

Selecting a Methodology for Business Information Systems Development: Decision Model and Tool Support

Damjan Vavpotič¹ and Olegas Vasilecas²

¹University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, 1000 Ljubljana, Slovenia
damjan.vavpotic@fri.uni-lj.si

²Vilnius Gediminas Technical University, Information Systems Research
Laboratory
Sauletekio al. 11, LT-10223 Vilnius-40, Lithuania
olegas.vasilecas@vgtu.lt

Abstract. The paper presents a decision model and a tool that helps to find an information systems development methodology (ISDM) for a computer-based business information system (IS) that is suitable to a certain IS development project or an organisation dealing with IS development. The intention of the model is not only to suggest a certain ISDM, but also to propose the properties an ISDM should have to suite the project or the organisation. It is designed in a way that facilitates experimentation with different project, organisation and ISDM properties. Based on the model we created a tool that has been applied on several cases in which we validated the correctness of its recommendations and established that it can have a significant positive contribution in the process of ISDM selection and in the process of improvement of existing ISDM.

Keywords: business information systems development, development methodology, decision model.

1. Introduction

An information system can be defined as a set of interrelated components that collect, manipulate, store and disseminate data and information and provide a feedback mechanism to meet an objective [34]. These components include users who perform information related tasks, and different information technologies (IT) that help business users perform these tasks or in some cases perform the tasks autonomously without user intervention. In the context of the paper we focus on IS that are computer-based and support business operations. In the continuation of the paper the abbreviation IS is used for such information systems.

Nowadays, computer systems are the most important IT component and are used to store data and to convert data to useful information. Development of computer support is therefore an important part of IS development. It is a relatively complex process that comprises tasks like gathering requirements for new IS, analysing the requirements, designing the new IS, implementing the new IS, etc. To optimize the development process of IS various IS development methodologies (ISDM) have been developed in the past decades that prescribe various approaches that can be applied during the development to improve the IS development process and the end product i.e. IS to be developed.

An ISDM 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 [3]. In the paper we consider ISDMs that focus on development of computer-based IS. These ISDMs can vary from heavyweight that precisely define every single step of the development, to lightweight ISDMs that only vaguely define the most important parts of the development process. The lightweight ISDMs gained their importance with the emergence of agile software development that stresses the using of the most suitable ISDM for a certain type of IS development project and organisation dealing with IS development [8]. The problem is, however, how to select an ISDM type that suits the requirements of an IS development organisation and its type of development projects. Organisations dealing with IS development often lack knowledge and experience to be able to objectively evaluate different types of ISDMs. Often the selection of an ISDM is based only on an advice of a consultant company trying to sell its own one. Consequentially, such approach often results in selection of an only partially suitable ISDM and can be considered as one of the important reasons for low ISDM adoption levels in IS development organisations. Fitzgerald [10], for instance, found out that 60 per cent of companies do not use ISDM at all and that only six per cent reported following ISDMs rigorously.

In this paper we present a decision model and tool support that IS development organisations can use to consider wider range of ISDM, and thus select a suitable ISDM that meets their requirements and expectations. The purpose of the model is not to appoint the use of a certain ISDM, but merely to suggest what kind of ISDM is suitable for a certain type of IS development project and/or IS development organisation.

The paper is organised as follows. Section 2 presents the background and the research method. It is followed by the description of the decision model in section 3 and description of tool support in section 4. Section 5 in which we present verification of the model in practice is followed by the conclusion.

2. Related work and research method

The problem of selecting a suitable ISDM has been addressed in different ways in the past. However, the proposed approaches and models typically consider only a relatively small number of different ISDM aspects, limit themselves only on selection of ISDM from an ISDM family, or offer general rules and guidelines but do not advise about specific ISDMs.

Cockburn [8], for instance, provides a decision model that helps select the suitable ISDM from a family of ISDMs named Crystal. To select an ISDM that suits the project, the model considers three main project properties: number of people involved in the project, criticality of the project, and priorities of the project. However, the model is limited to selection of an ISDM from the Crystal family of ISDM, although similar properties can be used also for selection of other ISDM. Another example is Rational Unified Process (RUP) [14] that provides guidelines for tailoring the ISDM (RUP in specific) to the needs of a development project. It considers various project parameters and provides rules for selecting the most suitable development lifecycle. However, it is limited only to customization of RUP and does not consider other ISDMs. Yet another example is offered by Mikulenas and Butleris [26], who present the idea of constructing a specialised evaluation model of suitability assessment that considers agile ISDMs only. As an alternative to these relatively specialised models and guidelines, more general guidelines for ISDM selection can be found. For instance, McConnell [25] provides general guidelines on how to select the most suitable development lifecycle and gives practical tips on best practices for various development environments. Similarly, different authors of agile ISDM (e.g. [11, 12, 30]) provide general recommendations for adaptation and use of these ISDM. Although these general guidelines and tips are very useful they do not include recommendations for selection of a specific ISDM. An example of a comprehensive method for evaluating software engineering methods and tools is DESMET [21]. It provides nine methods of evaluation and a set of criteria to help evaluators select an appropriate method. However it is intended to be used by an evaluator to plan and execute an evaluation but does not provide advice about specific ISDMs and their suitability to a certain organisation or project. Other approaches exist that help improve the suitability of an ISDM to a certain IS development project and IS development organisation like method engineering [4, 6]. The result of using such approaches is an ISDM that is tailored to the needs of such project and organisation from existing ISDM components. They consider a wide array of project and organisation properties that form the basis for selection of the most suitable ISDM components. These approaches, however, are not intended to verify whether certain ISDM is more or less suitable for a certain IS development project or organisation dealing with IS development; they rather focus on construction of a tailored ISDM. Furthermore, IT governance frameworks and best practices like Control Objectives for Information and Related Technology (COBIT) [16] and Information technology infrastructure library (ITIL) [28] exist that help improve IT related processes and provide

valuable knowledge regarding managing and improving these processes. Similarly, Information Services Procurement Library (ISPL) [17] that is based on former Euromethod [9] provides best practices in the field of acquisition processes related to Information Technology. Although all these frameworks provide valuable information and can be used to access and improve the general quality of different IT related processes they are not intended to provide advice about selection of a specific ISDM that is suitable for a specific project or organisation.

We first presented idea of the decision model that would help select a suitable ISDM for an IS development project at ISD conference in 2003 [38] where we proposed four basic requirements for such decision model. Firstly, the model should be extendable so that users could add their own properties and decision rules and adapt the model to the needs of their organisation. Users should also be able to include additional ISDM and project types. Secondly, the results of the model should be transparent i.e. users should be informed about the reasons why a certain ISDM is preferred over the other in given context and what are the ISDMs desired properties. Thirdly, the model should facilitate experimentation with various properties and their weights, thus enabling the user to receive the most valuable results for certain situation. Finally, the model should facilitate work with incomplete information, which enables the user to receive at least partial results even though he does not have enough information to enter all properties. Since this initial idea, the model has been considerably extended, improved and used on various practical cases.

The research was divided into two phases. In the first phase we used literature review to assemble various decision rules and models regarding ISDM selection. Then we integrated these rules into a comprehensive decision model and developed a tool that implemented the model. We tested the decision model by applying the tool on three real life projects. Based on the findings from these tests results we improved the decision model and used it in further research and practical applications. The aim of the second phase of the research was to validate the recommendations produced by the model and to confirm our proposed hypothesis that the model can significantly contribute to ISDM selection and improvement process in an IS development organisation. This part of the research was performed as a case study [41].

3. The decision model

The selection of an ISDM that suits a certain IS development project and/or a certain organisation dealing with IS development requires careful consideration of a variety of the project and the organisation properties and a variety of ISDM properties. Decision situations in which a large number of properties have to be considered can be well managed by multi-attribute decision models [37, 42] that also form the basis for our decision model. An

approach that is often used in similar situations is analytic hierarchy process (AHP) by Saaty [31]. Although we build sets of project/organisation properties and ISDM properties in a similar way as criterion hierarchies are built in AHP using only AHP approach is not sufficient as we have to determine which ISDM properties suite certain project/organisation properties. To determine the suitability of ISDM properties to project/organisation properties we have to use various decision rules. Therefore we propose a decision model that is a hybrid between a weighted score model and a rule-based model. The decision model uses two separate sets of properties: a set of properties that describe ISDMs (discussed in subsection 3.2) and a set of properties that describe projects and/or organisations (discussed in subsection 3.3). Both sets are organised as weighted score models where the weights of ISDM properties are pre-set and are not intended to be changed while weights of the properties that describe projects and/or organisations can be changed by users of the model who in this manner can emphasize the properties that they see as the most important for his project and/or organisation. To determine how suitable a certain ISDM property is for certain IS development project/organisation property the decision model considers various decision rules. We organised these part of the decision model as a rule-based model where the rules are organised in a matrix discussed in subsection 3.4. These rules present a link between ISDM and project/organisation properties.

The decision model facilitates two different types of users shown on Figure 1. The first type is an ISDM expert who has advanced knowledge in the field of ISDM and provides the values for the properties that define a certain ISDM – ISDM descriptions. His responsibility is to select ISDM descriptions that are considered by the decision model during the decision making process and if required he can also to define additional ISDM descriptions. The ISDM expert can also define new decision rules for ISDM selection. Although the general decision rules for ISDM selection are predefined and are part of the decision model the ISDM expert can define additional decision rules that are specific for a certain organisation or project. The second type of user is a business user who requires recommendation about suitability of different ISDMs for his IS development project or IS development organisation. The responsibility of the business user is to provide the values for properties that describe the IS development project or the IS development organisation – project and organisation descriptions. He can also set weights that define which of the project or the organisation properties are more important in his case and consequentially have greater influence on the final decision produced by the decision model.

The decision model can be used in two basic ways. Firstly, it can be used to recommend the most suitable ISDM for the given IS development project or IS development organisation. This recommendation is based on: ISDM descriptions, IS development project or organisation descriptions, and the decision rules that are part of the decision model. Secondly, the decision model can produce a set of ISDM properties that are recommended for certain IS development project or IS development organisation. This set of recommended properties can then be used to adapt and improve an ISDM

that already exists in the organisation. Details on how the decision model produces these recommendations are discussed in subsection 3.4.

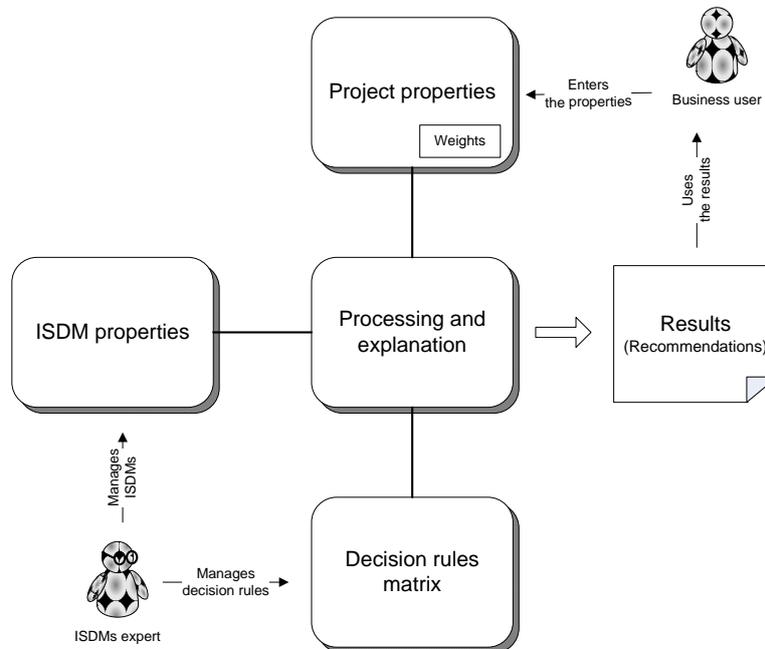


Fig. 1. The main components of the proposed decision model for ISDM selection

We limited the scope of the decision model to ISDM for conventional development of computer-supported IS and did not consider ISDMs for special purposes like ISDMs for ERP systems (e.g. [1]) or people oriented ISDMs (e.g. [27]). The types of ISDM that were considered include: agile ISDMs (e.g. Extreme programming [5]), object oriented ISDMs (e.g. IBM Rational Unified Process [14]), data and process oriented ISDMs (e.g. Information Engineering [24]) and other custom ISDM implemented and used by organisations in which we applied the decision model (e.g. see section 5).

3.1. Project properties

Based on the review of existing studies and models discussed in section 2 and our experience from practical tests of the model discussed in section 5 we propose the set of project properties shown in Table 1. We propose to divide the properties into five main property groups that are further divided into properties for which the predefined values are assigned. For instance, property group *Size and complexity of the system* has property *Planned future* that can take any of the predefined values: *Maintenance only*, *Minor upgrades*, and *Major upgrades or new versions based on this version*. For

certain properties we also defined quantitative measures that help the user of the decision model to select the suitable value. For instance *System size* can be expressed in man-month, where the definition of size is based on existing definitions (e.g. [40]) and is as follows: very small = less than 30 man-month, small = 30 to 100 man-month, medium = 100 to 250 man-month , large = 250 to 500 man-month, very large = more than 500 man-month. Other properties are defined in a similar manner.

Certainly it is possible to find additional properties that define a project. However, we tried to select properties that could be easily identified for most projects and are important for the decision process. The number and the contents of the properties proved sufficient during the use of the model in most cases.

Table 1. Property groups and properties of project

Property groups	Properties
Size and complexity of the system	<p>System size (defined in man-month or alternatively in equivalent KLOC, number of use-cases, functional points, etc.)</p> <p>System complexity (defined on the basis of number of components and/or architectural layers, number of connections to other systems, etc.)</p> <p>Criticality of the system (defined on the basis of severity of consequences of the system malfunction, a more critical system has to be more reliable)</p> <p>System history (defined on the basis of existence of preceding systems and on how the preceding systems would be used for creation of the new system)</p> <p>Planned future (defined on the basis of predicted future of the system, system can be only temporary and only maintenance is predicted or it can form a foundation for further upgrades and development of new systems even on different platforms, will the number of users of the system raise in the future, etc.)</p>
Project type	<p>Project priorities (defined as two main different priorities: productivity, where the most important aspect is that workable system is developed in short time; and legal liability where it is also important that artefacts are traceable and work on project is documented and tracked)</p> <p>Project predictability (defined by stability of requirements and maturity of business– projects with more stable requirements and more mature business can be better predicted)</p> <p>Time limitations (defines whether the project has very strict</p>

Property groups	Properties
System type	<p>time limitations that cannot be postponed)</p> <p>Cost limitations (defines whether the project has very strict cost limitations that cannot be changed)</p> <p>Types of target applications (defines common types of applications developed as the result of a project; these include traditional rich-client desktop applications, web-based applications and mobile applications)</p> <p>Architectures (defines types of architectures used in the project; these include service-oriented architectures & patterns, common object oriented architectures & patterns and structured architectures & patterns)</p> <p>DBMS type (defines types of DBMS used in the project; these include relational DBMS, hierarchical DBMS and object-oriented DBMS)</p> <p>Connectivity (defines the types of communication and connections to external systems used by the project)</p> <p>Legacy support (defines whether there is need for support of legacy technologies which require special approaches and what kind of support is needed)</p>
Development team and environment	<p>Experience in development (defines the level of developers' experience in the field of IS development)</p> <p>Experience in ISDM use (defines the level of developers' experience in the field of ISDM)</p> <p>Problem domain experience (defines the level of developers' experience in the field of system's problem domain)</p> <p>Willingness to learn (defines the level of developers' preparedness to learn to apply new approaches and technologies)</p> <p>Cooperation (defines the level of developers' preparedness to cooperate and share knowledge and experience with their colleagues)</p> <p>Discipline (defines the level of developers' discipline in following the rules and guidelines prescribed by an organisation)</p> <p>Team culture (defines the general team culture that can be autocratic/centralised or democratic/participative)</p> <p>Team location (defines physical location of the development them that can be centralised in one building or dispersed over different continents)</p>

Property groups	Properties
	<p>Team size (defines the size of the development team on a project or typical size of development team in an organisation)</p> <p>Number of development organisations involved in the project (defines the number of different development organisations that cooperate on the same project)</p>
Customer	<p>Requirements regarding ISDM and documentation (defines how rigorous are the customer's requirements regarding ISDM and the documentation produced)</p> <p>Cooperativeness (defines the level of customer's willingness to cooperate with developers especially in specifying the system)</p> <p>Problem domain knowledge (defines the level of customer's knowledge in the target problem domain)</p>

Different projects often stress different aspects of the development. For instance, property called *Legacy support* might play an important role in selection of the right ISDM for a project dealing with upgrade of a legacy system, but is significantly less important on projects that build new IS from scratch, though legacy support might be still desired. Therefore a weight (see Figure 1) can be assigned to each property that enables the user of the decision model to put emphasis on properties that are more important for specific project.

3.2. ISDM properties

We propose a set of ISDM properties shown in Table 2. The properties are grounded on literature discussed in section 2 and our experience gained from application of the decision model in practice (see section 5). We propose to organise the properties in seven main property groups that are further divided into properties for which the predefined values are assigned.

The ISDM descriptions are typically maintained by an ISDM expert and not by the business user of the decision model. The ISDM expert describes an ISDM by assigning the predefined values to the ISDM properties. It is possible that an ISDM has special properties and values that are not predefined, but are important for the decision process. In such case, the ISDM expert can add these new properties and values and also define additional decision rules that are in connection with these new properties and values. However, practical application of the decision model showed that the number and content of our proposed ISDM properties is sufficient for selection of ISDM in most cases.

Table 2. ISDM properties

Property groups	Properties
Process – general properties	<p>Process lifecycles (defines process which lifecycles are supported by an ISDM like waterfall, iterative, incremental etc.)</p> <p>Time distribution (defines how the development time is distributed during process; front-loaded ISDM focus on analysis and design while back loaded ISDM focus on implementation and testing)</p> <p>Development perspective (defines the primary development perspective of an ISDM that can be top-to-bottom or bottom-up)</p> <p>Prototype support (defines whether an ISDM explicitly supports use of prototypes)</p> <p>Artefact traceability (defines whether and how well artefact traceability is assured by an ISDM)</p>
Process – primary disciplines scope and detail	<p>Requirements acquisition (defines whether and in how much detail requirements acquisition is described by an ISDM)</p> <p>Analysis (defines whether and in how much detail analysis is described by an ISDM)</p> <p>Design (defines whether and in how much detail design is described by an ISDM)</p> <p>Implementation and integration (defines whether and in how much detail implementation and integration is described by an ISDM)</p> <p>Testing (defines whether and in how much detail testing is described by an ISDM)</p> <p>Deployment (defines whether and in how much detail deployment is described by an ISDM)</p> <p>Maintenance (defines whether and in how much detail maintenance is described by an ISDM)</p>
Process – supportive disciplines scope and detail	<p>Project management discipline (defines whether and in how much detail project management is described by an ISDM)</p> <p>Management of development environment (defines whether and in how much detail management of development environment is described by an ISDM)</p> <p>Configuration management (defines whether and in how much detail configuration management is described by an ISDM)</p> <p>Change management (defines whether and in how much detail change management is described by an ISDM)</p>

Selecting a methodology for business information systems development: decision model and tool support

Property groups	Properties
Techniques and methods	Modelling techniques and notation (defines which modelling techniques are recommended or prescribed by an ISDM to produce different ISDM artefacts) People and project management techniques (defines which techniques related to people and project management are recommended or prescribed by an ISDM to manage the development team)
Tools and development environment	Development languages (defines the development languages that recommended or suitable for use with an ISDM) Supportive tools (defines whether tools are available that automate certain parts of an ISDM's disciplines) Development frameworks and environments (defines the types of frameworks and development environments that are supported by an ISDM)
Type of development	Types of target applications (defines common types of applications that are supported and are normally developed by an ISDM) Architectures (defines types of architectures that are supported and are normally used by an ISDM) DBMS type (defines types of DBMS that are supported and are normally used by an ISDM) Connectivity (defines the types of communication and connections to external systems that are supported and are normally used by an ISDM) Legacy support (defines whether an ISDM includes support for legacy technologies and what kinds of technologies are supported)
Support and learning	Consistency of concepts throughout ISDM (defines the level of consistency of concepts used in different disciplines and other parts of an ISDM) Consistency of concepts use for a single role (defines the level of consistency of concepts used in different disciplines and other parts of an ISDM that are executed by a single role) Consistency of concepts in supportive tools (defines the level of consistency of concepts used in ISDM with concepts used in tools) Available support (defines what kind of support is available for ISDM)

Property groups	Properties
Available training for different roles (defines what kind of training is available for ISDM for different roles)	

The descriptions of ISDMs are needed in cases when the business user requires concrete names of ISDMs that are suitable for his IS development project/organisation. In cases when the business user only needs information about ISDM properties that are suitable for his project, but does not require a concrete ISDM name, the ISDM descriptions are not used (further discussed in subsection 3.4).

3.3. Decision rules

The decision rules form the backbone of the decision model. Our intention was to develop a set of rules that facilitate selection of ISDM using a computer supported decision model. To define the content of the rules we focused our efforts on collection and composition of existing rules that can be extracted from guidelines and models presented in literature discussed in section 2. It is important that these rules come from practice and have already been tested in real life environment. We transformed these rules from natural language representation found in the literature to format required by the decision model. We avoided changing the content and meaning of the decision rules during the transformation, however minor adaptations and generalizations of some of the rules were required so that they could be used in a computer supported decision model and that they could serve as common guidelines for ISDM selection.

We propose to organise the decision rules in a form of a two-dimensional matrix depicted in Figure 2. The header column and the header row of the matrix contain all ISDM properties and project properties correspondingly. Each property has all of its possible values enlisted. In this manner a matrix containing decision rules for each combination of an ISDM property and a project property is formed. A decision rule for a combination of two properties is written in a sub-matrix containing evaluations for each combination of values for the two properties. There are five evaluations that can be assigned to a certain combination of ISDM property value and project property value: very suitable (2), suitable (1), neutral (0), unsuitable (-1), very unsuitable (-2).

Selecting a methodology for business information systems development: decision model and tool support

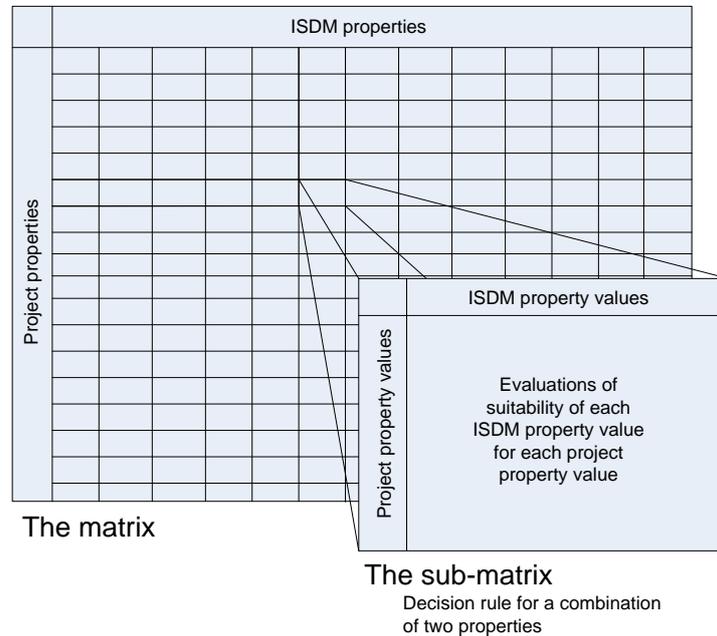


Fig. 2. The matrix containing decision rules

In the following example we demonstrate the structure and function of the decision rule sub-matrix. Table 3 shows an example of such sub-matrix. For the purpose of demonstration we focus only on one sub-matrix i.e. combination of one ISDM and one project property. *Implementation and integration* is one of ISDM's properties (scope of the ISDM). This property can take three different values: *ISDM contains no description of implementation and integration*, *ISDM covers implementation and integration in moderate detail*, and *ISDM covers implementation and integration in detail*. These values form the first of the two dimensions of a decision rule sub-matrix. *Criticality of a system to be developed* is one of project's properties that can take four different values: *system failure can result in a loss of comfort*, *system failure can result in a loss of discretionary money*, *system failure can result in a loss of essential money*, and *system failure can result in a loss of life*. These values form the second dimension of the decision rule sub-matrix. Three different values on the first dimension and four different values on the second dimension form a sub-matrix with twelve possible evaluations, one evaluation for each combination of the two dimensions' values. Based on literature [8] we defined the following rule: "In case the ISDM does not describe *Implementation and integration* at all it is very unsuitable for development of a system, failure of which can result in a loss of life." So we evaluate this combination of values as very unsuitable (-2) in the sub-matrix. Similarly, rules for all other combinations of ISDM property value (*Implementation and integration*) and project property value (*Criticality of a*

system) are defined. Certainly, in some cases the evaluations presented in Table 3 could differ slightly. However, in our experience moderate alterations of some evaluations typically do not affect the final result significantly as there are many other properties that are also considered during evaluation.

Table 3. An example of the decision rule sub-matrix

		Implementation and integration		
		no description	moderate detail	in detail
Criticality of a system to be developed	loss of comfort	0	2	0
	loss of discretionary money	-1	1	1
	loss of essential money	-2	1	1
	loss of life	-2	-1	2

The matrix is designed in a way that it can be expanded. However, adding a new rule is not a trivial task as each new rule requires testing. In our experience, the predefined rules suffice for most of typical cases. Expansion of the rules might be needed in special cases only i.e. to suite special types of organisations and projects. It is important to consider that any new rule should be added by an expert that has profound knowledge in the field of ISDMs and that each new rule should be carefully tested before inclusion in the decision rules matrix.

Table 4. An example of a decision rule explanation in natural language for business user

Project property	ISDM property
Project priorities	Artefact traceability
<p>In traditional ISDM traceability is an important part of the process, however in agile ISDM traceability is not guaranteed [based on [18]]. In case that project priority is traceability, processes that assure traceability are favoured over the processes that do not.</p>	

Furthermore, each decision rule sub-matrix has a corresponding natural language description that is based on the original description of the decision rule found in literature. This description helps the business user to better understand the reasons for recommendations produced by the decision model. Table 4 shows an example of such natural language description of the

Selecting a methodology for business information systems development: decision model and tool support

decision rule that describes the relation between project property *Project priority* and ISDM property *Artefact traceability*.

3.4. Processing and explanation

In this subsection we discuss our proposed approach for processing the project/organisation descriptions, ISDM descriptions and the decision rules matrix discussed in the preceding subsections in order to produce the recommendations for the selection of an ISDM that suits the needs of an IS development project or an IS development organisation. As recommendation without explanation is rarely sufficient, we propose an approach that also explains why a certain ISDM is preferred over the other for a certain IS development project/organisation.

In cases when it is required to select the most suitable ISDM not only for a project, but for the whole organisation dealing with similar projects the recommendation is produced by using the same approach as for a single project except that average values that are typical for the organisation are used.

By using the proposed approach two main types of results can be produced: firstly, a recommendation of one or more concrete ISDM that suit the project/organisation and secondly, a list of generally recommended ISDM property values for the project/organisation. The first type of result is useful especially in cases when there is a tension to use a well-known predefined ISDMs and a comparison of these concrete ISDMs is required. The second type of result gives a general advice on property values that an ISDM should have for an IS development project/organisation. This type of result can be used either during selection of a concrete ISDM or for development of a new ISDM or improvement of an existing ISDM.

The first type of result is presented as a two dimensional table where heading row contains the names of considered ISDMs, heading column contains names of considered projects and the content of the table are the results of the evaluation for each combination of ISDM and project. We propose the following formula to compute this type of result:

$$\text{Score}(\text{ISDM}_x, \text{Project}_y) = \sum_{a \in A} \sum_{b \in B} (\text{IPT}(a, \text{ISDM}_x) \times (\text{PPT}(b, \text{Project}_y) \times \text{PWT}(b)) \times \text{DRM}(a, b)) \quad (1)$$

where

$A = \{\forall a: a = \text{Characteristic}_{\text{ISDM}}\}$,

$B = \{\forall b: b = \text{Characteristic}_{\text{Project}}\}$,

ISDM_x = an instance of ISDM,

Project_y = an instance of project/organisation,

$\text{IPT}(a, b)$ is a cell in column a and row b of ISDM properties table,

$\text{PPT}(a, b)$ is a cell in column a and row b of Project properties table,

$\text{PWT}(b)$ is a record in row b of Project properties weight table,

$DRM(a,b)$ is a cell in column a and row b of Decision rules matrix.

The second type of result is also presented in a two dimensional table where heading row contains the names of all ISDM properties and heading column contains the names of all projects. The content of the table shows the most suitable values of ISDM properties for each project, regardless of whether a concrete ISDM having all these properties exists or not. This type of result is therefore especially useful when the user is planning to develop a new custom ISDM that can actually possess most of such recommended properties. We propose the following formula to compute the recommended value for $ISDMProperty_x$ and $Project_y$:

$$RValue(ISDMProperty_x, Project_y) = DRM(a', r) . \quad (2)$$

where

$ISDMProperty_x$ = the ISDM property for which we want to find the suitable value,

$Project_y$ = an instance of a project/organisation,

r = the number of the row in DRM table that contains the names of property values,

a' = the number of the column in DRM table that contains the name of the most suitable property and is computed by using the following formula:

$$a' = \arg \max_{a \in A_{ISDMProperty_x}} (PVScore(a, b)) . \quad (3)$$

where

$$PVScore(a, b) = \sum_{b \in B} ((PPT(b, Project_y) \times PWT(b)) \times DRM(a, b)) . \quad (4)$$

and

$A_{ISDMProperty_x}$ is a subset of columns for $ISDMProperty_x$.

To explain and better understand the recommendations computed by the discussed formulas we propose to compute positive or negative contribution of each individual combination of ISDM property and project property for selected ISDM and project. Using such explanations it is possible to quickly detect the ISDM properties that are evaluated as the most unsuitable for an IS development project/organisation. We propose the following formula to calculate the score of individual combinations of properties:

$$PScore(ISDM_x, Project_y, i, p) = IPT(i, ISDM_x) \times (PPT(p, Project_y) \times PWT(p)) \times DRM(i, p) . \quad (5)$$

where

$ISDM_x$ = an instance of ISDM,
 $Project_y$ = an instance of project/organisation,
 i = the selected property of $ISDM_x$,
 p = the selected property of $Project_y$.

Complementary way of explaining the recommendations that is important especially for business users that are less experienced in the field of ISDM is using natural language descriptions of the decision rules discussed in section 3.3.

4. Implementation of the decision model

Processing of large number of decision rules to produce a recommendation is a complex and time consuming task that cannot be done without appropriate tool support. Therefore, for testing of the decision model we had to use tool support already in the beginning of the research. We developed a prototype tool using MS Excel that enabled us to experiment with different properties and to fine tune the decision rules. This prototype was also used in real projects to facilitate selection of the suitable ISDM. The early version of prototype is presented in [15, 38].

However, although MS Excel is a versatile environment in which we were able to experiment with the decision model quite easily, practical tests showed that this environment has some important drawbacks. Firstly, it was difficult to create a user friendly interface in this environment. This hindered direct use of the prototype outside of the research group. Consequentially, only the results of an evaluation were presented to the companies' experts, but they were unable to test the prototype directly. We did not see this as an important obstacle at the first stage of testing, but later when we tried to enable the companies' experts to experiment with the tool directly this presented a significant difficulty. Secondly, when the number of rules and properties increased the maintenance of the Excel based prototype was becoming more and more awkward. Thirdly, we sought to make the decision support tool accessible through Web so that users could easily access the model without having to install the tool in their computers. However, the model in MS Excel format was not convenient for use in Web environment.

Consequentially, it was necessary to redevelop the tool which implements the proposed decision model. We developed new tool using MS Visual Studio [15]. The tool is designed as a web application that is divided onto the four modules. Figure 3 shows a use-case diagram depicting the main functionality of the tool. The tool is used by two different actors – a business user and a ISDMs expert. The tool allows the business user to maintain projects and their properties, to run analyses and receive the recommendations, and to

explore these recommendations. The ISDMs expert can maintain ISDMs and their properties, add new ISDMs, and modify set of decision rules.

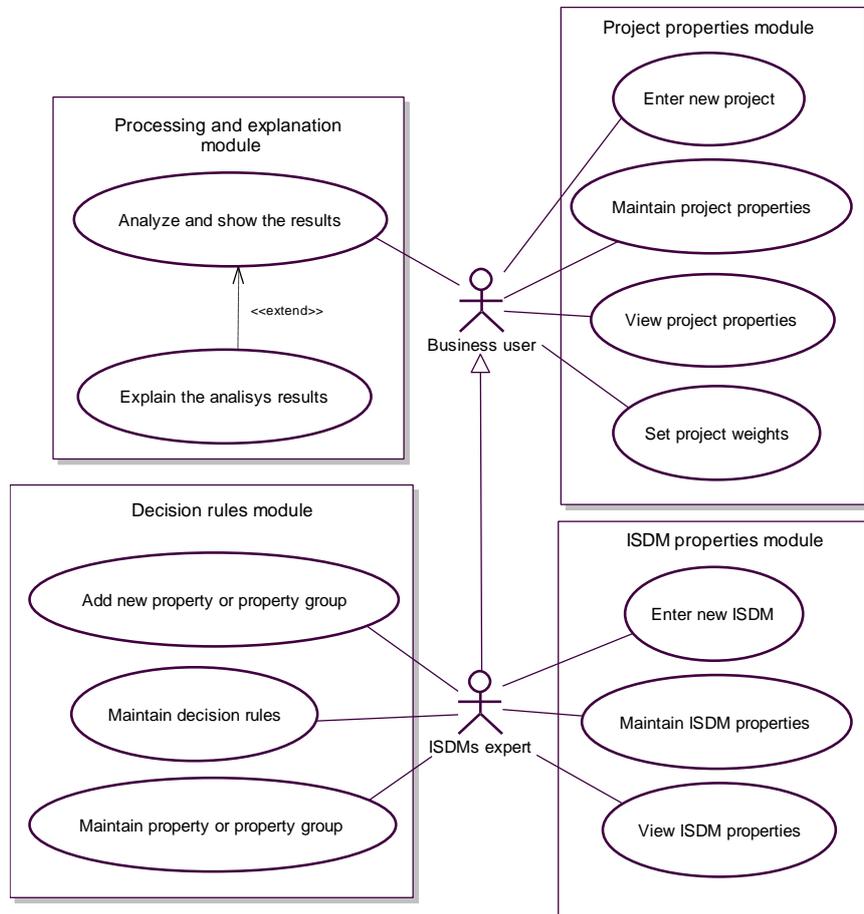


Fig. 3. Use-case diagram depicting the main functions of the tool

The tool uses a relational database to store the data that is the main integration point of the four modules. The E-R diagram in Figure 4 shows the structure of the database. Both ISDM and project properties are stored in the entity type called *Property*. More *Property* values can be assigned to each *Property* instance. The *Decision rule* entity type contains evaluations for each combination of ISDM and project property value thus forming the decision rules matrix. *Decision rule description* contains textual descriptions of rules stored in *Decision rule* entity type and describes one decision rule sub-matrix. The database supports more *Decision rule sets* so that the user can experiment with variations of decision rules. This is especially useful in cases when certain decision rules apply only for certain type of development (e.g.

development of web applications). *Project* and *ISDM* are defined by their *property values* and together form a decision *Case*. The *Case* entity is used to select a subset of *ISDM* and/or *projects* which are considered during an evaluation. Each *Property* of *Property group* can be assigned *Property weight* and *Property Group Weight* correspondingly. By using weights the user is able to emphasize the properties that are more important in his case. To ease the experimentation the database supports more *Weight sets*.

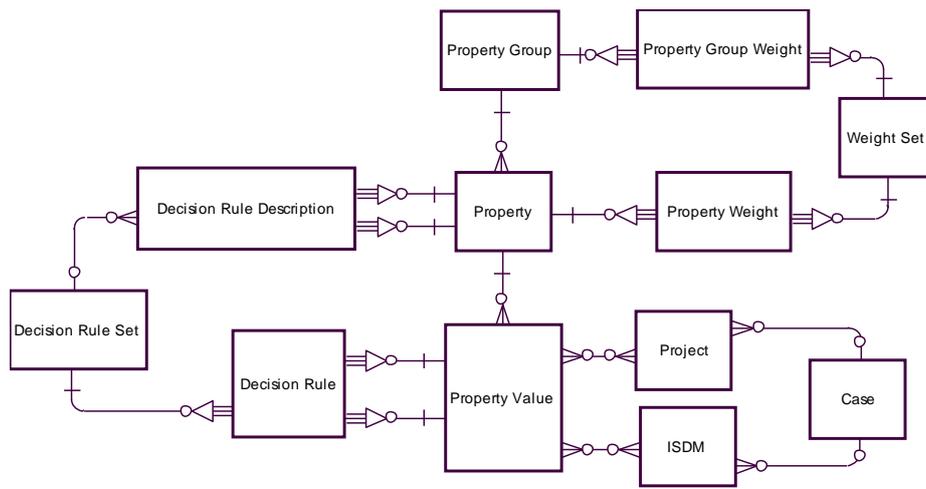


Fig. 4. ER model of the tool's database

Although the project properties module and the ISDM properties module are used to describe two different types of properties – ISDM or project, and are used by two different types of users – business user or ISDM expert, they have quite similar functionality. In both cases the user describes the project or the ISDM by setting the values of the predefined properties. In cases when the user lacks certain information he can choose undefined value. This means that the property will not be considered during evaluation. An addition to project properties module is an interface that allows user to modify weights for different properties of his project. This way the user can define which aspects of development are more important for his case. Figure 5 depicts the main form of project properties module.

User: guest

[Back](#)

Project properties

Project name:

Properties group:

- Size and complexity of the system
- Project type
- System type
- Development team and environment
- Customer

Properties in the selected group:

Property name	Property description	Value
System size	What is the size of the system, considering various factors like number of use-cases, KLOC, functional points, etc.	Small
Complexity of the system	What is complexity of the system considering complexity of architecture and IT, number of connections to other systems, etc.	Medium
Criticality of the system	How important is it that the system functions correctly? What are the consequences of the system malfunction?	Low
System history	Is the project about an upgrade of a legacy system, a completely new system etc.	Upgrade of an existing sys
Planned future	What kind of upgrades and expansions are expected? Is this only a temporary system that will be soon replaced by another one? Etc.	Minor upgrades

Fig. 5. Screenshot of the main form in Project properties module

Decision rules module is accessible only for an ISDM expert. The primary task of the module is to modify decision rules matrix. The module can store more rule sets which enables the expert to create rule sets for different situations. Besides modifying the rules matrix, the module also allows an expert to add properties and property values. Adding new property or property value is a relatively complex task because the expert has to create a large number of decision rules for each new property. Figure 6 shows the main form of decision rules module that enables an ISDM expert to redefine existing decision rules or to define new decision rules for the chosen decision rule sub-matrix (discussed in section 3.3). Figure 6 shows the decision rule for a combination of project/organisation property *System size* and ISDM property *Requirements acquisition*. In this case one property can take five different values and another can take three different values thus forming a matrix with 15 possible evaluations. As number of different values that can be taken by different properties can vary the form is dynamically generated to show different decision rules. In case that an additional value is added to an existing ISDM property additional rules have to be defined that map this additional value to values of project/organisation properties. In such case the

Selecting a methodology for business information systems development: decision model and tool support

form shows existing evaluations that form the decision rule and also enables a user to enter the missing evaluations to complete the rule. Below the decision the form shows textual description of the rule that explains the rule and also references the rule to existing research. In case shown in Figure 6 the rule is based on [2] and [8].

	Very small	Small	Medium	Large	Very large
No description	0	-1	-2	-2	-2
Light weight description	2	2	1	-1	-1
Detailed description	-2	-1	1	2	2

Textual description of the decision rule:
Larger systems require more formal approaches that help manage the complexity of the requirements acquisition of such systems. On the contrary, for smaller systems are best suited less defined approaches that do not prescribe procedures and documentation that are not needed for the development of such systems. [based on Cockburn2002, Ambler2002]

Possible evaluations in the decision rule: (-2) highly incompatible values (-1) incompatible values (0) neutral (1) compatible values (2) highly compatible values

Fig. 6. Screenshot of the main form in Decision rules module

Processing and explanation module produces the recommendation report. The report shows the results of the comparison of different ISDM and their appropriateness for different projects and the most suitable ISDM properties for the projects. The module can also produce explanation report for a selected combination of ISDM and project property as described in section 3.4.

5. Verification of the decision model in practice – case studies

So far the decision model has been applied in five cases i.e. three organisations dealing in IS development and two IS development projects.

The recommendations produced by the decision model were used in the two ways: as input in the process of selection of a suitable ISDM for a new project and as input in the process of improvement of existing organisation's ISDM. These practical applications of the decision model enabled us to verify our proposed two main hypotheses about the decision model.

The first hypothesis was that ISDM experts agree with and approve the recommendations for ISDM selection produced by using the decision model. We asked four ISDM experts to examine the recommendations produced by using the decision model on the five cases. ISDM experts agreed with and approved the recommendations for all five cases, which was not surprising as the decision model's rules are based on generally acknowledged guidelines for ISDM selection available in existing literature discussed in section 2.

The second hypothesis was that the recommendations produced by using the model contain nontrivial information that can be efficiently used to select a suitable ISDM for an organisation or project. We tested this hypothesis in each of the five cases by discussing the results with business users of the decision model i.e. organisation's employees or project members responsible for management of their ISDM who used the decision model. We were especially interested in their opinion whether the model significantly contributed to their understanding of the problem and their decision. On the whole we discussed the results with fourteen business users in the five cases. Six of these users changed their initial decision after using the decision model. Three of the users did not have any initial decision and the model helped them to reach the decision. The initial decision of five users was the same as the model's recommendation. All fourteen users reported that the model helped them to understand the problem of ISDM selection better. They also indicated that they felt more confident in their decision after using the model.

In the following subsections we present two cases in detail. They show how the model contributed to the decision making process during ISDM selection or improvement. In case A the model was used to select a suitable ISDM for a new IS development project and in case B the model was used to improve existing IS development organisations' ISDMs.

5.1. Case A – selecting a suitable ISDM for a new project

Case A was a project where two development teams cooperated to develop and deploy a part of IS for a government institution. The teams came from two IS development companies. Throughout the project the first team that was responsible for the development of the central part of the IS was expected to have from 9 to 12 members and the second team that was responsible for the development of a subsystem was expected to have 5 members. Even though both companies used certain internal standards and frameworks, and had some experience with iterative project management they did not follow any formal ISDM. On both companies' preceding projects the development process was typically organised by a project manager in an

ad hoc manner. However, in this case the government institution required the use of a formally defined IS development documentation and procedures. Although, the institution did not prescribe the use of a particular ISDM, it required that the companies define which documentation would be produced, which activities and in what order would be followed, how the progress of the project would be monitored, etc. Therefore the companies decided that for the needs of the project they would use one of publicly or commercially available ISDM that they would partially adapt to their needs.

The selection of ISDM was difficult as the companies did not have much experience in the field of ISDM. To ease the selection process and to limit the number of ISDM candidates they used the decision model proposed in this paper. The decision model was used by the project manager and some of the team members of the both teams. Initially they needed our help to correctly understand and set the necessary parameters. After a short introduction they were able to use the decision model independently and make experiments with different settings. This experimentation was also one of the main advantages of using the decision model as they could get better understanding of appropriateness of various ISDM.

Eight different ISDM were considered: Extreme programming (XP), Scrum [30], Rational Unified Process for small projects (RUP for small projects), Rational Unified Process (RUP), Oracle Custom Development Method (CDM), Unified Methodology of IS development (UMISD - used in government institutions), Rapid Application Development (RAD) [25] and a custom ISDM that was developed for the needs of one of the two companies some time ago, but was never used in practice. Table 5 shows the input values used in the evaluation.

Table 5. The input values used in the evaluation

Size and complexity of the system System size = small System complexity = medium Criticality of the system = medium System history = new system Planned future = maintenance and minor upgrades	Development team and environment Experience in development = high Experience in ISDM use = low Problem domain experience = medium Willingness to learn = medium Cooperation = medium Discipline = medium Team culture = open & participative Team location = centralized Team size = small to medium Number of devel. org. involved in the project = 2-3
Project type Project priorities = traceability Project predictability = predictable Time limitations = strict Cost limitations = sufficient	
System type	

Types of client applications = web Architecture = three-tier OO DBMS type = relational Connectivity = present IT Legacy support = not needed	Customer Requirements regarding ISDM and documentation = high Cooperativeness = medium Problem domain knowledge = medium
--	--

The recommendation produced by the decision model was RUP for small projects (see Figure 7). This result was somewhat unexpected by the project management that favoured XP at the time. Therefore, the explanation of the recommendation that presented weaknesses and advantages of each ISDM was even more important. The explanation exposed two major weaknesses of using XP in the project. Firstly in case of using XP, there was lack of formal documentation that would be shared among the two teams and presented to the customer, and secondly, there was lack of traceability which was a project priority. Even though XP had an advantage in time to learn, RUP for small projects was considered a better choice as it offered formal documentation and facilitated traceability.

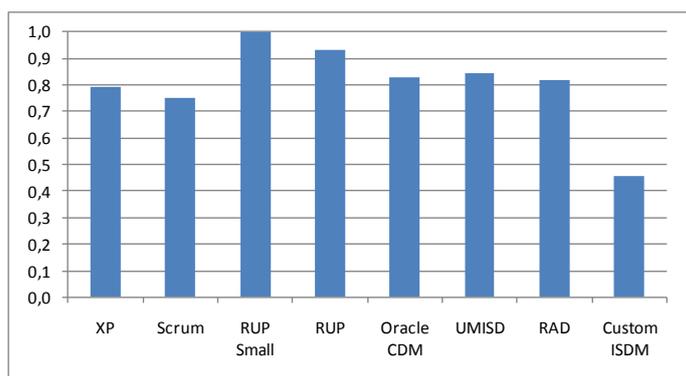


Fig. 7. Normalized evaluation scores for all eight ISDM

After examining the explanation of the results the project management agreed that the RUP for small projects was the better choice for their case. They acknowledged that due to their low experience in the field of ISDM they neglected some of the important aspects of ISDMs and noted that the model helped them to get a better understanding of ISDM field in a relatively short time. The case and the recommendations of the proposed decision model were additionally examined by external ISDM experts who agreed with recommendations produced by the decision model.

A light configuration of RUP for small project was actually used on the project. It enabled the companies to produce the required documentation, assure traceability and successfully complete the project. The decision to use RUP for small projects was actually seen as a compromise between the

Selecting a methodology for business information systems development: decision model and tool support

customer's requirements for formal documentation and traceability, and companies' existing development experience and initial tendency to use XP.

5.2. Case B – improving existing organisation's ISDM

Case B was an organisation dealing in IS development. Its main field of operation was development of pre-packaged business solutions. The work was organised in four teams each working in one product or project. The typical size of their team that worked on one product was 7 to 10 members, though occasionally they worked on custom projects that involved fewer members. The leader of each team was responsible for organisation of the team's development process. Consequentially, different informal development approaches were used in each team. The management planned to partially formalize the existing informal development process and upgrade it with selected parts of publicly or commercially available ISDMs. The main goal was to standardize their development processes which should enable: employees to easily switch between teams and simultaneously work on more products or projects, easier employment of new employees, better cooperation with other organisations, etc.

The decision model was used in two main ways. Firstly, it was used to help produce a list of recommended ISDM properties for each development team, and secondly, it was used to establish a set of suitable ISDM that could serve as a reference for improvement and standardization of the development process. Six different ISDM were considered: Scrum, Feature driven development (FDD) [29], Extreme programming (XP), Dynamic Systems Development Method (DSDM) [35], RUP and RUP for small projects.

A list of suitable ISDM properties for each of the four development teams was created in cooperation with each team leader. The leaders were encouraged to experiment with the decision model to get better understanding of the ISDM field and the needs of their team. Our initial evaluation showed that all four teams share similar characteristics. Not surprisingly, the recommendations produced by the model for the three of the four teams were quite similar (see Figure 8). In all three cases less rigorous approaches that follow agile principles were recommended and Scrum and XP were found to be the most suitable ISDMs. However, there were important differences in the recommendations for the remaining team for which more rigorous properties were recommended and RUP for small projects was proposed as the most suitable ISDM. After performing detailed analysis of the results and discussion with the team leaders, we discovered that the leader of the fourth team had different expectations of ISDM than leaders of the other three teams. The first three leaders put emphasis on simplicity of ISDM as they believed that learning and adoption of ISDM were the largest obstacles. The opinion of the fourth leader was that learning ISDM is not so difficult, so he did not see this as an important obstacle. His expectations focused mainly on establishing traceability and formality for his projects. Consequentially, the model's recommendation was a more rigorous ISDM.

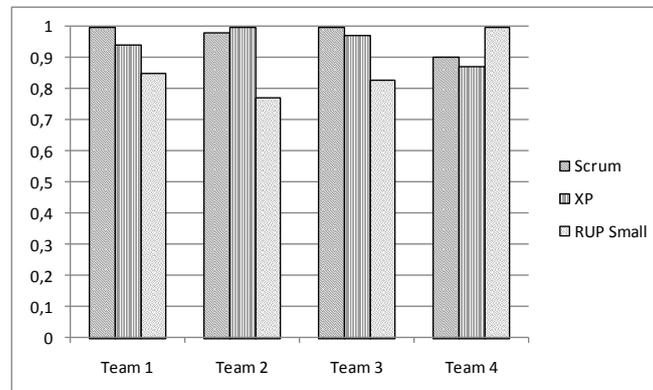


Fig. 8. Normalized evaluation scores for Scrum, XP and RUP for Small projects for each of the four development teams

To reach the final decision the advantages and disadvantages of the three recommended ISDM were presented to the leaders. They agreed to use Scrum as a reference ISDM. They would formalize and adapt their development process to follow basic principles of Scrum, however on projects requiring more formality they would add certain RUP's artefacts to Scrum's Product backlog [30] to assure better traceability.

All four project leaders found the decision model to be a useful tool that enabled them to gain understanding of appropriateness of various ISDM faster. They stressed the importance of possibility of experimentation with the model and confirmed that information offered by the decision model was relevant. The recommendations were also reviewed by external experts who validated that they are suitable for the needs of the organisation.

6. Conclusion

The analysis of related literature (e.g. [12]) and our experience shows that organisations and IT departments dealing with development of computer-based business IS often lack in-depth knowledge of ISDM field. This hampers their selection of ISDM and often limits their choice to ISDM vendors who favour their own ISDM. To improve this situation and help organisations to actively participate in the process of ISDM selection, we propose a decision model and a tool based on the decision model for help in ISDM selection.

The decision model and the tool have been applied in several practical cases confirming that they can be efficiently used to improve the ISDM selection process and help make better decision. The decision model does not substitute ISDM experts, but is merely a tool that on the one hand helps people responsible for ISDM selection to make a more informed decision and on other hand helps an ISDM expert to obtain better understanding of the

Selecting a methodology for business information systems development: decision model and tool support

needs of an IS development organisation. An important precondition for successful application of the decision model is that users of the decision model are cooperative and prepared to learn through experimentation with the decision model.

Our further work will focus on extension of the decision model's characteristics and rules. We intend to focus especially on ISDM that include explicit support for service oriented architecture (e.g. [13, 23]), service oriented frameworks (e.g. [36]) and approaches that support model driven architecture (e.g. [20]). Furthermore, we intend to examine possibilities to apply the decision model in the field of specialized methodologies like methodologies for building ontology (e.g. [40]), methodologies dealing in enterprise architecture (e.g. [39]), development approaches based on business rules perspective (e.g. [7, 19, 33]) and other alternative approaches and studies on existing ISDM (e.g. [22, 32]).

References

1. Ahituv, N., Neumann, S., Zviran, M.: A system development methodology for ERP systems. *The Journal of Computer Information Systems*, Vol. 43, No. 3, 56-67. (2002)
2. Ambler, S.W., Jeffries, R.: *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process* John Wiley & Sons. (2002)
3. Avison, D.E., Fitzgerald, G.: *Information systems development : methodologies, techniques and tools*. 3rd ed. McGraw-Hill, London, xv, 592 p. (2003)
4. Bajec, M., Vavpotic, D., Krisper, M.: Practice-driven approach for creating project-specific software development methods. *Information and Software Technology*, Vol. 49, No. 4, 345-365. (2007)
5. Beck, K., Andres, C.: *Extreme Programming Explained: Embrace Change*. 2 ed. Addison-Wesley Professional, 224. (2004)
6. Brinkkemper, S., Lyytinen, K., Welke, R.J.: *Method Engineering – Principles of method construction and tool support*. IFIP TC8. Chapman & Hall, Atlanta, USA. (1996)
7. Ceponiene, L., Nemuraite, L., Vedrickas, G.: Separation of event and constraint rules in UML&OCL models of service oriented information systems. *Information Technology and Control*, Vol. 38, No. 1, 29-37. (2009)
8. Cockburn, A.: *Agile software development*. Agile software development series. Addison-Wesley, Boston, 304. (2002)
9. Euromethod: Welcome to Euromethod. (2011) [Online]. Available: <http://projekte.fast.de/Euromethod/> (current 2011 28th of June)
10. Fitzgerald, B.: An empirical investigation into the adoption of systems development methodologies. *Information & Management*, Vol. 34, No. 6, 317-328. (1998)
11. Highsmith, J.: *Adaptive software development: A Collaborative Approach to Managing Complex Systems*. Dorset House Publishing, 392. (2000)
12. Highsmith, J.: *Agile Software Development Ecosystems*. Addison Wesley, 448. (2002)
13. IBM: *IBM RUP for Service-Oriented Modeling and Architecture v 2.4* (IBM Rational Method Composer Method plug-in). IBM Corp. (2006)

14. IBM: Rational Unified Process v 7.0.1 (IBM Rational Method Composer plugin). IBM Corp. (2006)
15. Infolab: ISDM Selection Tool Web Page. (2009) [Online]. Available: <https://infolab.fri.uni-lj.si/ISDMSelection> (current 2010 12/2)
16. ISACA: COBIT Framework for IT Governance and Control. (2011) [Online]. Available: <http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx> (current 2011 28th of June)
17. ISPL: Information Services Procurment Library. (2011) [Online]. Available: <http://projekte.fast.de/ISPL/> (current 2011 28th of June)
18. Jacobsson, M.: Implementing Traceability In Agile Software Development, in Department of Computer Science. Faculty of Engineering, LTH: Lund, Sweden. (2009)
19. Kalibatiene, D., Vasilecas, O.: Ontology Axioms for the Implementation of Business Rules. *Technological and Economic Development of Economy*, Vol. 16, No. 3, 471–486. (2010)
20. Karsai, G., Neema, S., Sharp, D.: Model-driven architecture for embedded software: A synopsis and an example. *Science of Computer Programming*, Vol. 73, No. 1, 26-38. (2008)
21. Kitchenham, B., Linkman, S., Law, D.: DESMET: a methodology for evaluating software engineering methods and tools. *Computing and control engineering journal*, Vol. 8, No. 3, 120-126. (1997)
22. 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)
23. López-Sanz, M., et al.: Modelling of Service-Oriented Architectures with UML. *Electronic Notes in Theoretical Computer Science*, Vol. 194, No. 4, 23-37. (2008)
24. Martin, J.: *Information Engineering, Book I: Introduction*. Prentice Hall. (1989)
25. McConnell, S.: *Rapid development : taming wild software schedules*. Microsoft Press, Redmond, Wash., xix, 647 p. (1996)
26. Mikulenas, G., Butleris, R.: An approach for constructing evaluation model of suitability assessment of agile methods using analytic hierarchy process. *Electronics and Electrical Engineering*, Vol. 10, No. 106, 99-104. (2010)
27. Mumford, E.: *Effective Requirements Analysis and System Design: The ETHICS Method*. Macmillan, Basingstoke, UK. (1995)
28. OGC: Welcome to the Official ITIL Website. (2011) [Online]. Available: <http://www.itil-officialsite.com/> (current 2011 28th of June)
29. Palmer, S.R., Felsing, J.M.: *A practical guide to feature-driven development*. Prentice Hall. (2002)
30. Rising, L., Janoff, N.S.: The Scrum software development process for small teams. *IEEE Software*, Vol. 17, No. 4, 26-32. (2000)
31. Saaty, T.L.: *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. RWS Publications, Pittsburgh, Pennsylvania. (2008)
32. Sánchez, P., et al.: Model-driven development for early aspects. *Information and Software Technology*, Vol. 52, No. 3, 249-273. (2010)
33. Smaizys, A., Vasilecas, O.: Business Rules Based Agile ERP Systems Development. *Informatica*, Vol. 20, No. 3, 439-460. (2009)
34. Stair M., R., Reynolds W., G.: *Principles of Information Systems, A Managerial Approach*. Eight Edition ed. Thomson course technology. (2008)
35. Stapleton, J.: *DSDM Dynamic Systems Development Method: The Method in Practice*. Addison Wesley. (1997)

Selecting a methodology for business information systems development: decision model and tool support

36. Šaša, A., Jurič, M.B.,Krisper, M.: Service-oriented framework for human task support and automation. IEEE transactions on industrial informatics, Vol. 4, No. 4, 292-302. (2008)
37. Triantaphyllou, E.: Multi-Criteria Decision Making Methods: A comparative Study. 1 ed. Applied Optimization. Vol. 44. Kluwer Academic Publishers, 320. (2000)
38. Vavpotic, D., Bajec, M.,Krisper, M.: Software development methodology evaluation model. In 12th International Conference on Information Systems Development. Kluwer Academic / Plenum Publishers, New York, Melbourne. (2004)
39. Wegmann, A.: On the Systemic Enterprise Architecture Methodology. In International Conference on Enterprise Information Systems (ICEIS). 483-490. (2003)
40. Wongthongtham, P., et al.: Ontology-based multi-site software development methodology and tools. Journal of Systems Architecture, Vol. 52, No., 640-653. (2006)
41. Yin, R.K.: Case study research : design and methods. 3rd ed. Applied social research methods series ; v. 5. Sage Publications, Thousand Oaks, Calif., 200. (2003)
42. Yoon, K.P., Hwang, C.-L.: Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences) Sage Publications, Inc, Thousand Oaks, London, New Delhi, 83. (1995)

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.

Damjan Vavpotič and Olegas Vasilecas

Olegas Vasilecas. Prof. habil. (hp) dr. Olegas Vasilecas is full time professor at the Information Systems Department, and head of Information Systems Research Laboratory in Vilnius Gediminas Technical University. He is author of more than 200 research papers and 5 books in the field of information systems development. His research interests: knowledge, including business rules and ontology, based information systems development. He delivered lectures in 7 European universities including London, Barcelona, Valencia, Athens and Ljubljana. O. Vasilecas carried out an apprenticeship in Germany, Holland, China, and last time in Latvia and Slovenia universities. He supervised 7 successfully defended doctoral theses and now is supervising 5 more doctoral students. He was leader of many international and local projects. Last time he leaded „Business Rules Solutions for Information Systems Development (VeTIS)“ project carried out under Lithuanian High Technology Development Program.

Received: March 15, 2011; Accepted: September 1, 2011.