are used.

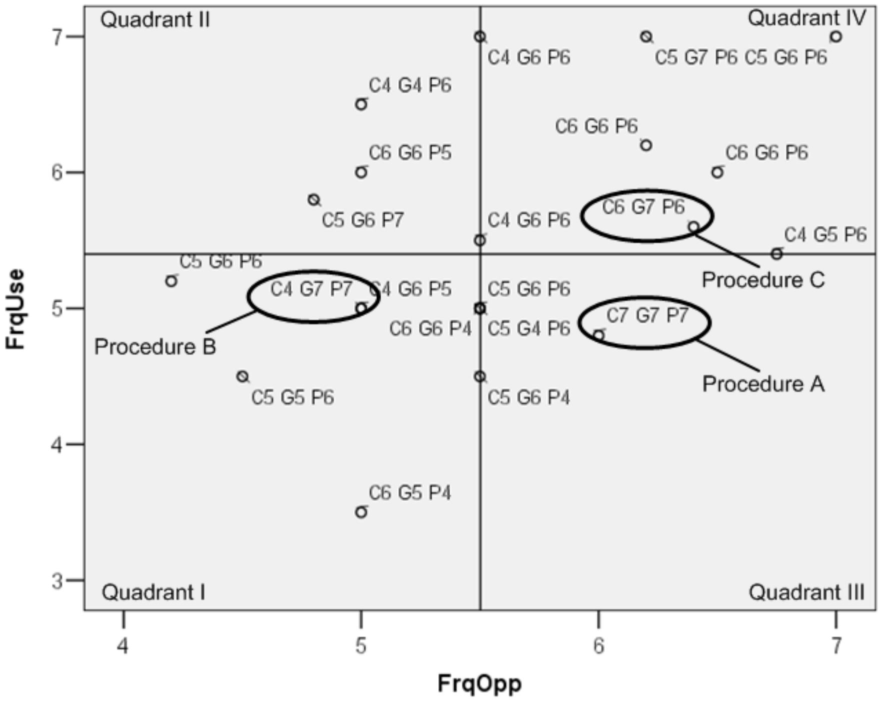

Fig. 3. Scatter chart showing evaluation of SDM procedures in Case B

Procedure B is describing how to document customer requirements for different software system functions in detail. This detailed description of software system functions is used to better understand the requirements and is typically performed only for more complex functions while less complex functions are only briefly described. The evaluation of Procedure B showed that the procedure was too cumbersome to be used for description of simpler functions therefore opportunities for its use were limited to complex procedures. Furthermore the developers did not use the procedure consistently as they perceived it as additional work that was not used by programmers in implementation phase of software development. The procedure was evaluated as having high impact on goals and product since it improved technical quality of the software and its use resulted in software that better matched customer expectations. Based on the discussion of the individual evaluations of Procedure B with the evaluation participants the management decided to simplify the procedure. For instance, they substituted parts of textual description of software functions with graphical representation of user interface. In this manner management hoped to lower the costs and increase the opportunity for use of Procedure B also for less complex software functions. Additionally, management hoped that developers would find such adapted Procedure B more useful.

Procedure C is describing the documentation of source code that betters the understanding, easies maintenance and lowers the cost of upgrades of software. The procedure has many opportunities for use, is used often and importantly impacts enterprise performance. Even though the management was satisfied with the adoption levels of this procedure they additionally wanted to expand its use on parts of the code developed for database management systems. In this way they wanted to make sure that the procedure remains firmly embedded in the fourth quadrant for the foreseeable future due to its importance for enterprise performance.

The management used the proposed model to manage other procedures shown in Figure 3 in a similar way as Procedures A, B and C. They prioritised the procedures that had a high impact on performance (RQ2) and confirmed that the proposed model provided them with nontrivial information about SDM procedures (RQ3) which enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.3. Case C

Case C is a mid-sized enterprise that develops its own pre-packaged business solutions for SMEs. Its speciality is that the management tries to standardize and organize its development process mainly through the use of various commercial development tools. It uses object-oriented design and development tools, an automated testing environment, tools for management of requirements and changes, tools for project management etc. As the enterprise's development process mainly relies on the use of the development tools, the description of the SDM is relatively coarse comparing to the other three cases. It comprises eight SDM procedures that offer general description of different working fields like programming, testing, requirement acquisition etc. The SDM is grounded on Information engineering [35], though it was significantly simplified and partially reorganised to support object-oriented development environment. Three technical managers each of whom was responsible for several of the SDM procedures and five SDM users participated in the evaluation.

As in case A and B the procedures were catalogued, evaluated and results discussed by the evaluation participants. After the discussion of the evaluation results the management focused on improving specific SDM procedures. They especially focused on improving the SDM procedures with the highest impact on costs, since the competition forced them to reduce their development costs. Three individual procedures marked as A, B and C that show the course of managerial actions undertaken are presented in Figure 4. Management expected the greatest enterprise performance benefits from the improvements of these three procedures.

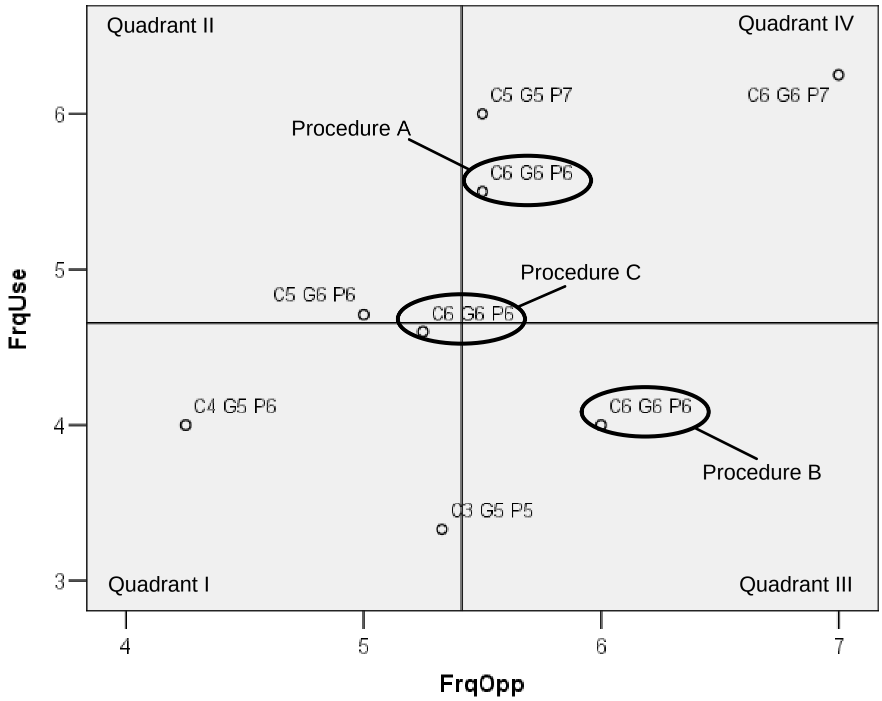

Fig. 4. Scatter chart showing evaluation of SDM procedures in Case C

Procedure A is describing the approach and notation for modelling business processes. The opportunities for use of the procedure are not uncommon however there are still possibilities for improvements. The enterprise policy was to formally model only key and complex business processes while simpler business processes were purposely left out and only addressed through communication between development teams and customers in later phases of development, mainly in requirements acquisition phase and partially in implementation phase. The main motive for such policy was that the management did not want the developers to spend too much effort on business modelling but rather on design and development of programming code. The discussion showed that such policy was not always adequate since it sometimes resulted in development of different programming code for similar activities used in various business processes and functions. To reduce the cost of coding of individual activities they decided to additionally standardise the modelling of certain simpler business processes and activities which facilitates reuse of programming code. In addition the management decided to prescribe a standardised set of architectural patterns [48] for future modelling.

The previously described enterprise policy to formally model only key and complex business processes had also a strong impact on Procedure B.

Procedure B describes the approach and notation for requirements acquisition, however developers did not use it regularly and instead often preferred to use their own simpler ad-hoc approaches that were spontaneously established through long term cooperation with customers. The reason for the limited use of Procedure B was that the development teams interpreted the aforementioned policy as justification for not spending much effort on requirements acquisition for simpler business processes while key and complex business processes were in their opinion already sufficiently described through the use of Procedure A. The discussion showed that such approach caused several problems mainly related to inconsistency of the used notation [50] which resulted in misinterpretation of requirements in the implementation phase as well as difficulties in cost monitoring and project management due to the fact that such ad-hoc requirements acquisition made it hard to clearly specify projects tasks and estimate development teams workloads. To address this situation management decided that requirements should be described formally and that only tasks related directly to formally described requirements can be reported and will count as work hours done by the development team.

Procedure C describes the approach and notation for creation of logical database design that can be used for generation of a database schema in a database management system. Although the procedure assumes the use of a database modelling and generation tool it does not prescribe a specific tool. The enterprise promoted the use of several different database design tools in the past few years, however the use of the tools remained limited and simpler database schemas continued to be created and modified manually. The discussion showed that this was mostly due to general lack of deeper knowledge about any of the available tools as a significant number of potential users i.e. developers did not know how to use the tools to perform more complex tasks. To address this situation the managers decided to prescribe the use of one specific database development tool and organize an in-depth training for the developers.

Other procedures shown in Figure 4 were evaluated in a similar way as Procedures A, B and C. The priority was given to the procedures that had the highest impact on performance (RQ2). Management also confirmed that the proposed model provided them with nontrivial information about SDM procedures (RQ3). This enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

5.4. Case D

Case D is an enterprise developing financial business software. The enterprise uses a structured SDM [4] that has been considerably modified and improved over the years to support new development approaches and IT. The enterprise has a relatively long history in software development and uses a variety of IT. Their SDM is divided into three branches that are specialized for the needs of different financial software products and used by different developer groups. Two of the three branches of SDM are intended for projects that are completely run inside the enterprise; the remaining branch is intended for projects in which the implementation is outsourced. The SDM consists of 24 comprehensive procedures that cover different parts of work for each group of developers. Five technical managers and 15 SDM users took part in the evaluation. The selected technical managers had advanced knowledge of different procedures of their SDM and understood the procedure’s impact on enterprise performance.

The SDM procedures were catalogued, then evaluated and discussed. Three individual procedures marked as A, B and C seen in Figure 5 were considered as the best candidates for improvement by the management.

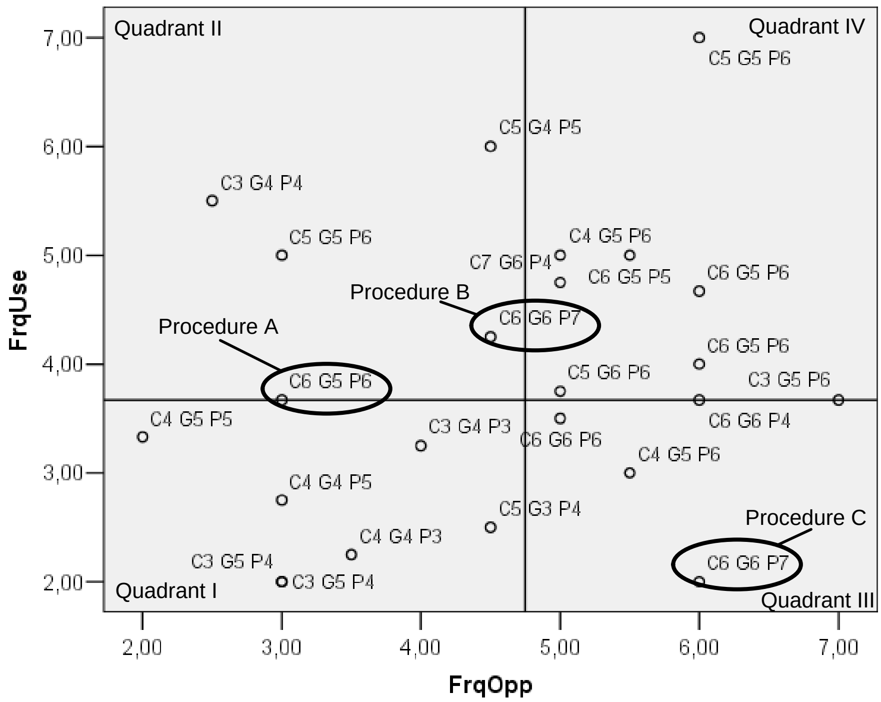

Fig. 5. Scatter chart showing evaluation of SDM procedures in Case D

Procedure A describes the use of a tool for database modelling and schema generation. Although the tool was not limited to modelling and generation of new databases, the Procedure A did not consider this fact and prescribed the use of the tool only for generation of new databases. The opportunities for use of Procedure A were therefore evaluated as quite limited. Furthermore, actual use of the procedure was also low due to the fact that older developers comprising the majority of workforce resisted change. They wanted to keep their “old and proven routines” of manual database schema development. The management considered this situation very problematic since a fast enforcement of the change could create a hostile work environment. Thus they decided to address the situation as a part of a package of long-term initiatives. These initiatives were designed to lower the resistance to change [55] by lowering the average age of the developers, training the developers in the use of the prescribed procedures and reorganizing the development teams through systematic promotion of the developers open to change. Furthermore they also decided to increase the opportunities for use of Procedure A by adapting it so that it can also be used for modifications of existing database schemas.

Procedure B describes the approach to automated software testing with tool support. The opportunity to use the procedure was limited due to significant number of custom development projects, where automation of testing was considered to be too expensive. The management considered the use of test automation as a cost effective way to improve quality of pre-packaged software. However, management similarly as for Procedure A considered this to be a problematic situation and decided to address it again as part of the long-term initiatives described in the preceding paragraph.

Procedure C describes the approach of software implementation and integration. The opportunity for use was very high since the procedure can be used in every software development project except when software implementation is outsourced. Contrastingly, actual use was very low. The discussion uncovered that the cause of the low use were again the established routines and ad-hoc approaches. These were already detected as the main reason for the resistance to change in the analysis of the preceding procedures A and B. For this reason Procedure C was also addressed through the mentioned long term initiatives.

Although the management was aware of significant resistance to change by the majority of older developers before using the proposed evaluation model they were not aware of the severity of the resistance to change. The realization of the problem's severity motivated them to start the above described long-term initiatives (RQ3). Equally important, the proposed evaluation model enabled them to identify the procedures that can contribute most to enterprise performance if their use and/or opportunities for use are improved (RQ2, RQ3). This enabled them to significantly improve their SDM related investment and adoption decisions (RQ4).

6. Cross-case study results and discussion

In all four cases similar patterns were observed. The proposed model was successfully used by evaluation participants to identify SDM procedures with important enterprise performance benefits (RQ2). Even in case 3 that had the coarsest SDM procedure descriptions the identification of SDM procedures with important enterprise performance benefits was not problematic although it required more discussion. The identified SDM procedures were the ones that then received most of the management improvement efforts which confirmed that the proposed model provides nontrivial information to the management (RQ3). Additionally, managers in all four enterprises confirmed that SDM related investment and adoption decisions were improved by use of the proposed model (RQ4). The management also confirmed the usefulness of the presentation of the proposed model in a scatter chart linking adoption levels and enterprise performance impacts of individual SDM procedures. The four cases thus clearly show the replication of results consistent to the expectations formulated in our research questions.

The application of the proposed model in practice showed that the evaluation participants on average expressed a positive bias when evaluating their SDM procedures. An exception was the FrqUse in Case D where resistance to change caused a small negative bias. The discussions showed that the evaluation participants perceived themselves as capable software developers that can tackle any procedure prescribed by the technical managers as long as they perceive it as useful. To overcome this problem of possible positive or negative bias the SDM procedures adoption needs to be observed relatively to each other and not on an absolute scale. Therefore the medians were proven to be a better option to group the SDM procedures into the four quadrants than the centres of the adoption scales.

Because of limited resources it was not possible to conduct more case studies. However, according to Yin [60] more than two cases already make a strong argument. The study was limited to software development SMEs which predominantly undertook direct revenue earning projects. Further research should broaden the spectrum of SMEs and include SMEs that also undertake other project types and also test the proposed model in larger enterprises and government institutions.

7. Conclusion and further work

The paper proposes a model for SDM evaluation that concurrently takes into account adoption levels and enterprise performance impacts of SDM procedures. The proposed model allows a software development enterprise to comprehensively evaluate its SDM procedures and to develop appropriate actions for their improvement. The proposed model has been applied in four software development SMEs. These cases showed that the proposed model can significantly improve management understanding of SDM related issues and significantly improve the management of SDM by allowing the managers to focus on the SDM procedures with highest impact on enterprise performance.

The proposed model builds on existing SDM adoption models and augments them by introducing enterprise performance impact of SDM procedures as an important additional dimension. This additional dimension enables managers to focus their actions on improving key SDM procedures and allows them to employ portfolio management practices to management of SDM procedures. Furthermore, it offers them suggestions on how to improve SDM procedures by dividing SDM procedures into four separate groups (quadrants on a scatter chart) that require different improvement actions.

Further research should focus on the question whether the proposed model can be successfully used in larger enterprises dealing with software development. Moreover, it is possible to expand the set of SDM adoption measures by considering how frequently SDM procedures could be used in an optimally organised software development process. Such measurement could introduce the quality of an organisation as an important additional factor that moderates the interaction between SDM adoption and enterprise performance.

References

1. Agarwal, N., Urvashi, R.: Defining 'success' for software projects: An exploratory revelation. International Journal of Project Management, Vol. 24, No. 4, 358-370. (2006)
2. Ajzen, I.: The Theory of Planned Behavior. Organizational Behavior and Human Decision Processes, Vol. 50, No., 179-211. (1991)
3. Atkinson, R.: Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. International Journal of Project Management, Vol. 17, No. 6, 337-342. (1999)
4. Avison, D.E., Fitzgerald, G.: Information systems development : methodologies, techniques and tools. 4th ed. McGraw-Hill, 656. (2006)
5. Babič, K.: Managerski odkupi v slovenskih razmerah. Ekonomska fakulteta: Ljubljana. 44. (2003)
6. Baumol, W.J.: Business Behavior, Value and Growth. MacMillan, New York, 164. (1959)
7. Belassi, W., Tukel, O.I.: A new framework for determining critical success/failure factors in projects. International Journal of Project Management, Vol. 14, No. 3, 141-152. (1996)
8. Brezar, M.: Vpliv političnega kadrovanja na poslovanje podjetij v državni lasti: analiza na primeru slovenskih družb Mercator d.d. in Petrol d.d. Ekonomska fakulteta: Ljubljana. 38. (2007)
9. Brinkkemper, S., Lyytinen, K.,Welke, R.J.: Method Engineering – Principles of method construction and tool support. IFIP TC8. Chapman & Hall, Atlanta, USA. (1996)
10. Bygstad, B., Munkvold, B.E.: Exploring the role of informants in interpretive case study research in IS. Journal of Information Technology Vol., Available online from 2010/08/24. (2010)
11. Chandler, A.D.: The Emergence of Managerial Capitalism The Business History Review, Vol. 58, No. 4, 473-503. (1984)
12. Coase, R.H.: The nature of the firm. Economica, Vol. 4, No. 16, 386-405. (1937)
13. Cockburn, A.: Agile software development. Agile software development series. Addison-Wesley, Boston, 304. (2002)
14. Cyert, R.M., March, J.G.: A Behavioral Theory of the Firm Blackwell, Oxford. (1963)

15. Davis, F.A.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly Vol. 8, No. 3, 318-339. (1989)
16. Dedrick, J., Gurbaxani, V.,Kraemer, K.L.: Information Technology and Economic Performance: A Critical Review of the Empirical Evidence. ACM Computing Surveys (CSUR), Vol. 35, No. 1, 1-28. (2003)
17. Dewett, T., Jones, G.R.: The role of information technology in the organization: a review, model, and assessment. Journal of Management, Vol. 27, No. 3, 313-346. (2001)
18. Downs, G.W., Jr., Mohr, L.B.: Conceptual Issues in the Study of Innovation. Administrative Science Quarterly, Vol. 21, No., 700-714. (1976)
19. Fishbein, M., Ajzen, I.: Belief, attitude, intention, and behavior : an introduction to theory and research. Addison-Wesley Pub. Co., Reading, Mass., xi, 578 p. (1975)
20. Fitzgerald, B.: Am empirical investigation into the adoption of systems development methodologies. Information & Management, Vol. 34, No. 6, 317-328. (1998)
21. Fitzgerald, B.: Systems Development Methodologies: The Problem of Tenses. Information Technology & People, Vol. 13, No. 3, 174-185. (2000)
22. Freeman, R.E.: Strategic management: A stakeholder approach. Pittman Publishing, Marshfield, MA. (1984)
23. Gallivan, M.J.: Organizational adoption and assimilation of complex technological innovations: development and application of a new framework. The DATABASE for Advances in Information Systems, Vol. 32, No. 3, 51-85. (2001)
24. Gallivan, M.J.: The influence of software developers' creative style on their attitudes to and assimilation of a software process innovation. Information & Management, Vol. 40, No. 5, 443-465. (2003)
25. Grant, R.M.: Toward a Knowledge-Based Theory of the Firm. Strategic Management Journal, Vol. 17, No. Special Issue, 109-122. (1996)
26. Green, G.C., Hevner, A.R.,Collins, R.W.: The impacts of quality and productivity perceptions on the use of software process improvement innovations. Information and Software Technology, Vol. 47, No. 8, 543-553. (2005)
27. Huisman, M., Iivari, J.: The individual deployment of systems development methodologies. In: Lecture Notes in Computer Science. Vol. 2348. Springer. 134-150. (2002)
28. Huisman, M., Iivari, J.: The organisational deployment of systems development methodologies. In Advances in Methodologies, Components, and Management. Kluwer, 87-99. (2003)
29. Huisman, M., Iivari, J.: Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. Information & Management, Vol. 43, No. 1, 29-49. (2006)
30. IBM: Rational Unified Process v 7.0.1 (IBM Rational Method Composer plugin). IBM Corp. (2006)
31. Jugdev, K., Müller, R.: A retrospective look at our evolving understanding of project success. Project Management Journal, Vol. 36, No. 4, 19-31. (2005)
32. Karlsson, F., Agerfalk, P.: Exploring agile values in method configuration. European Journal of Information Systems, Vol. 18, No. 4, 300-316. (2009)
33. Khalifa, M., Verner, J.M.: Drivers for software development method usage. IEEE Transactions on Engineering Management, Vol. 47, No. 3, 360-369. (2000)
34. Lavbič, D., Lajovic, I.,Krisper, M.: Facilitating information system development with Panoramic view on data. Computer Science and Information Systems, Vol. 7, No. 4, 737-767. (2010)

35. Martin, J.: Information Engineering, Book I: Introduction. Prentice Hall. (1989)
36. Milis, K., Vanhoof, K.: Analysing success criteria for ICT projects. Int. J. Nuclear Knowledge Management, Vol. 2, No. 4. (2007)
37. Moore, G., Benbasat, I.: Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation. Information Systems Research Vol. 2, No. 3, 192-222. (1991)
38. Perrow, C.: The Analysis of Goals in Complex Organisations. American Sociological Review, Vol. 26, No. 6, 854-866. (1961)
39. Pluta, J.: Developing Web 2.0 Applications with EGL for IBM i. MC Press, Lewisville, 200. (2009)
40. Porter, M.E., Kramer, M.R.: Strategy and Society: The Link Between Competitive Advantage and Corporate Social Responsibility. Harvard Business Review, Vol. 2006, No. December, 78-92. (2006)
41. Prašnikar, J., Svejnar, J.: Investment, Wages and Ownership During the Transition to a Market Economy: Evidence from Slovenian Firms, in William Davidson Institute Working Papers Series. (1998)
42. Putterman, L., Kroszner, R.S.: The Economic Nature of the Firm. Cambridge University Press, Cambridge, 400 (1996)
43. Ralyte, J., Deneckere, R.,Rolland, C.: Towards a Generic Model for Situational Method Engineering. In 15th International Conference on Advanced Information Systems Engineering. (2003)
44. Riemenschneider, C.K., Hardgrave, B.C.,Davis, F.D.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Transactions on Software Engineering, Vol. 28, No. 12, 1135-1145. (2002)
45. Rogers, E.M.: Diffusion of innovations. 5th ed. Free Press, New York, xxi, 551 p. (2003)
46. Simon, H.A.: Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations. The Free Press, New York. (1947)
47. Simon, H.A.: Rational choice and the structure of the environment. Psychological Review, Vol. 63, No. 2, 129-138. (1956)
48. Šaša, A., Krisper, M.: Enterprise architecture patterns for business process support analysis. Journal of Systems and Software, Vol. 84, No. 9, 1480-1506. (2011)
49. Trkman, M., Trkman, P.: A wiki as intranet: a critical analysis using the Delone and McLean model. Online Information Review, Vol. 33, No. 6, 1087-1102. (2009)
50. Vasilecas, O., Dubauskaitė, R.,Rupnik, R.: Consistency checking of UML business mode. Technological and Economic Development of Economy, Vol. 17, No. 1, 133-150. (2011)
51. Vavpotic, D., Bajec, M.: An approach for concurrent evaluation of technical and social aspects of software development methodologies. Information and Software Technology, Vol. 51, No. 2, 528-545. (2009)
52. Vavpotic, D., Vasilecas, O.: An Approach for Assessment of Software Development Methodologies Suitability. Elektron. Elektrotech., Vol. 8, No. 114, 107-110. (2011)
53. Veblen, T.: The Theory of the Business Enterprise. Transaction Books, New Brunswick, New Jersey. (1904)
54. Venkatesh, V., Davis, F.D.: A theoretical extension of the Technology Acceptance Model: Four longitudinal field studies. Management Science, Vol. 46, No. 2, 186-204. (2000)

55. Vrhovec, S., Rupnik, R.: A model for resistance management in IT projects and programs. Electrotechnical Review, Vol. 78, No. 1-2, 24-37. (2011)
56. Wateridge, J.: IT projects: a basis fo success. International Journal of Project Management, Vol. 13, No. 3, 169-172. (1995)
57. White, D., Fortune, J.: Current practice in project management - an empirical study. International Journal of Information Management, Vol. 20, No. 1, 1-11. (2002)
58. Williamson, O.E.: Markets and Hierarchies: Analysis and Antitrust Implications. The Free Press, New York. (1975)
59. Wynekoop, J.L., Russo, N.L.: Systems-Development Methodologies - Unanswered Questions. Journal of Information Technology, Vol. 10, No. 2, 65-73. (1995)
60. Yin, R.K.: Case study research : design and methods. 4th ed. Sage Publications, Thousand Oaks, Calif., 240. (2008)

Damjan Vavpotič. Dr. Damjan Vavpotič is a senior-lecturer of Information Systems and Information Systems Development in the Faculty of Computer and Information Science at the University of Ljubljana. He received his doctorate in Information Systems Engineering from University of Ljubljana, Slovenia. His research interests include information systems development methodologies, methodology adoption and evaluation, and agile methodologies. His research has appeared in journals such as Information and Software Technology, Informatica, Applied Informatics, and Nurse Education Today, and in proceedings of many international conferences. He participated in many projects dealing with evaluation and improvement of ISDMs, IS strategic planning, and IS development.

Tomaž Hovelja. Dr. Tomaž Hovelja received bachelor's degree, master’s degree and his PhD in Business Administration from the Economic Faculty at the University of Ljubljana. He is employed as an assistant at the Faculty of Computer and Information Science at the University of Ljubljana. His research areas are social, economic and organizational factors of IT deployment in enterprises and IT projects success criteria. His research has appeared in journals such as Economic Annals, Economic and Business Review for Central and South-Eastern Europe, Management - Journal of Contemporary Management Issues and others.

Received: May 3, 2011; Accepted: November 28, 2011.