ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2011. No. 8(114)
ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
T 120
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

An Approach for Assessment of Software Development Methodologies Suitability

D. Vavpotic
Faculty of Computer and Information Science, University of Ljubljana,
Trzaska c. 25, 1000 Ljubljana, Slovenia, e-mail: damjan.vavpotic@fri.uni-lj.si

O. Vasilecas
Information Systems Research Laboratory, Vilnius Gediminas Technical University,
Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania, olegas.vasilecas@vgtu.lt

crossref http://dx.doi.org/10.5755/j01.eee.114.8.707

Introduction

Different software development methodologies (SDM) have been developed in the past decades to optimize and improve the process of software development. One of the reasons for the variety of available SDM is the need for development of different types of software systems which can range from small and simple (e.g. software for simple web-shops) to highly specialised and complex software systems (e.g. software for information systems used in manufacture based on complex mathematical models [1]). Consequentially, organisations dealing with software development face the problem how to select a SDM that best suits their requirements. Unfortunately, these organisations often lack knowledge and experience to be able to objectively evaluate different types of SDMs and select a suitable one. This leads to selection of a SDM that is only partially suited to the organisation's requirements. To improve this situation we propose an approach that aids organisations dealing with software development to select a SDM that is suited to their requirements and expectations.

Both, researchers and practitioners have addressed the problem of selecting a suitable SDM. For example, Cockburn [2] provides a decision model that helps select the suitable SDM from a family of SDMs named Crystal. Another example can be found in [3], where a specialised evaluation model that considers agile SDMs is discussed. An example can also be found in Rational Unified Process (RUP) in form of guidelines for tailoring RUP to the needs of a development project. Different guidelines for SDM selection are also provided by the research in the field of method engineering [4]. However, all these approaches consider only a relatively small number of different SDM properties and often do not provide recommendations for selection of a specific SDM but give only general guidelines or are limited only to one SDM family. Nevertheless, they can serve as basis for creation of general decision rules regarding SDM selection.

The course of our research was as follows. In the first part of the research we reviewed literature regarding SDM selection and studied different decision model designs. We proposed to use multi-attribute approach [5] that is suitable to manage large property sets. Based on the reviewed literature and our experience we proposed sets of properties and decision rules and integrated them into a coherent approach. In the second part of the research we tested the approach on already completed real-life projects. We validated the results produced by the approach by discussing them with experts in the field of SDM. In the third part of the research we applied the approach in a real-life case to help select a suitable SDM. This part of the research was performed as a case study [6].

The paper is organised in the following manner. A general overview of the research and related works presented in introduction is followed by "Properties used in the approach" section that gives an overview of the proposed SDM and project properties. Next, in "Decision rules" section we discuss the organisation and use of the decision rules and in "Processing the properties and the decision rules" section we explain how the recommendations for the selection of a SDM are computed. Practical application of the approach and its results are presented in "Case study: application of the approach in a software development company" section. Finally, "Conclusion and further work" section summarizes the results of the research and presents further possibilities for improvements of the approach.

Properties used in the approach

We propose an approach that comprises two sets of properties: one describing SDMs, and another one describing software development projects. The approach determines the suitability of a property of a certain SDM for a property of a certain software development project by considering a number of decision rules. We propose the set of project properties and the set of SDM properties shown in Tables 1 and 2 which are based on the review of existing sources and models discussed in introduction and on our experience.

Table 1. Project properties

Property group	Properties
Size and type of the system	System size, System complexity, Criticality of the system, Types of client applications, Architecture, DBMS type, Legacy support
Priorities of a project	Traceability required, Repeatability required, Available time, Available budget
Development team and environment	Technical experience, Problem domain experience, Technical knowledge, Problem domain knowledge, Cooperation, Discipline, Team culture, Team size
Customer	Cooperativeness, Problem domain knowledge, Technical knowledge

Table 2. SDM properties

Property group	Properties
Process – general properties	Process lifecycles, Development perspective, Artefact traceability
Process – primary disciplines scope and detail	Business modelling, Requirements acquisition, Analysis, Design, Implementation and integration, Testing, Deployment
Process – supportive disciplines scope and detail	Project management discipline, Configuration management, Change management, Process management
Techniques and methods	Diagramming techniques, Project management techniques
Tools and development environment	Requirement for specialised tools, Program languages, Development frameworks
Type of development	Types of client applications, Architecture, DBMS type, Legacy support
Support and learning	Available support, Available literature, Available courses

We are aware that it is possible to define additional properties that describe a project and a SDM that are not enlisted in Table 1 and Table 2. However, our aim was to identify properties, which can be used as a valuable input in the decision making process and can be defined for most projects and SDMs. Nevertheless, different projects often stress different aspects of the development. Therefore we propose that a weight is assigned to each project property. This enables the user of the approach to put emphasis on properties that are more important for a specific project.

We describe projects and SDMs by using a matrix in which the selected properties are assigned value 1 and properties that are not selected are assigned value 0.

Decision rules

The decision rules form the backbone of the approach. In the first step we developed a set of rules that is used for selection of SDM. We focused our efforts on collection, composition and compilation of existing rules and our proposed rules to define the content of the rules used in the approach. It is important that these rules come from practice and have already been tested in real life environment. In the second step we transformed these rules usually represented in natural language to formal representation required by the approach. We avoided changing the content and the meaning of the decision rules during the transformation, however certain adaptations and generalizations of some of the rules were required so that they could be used in our approach.

We propose to use a two-dimensional matrix as the most appropriate way for the representation of our decision rules due to a large number of these rules. The header row and the header column of the matrix contain all SDM properties and project properties correspondingly. Each property has enlisted all values it can take. In this manner a matrix containing decision rules for each combination of a SDM property and a project property is formed. A decision rule for a combination of two properties is represented in a sub-matrix containing evaluations for each combination of values for the two properties. The decision rule is written in a form of five different values ranging from very unsuitable (-2) to very suitable (2) that can be assigned to each combination of SDM property value and project property value.

Table 3 presents the structure of a decision rule sub-matrix. It shows an example of such sub-matrix in which we consider a SDM property named *Level of detail of requirements acquisition process definition* (DRA) and a project property named *Team size* (TSZ).

Table 3. An example of decision rules sub-matrix

		DRA		
		ND	LD	DD
TSZ	VST	0	2	-2
	ST	-1	2	0
	MST	-2	1	1
	LT	-2	1	2

DRA is a property of a SDM that can take three different values: *No definition* (ND) of the requirements acquisition process, *Limited definition* (LD) of the requirements acquisition process and *Detailed definition* (DD) of requirements acquisition process. These values are shown in the header rows of the Table 3.

TSZ is a project property that can take four different values: *Very small team* (VST), *Small team* (ST), *Medium-sized team* (MST) and *Large team* (LT). TSZ is defined as number of people in a team executing the software development process where very small team is up to five people and large team is more than 50 people. These values are shown in the header columns of the Table 3.

As discussed three different values can be selected for SDM property DRA and four different values for project property TSZ. All combinations of DRA and TSZ form an evaluation sub-matrix with twelve possible evaluations of suitability of SDM property for a project property. For instance, based on literature [2] we define the following rule: "In case that a SDM does not describe requirements acquisition process at all such SDM is very unsuitable for a large development team." So we evaluate the combination of DRA value *No definition* and TSZ value *Large team* in the evaluation sub-matrix as very unsuitable (-2). Similarly, rules for other combinations of project property values and SDM property values are defined.

Processing the properties and the decision rules

In this section we discuss our proposed approach for processing the properties of project and SDM, and the decision rules discussed in the preceding sections in order to produce the recommendations for the selection of a SDM that suits the needs of a software development project. The main result of this approach is evaluation that indicates how suitable a certain SDM is for the given project. This enables us to relatively quickly limit the set of SDM candidates. The processing starts with computation of Methodology-Project-Value (MPV) for each combination of SDM and project. We propose the following formula for computation of MPV for a single combination of project and SDM

$$MPV_{m,p} = \sum_{x \in X} \sum_{y \in Y} \left(R_{x,y} \cdot M_{x,m} \cdot P_{y,p} \cdot W_y \right), \qquad (1)$$

where $X = \{\forall x : x$ – row number of SDM property value$\}$; $Y = \{\forall y : y$ – row number of Project property value$\}$; m = column number which contains description of the SDM under evaluation; p – column number which contains description of the project under evaluation; $R_{x,y}$ – value in column x and row y of the decision rules matrix; $M_{x,m}$ – value in row x and column m of the matrix describing properties of a SDM; $P_{y,p}$ – value in row y and column p of the matrix describing properties of a project; W_y – value in row y of the weights table containing the weight for project property value.

To explain and better understand the computed MPV we propose to compute positive or negative contribution of each individual combination of SDM property and project property for selected SDM and project. By using such explanation it is possible to quickly detect the SDM properties that are evaluated as the most unsuitable or the most suitable for a certain software development project. We propose the following formula to compute the Methodology-Project-Properties-Value (MPPV) of individual combinations of properties for a single combination of project and SDM

$$MPPV_{m,p}(x,y) = \left(R_{x,y} \cdot M_{x,m} \cdot P_{y,p} \cdot W_y \right), \qquad (2)$$

where all variables have the same meaning as in the previous formula for computation of MPV.

Computing MPV for all candidate SDM and all considered projects enables us to compare suitability of different SDM for different projects.

Case study: application of the approach in a software development company

The approach was used in a company dealing with development of custom information systems. At the time the company employed about 35 employees. As the problems related to lack of formal SDM were becoming more and more apparent the company management decided to introduce a standardised SDM that would facilitate further growth of the company. However, due to variety of SDM available and variety of projects that the company undertook it was difficult to select a SDM that would best suit the company's needs. To ease this problem our proposed approach was used and the results presented to the company's management.

The following six SDMs were considered: SCRUM, test driven development (TDD), extreme programming (XP), RUP, RUP for small projects and Oracle Custom Development Method (CDM). Additionally their existing software development practices were considered as the seventh SDM.

The suitability of the SDMs for five different project types was examined. Four of these are typical project types performed in the company (types A, B, C and D) while the fifth is a hypothetical project (type X). Type A is a small size project with relatively stable requirements; type B is medium size project using technologies that developers did not use on preceding projects; type C is small to medium size project with changing requirements; type D is large project with changing requirements; type X is hypothetical very large project type used only for comparison.

Fig. 1 shows normalised MPVs computed by using the approach for different combinations of project types and SDMs on a radar chart. The chart clearly shows that some SDM cover requirements of company's typical projects better than other and that some SDM are more suitable for larger projects (types D and X) while other are more suitable for smaller projects (types A, B and C). In the middle of the chart we can see that existing practices cover only a small area of the chart which indicates the possibility for significant improvements of these practices. The SDM that best suits the company's needs is RUP for small projects as it has the highest MPV for project types A and B and has relatively high MPV for project type C.

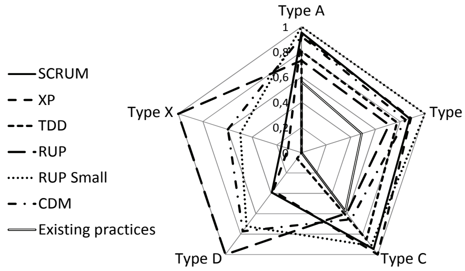

Fig. 1. Radar chart showing normalized MPVs for the company's typical project types

Based on the analysis of suitability of different SDM for different company's projects that was created with aid of the approach the company's management decided to select RUP for small projects as their reference SDM. They noted that the results produced by the approach were very valuable in the SDM selection process as they helped them to better understand the suitability of different SDM for the company's typical projects.

Conclusions and future work

Organisations dealing with software development often do not have enough knowledge, experience, time and other resources for deep study of different SDMs and for selection of one that best suits the needs of their projects. To support selection process we propose an approach based on properties of project, properties of SDM and decision rules that aids such organisations in the process of SDM selection. Application of the approach in practice showed that it can be considered as an important aid in the process of SDM selection. One of the key advantages of using the approach is that its users can quickly examine suitability of a number of different SDMs to a number of different projects even though they do not have in-depth knowledge of the examined SDMs. Advanced users are able to obtain a broader overview of SDMs' suitability which enables them to select a SDM that is more suitable for a software development project and/or organisation.

In our further work we intend to focus our efforts on further improving the approach by introducing additional socio-technical aspects of the development team that have been found to play an important role in successful SDM adoption [7]. We also intend to examine possibilities to use our approach for selection of other types of methodologies like methodologies in the field of ontology development [8], model driven development [9] and reverse engineering [10]. Furthermore, we will examine the possibility to use the approach for selection of specialised software development and modelling frameworks [11]. Such expansion of our approach will further improve its value for organisations dealing in software development.

References

1. **Kalnius R., Eidukas D.** Probability Model Quality of Informations Systems. // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 4(92). – P. 13–18.
2. **Cockburn A.** Agile software development. – Boston: Addison–Wesley, 2002. – 304 p.
3. **Mikulenas G., Butleris R.** An approach for constructing evaluation model of suitability assessment of agile methods using analytic hierarchy process // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 10(106). – P. 99–104.
4. **Bajec M., Vavpotic D.** A framework and tool–support for reengineering software development methods // Informatica. – Vilnius: Institute of Mathematics and Informatics, 2008. – No. 19(3). – P. 321–344.
5. **Triantaphyllou E.** Multi–Criteria Decision Making Methods: A comparative Study. Kluwer Academic Publishers, 2000. – 320 p.
6. **Yin R. K.** Case study research : design and methods. – Thousand Oaks, Calif.: Sage Publications, 2008. – 240 p.
7. **Vavpotic, D., Bajec M.** An approach for concurrent evaluation of technical and social aspects of software development methodologies // Information and Software Technology. – Elsevier, 2009. – No. 51(2). – P. 528–545.
8. **Lavbič, D., Krisper M.** Facilitating ontology development with continuous evaluation // Informatica. – Vilnius: Institute of Mathematics and Informatics, 2010. – No. 21(4). – P. 533–552.
9. **Pavalkis S., Nemuraite L., Tarvydas P., Noreika A.** Specification of Finite Element Model of Electronic Device using Modeldriven Wizard–based Guidance // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 4(92). – P. 59–62.
10. **Paradauskas B., Laurikaitis A.** Extracting Conceptual Data Specifications from Legacy Information Systems. // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 1(107). – P. 46–50.
11. **Šilingas, D., Butleris R.** Towards Implementing a Framework for Modeling Software Requirements in Magicdraw UML // Information Technology and Control, 2009. – No. 38(2). – P. 153–164.

Received 2011 04 14

D. Vavpotic, O. Vasilecas. An Approach for Assessment of Software Development Methodologies Suitability // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 8(114). – P. 107–110.

Nowadays many different software development methodologies (SDM) exist that can be used to optimise and improve software development processes in organisations that deal with software development. However, problem is that these organisations often do not have enough knowledge and experience in the field of SDM to be able to objectively evaluate different SDM and select the one that is suitable for their requirements and expectations. To solve this problem we propose an approach that aids such organisations to determine how suitable different SDMs are for their software development requirements. The application of the approach in a software development company showed that it can notably improve the process of SDM selection in a real-life situation because the process is more transparent and results are more grounded. Ill. 1, bibl. 11, tabl. 3 (in English; abstracts in English and Lithuanian).

D. Vavpotic, O. Vasilecas. Programų sistemų kūrimo metodikų tinkamumo vertinimo būdas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 8(114). – P. 107–110.

Šiuo metu yra parengta daug skirtingų programų sistemų kūrimo metodikų (SDM), skirtų programų sistemų kūrimo procesams optimizuoti ir tobulinti organizacijose, kurios dirba šioje srityje. Tačiau problema yra ta, kad tokios organizacijos dažnai stokoja SDM srities žinių ir patirties, kad galėtų objektyviai įvertinti skirtingas SDM ir pasirinkti vieną, kuri tenkintų jų poreikius ir lūkesčius. Šiai problemai spręsti straipsnyje siūlomas būdas, kuris padėtų programų sistemas kuriančioms organizacijoms geriau suprasti, kiek skirtingos SDM tenkina organizacijų ir jų vykdomų projektų reikalavimus. Pasiūlyto būdo realizacija, naudojant sistemos prototipą, buvo pritaikyta programų sistemų kūrimo įmonėje. Išanalizavus gautus rezultatus paaiškėjo, kad realioje situacijoje pasiūlytasis būdas gali gerokai patobulinti SDM pasirinkimo procesą, padaryti jį skaidresnį, o rezultatą pamatuojamą. Il. 1, bibl. 11, lent. 3 (anglų kalba; santraukos anglų ir lietuvių k.).