Get Your Organization Ready for the Next Generation of Information Systems Development Methodologies

Fleming Woo
The Hong Kong Polytechnic University
Hong Kong

Romas Mikusauskas
London Metropolitan University
London, UK

Dean Bartlett
London Metropolitan University
London, UK

Rob Law
The Hong Kong Polytechnic University
Hong Kong

Summary
Organizations are generally expected to achieve significant benefits by selecting and adopting a robust information systems development methodology (ISDM). However, the study reported herein indicates that the extent of adoption of a methodology for IS development is quite inadequate. As different types of systems may require different types of tools, techniques, software processes and programming paradigms, no single set of solutions is best for all situations. When considering the adoption of a systems development methodology, management must take into account the industry, organization, projects, system products and individual developers involved. This paper provides an overview of the typical components of such a methodology, relates the components to the various stages of adoption, and examines the reasons for adopting or not adopting a methodology. Finally, based on the empirical findings, a framework is proposed to help management make informed decisions regarding the adoption and implementation of methodologies that can meet the specific IS development needs of their organization.
***Key words**: information systems development methodology, adoption of technologies, diffusion of innovations, methods and methodologies.*

1. Introduction

Many tools and methods have emerged in response to the huge demand for information systems development and the increased complexity of systems. Tools and techniques that are not effective have become obsolete, and subsequently been replaced. Those that have enduring value in systems development include: fourth-generation languages (4GL), computer-aided software engineering (CASE), structured development methodology (SDM), the Rational Unified Process (RUP), the Agile Development Platform (ADP) and relational database management systems (RDBMSs). Currently, the norm is to use object-oriented tools, techniques and processes, which include: object-oriented programming (OOP), object-oriented methodologies (OOMs), the Unified Modeling Language (UML), object-oriented database management systems (ODBMSs) and component-based development (CBD). New classes of methodologies have emerged in recent years, including software product lines (SPLs) and aspect-oriented software engineering (AOSE). Regarding social aspects, organizations have recently turned their attention to the Personal Software Process (PSP) and eXtreme Programming (XP). A study conducted by Johnson [14] revealed that both OO and non-OO software developers consider OOMs to be superior to traditional systems development methodologies. However, the existing literature reveals that the rate of adoption of innovative technologies in information systems development is slow [23], that studies of the evaluation and selection of technologies are few and that often, many claimed benefits cannot be realized [10].

For example, OOMs are intended to streamline processes, adjust the manageability and productivity of the development process, improve the quality of system deliverables and facilitate reuse. Other advantages of OOM include: software design becoming more traceable to system requirements, better transition between requirement analysis and design, more resilience to changes in software, the production of more reusable components and the delivery of more stable systems. However, many system developers do not view object orientation as the way to solve their problems, and opt to keep using structured development methodologies in their planning, requirement analysis and design of information systems. They also continue to use a relational rather than an object-oriented database management system [30]. For many years, the Java language, which is an object-oriented Web programming tool, has frequently been adopted because an increasing number of applications are Web based and deploy a graphical user interface.

2. Background

Fitzgerald [8] reviewed the many reasons for the use of a methodology, including its ability to decompose a complex process, visible steps to facilitate project management, provision of a structural framework for the

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

deployment of tools, techniques and resources, allowance for skill specialization and division of labor, contribution to the acquisition of domain knowledge, enhancement of communication and interoperation among developers and improvement in quality and productivity. He also identified barriers to the adoption of a methodology, including the great variety of methodologies, arbitrary deviation of one methodology from another, lack of an empirical foundation, inability to match the project type or meet project objectives in all situations, suppression of developer creativity, failure of many methodologies to meet the need for the customization of commercial packages, failure to consider out-source and contract development and complexity of methodologies, which makes them difficult to learn.

The existing frameworks to evaluate ISDMs focus on comparing and evaluating these methodologies from the perspective of processes [5], graphical notations [18] and structures, functions and behaviors [11]. Iivari [12] stated that there is a lack of frameworks to evaluate the work done in OOMs. Mrdalj [21] identified the need for the collection, classification and evaluation of the available materials. Ladden [17] analyzed the factors that should be considered in developing an OOM. Monarchi and Puhr [19] affirmed the need for an evaluative rather than a prescriptive model in the selection of OOMs. They evaluated 23 different methodologies based on the completeness and cohesiveness of the processes and representations, and identified their strengths and weaknesses. Surprisingly, research into methods for evaluating, selecting and adopting an ISDM has slowed in recent years.

3. Components of an ISDM

"Methodology" is a confusing term in the literature. Ovaska [22] surveyed a sample of six definitions of systems development methodology and found a wide range of meanings. Some authors use the terms "methodology" and "method" interchangeably, whereas others regard methodology as the study of methods [25, 31]. The increasing number of studies in this area has led to the recent emergence of a new field, method engineering [3].

Whitten and Bentley [27] stated that "a methodology in systems development is a process which is derived from the logical problem-solving process called a systems development life cycle" (p. 72), and that "a methodology is the physical implementation of the logical life cycle that incorporates (1) step-by-step activities for each phase, (2) individual and group roles to be played in each activity, (3) deliverables and quality standards for each activity, and (4) tools and techniques to be used for each activity" (p. 73). Whitten and Bentley [28] recently updated their definition, describing methodology as "a formalized approach to the systems development process; a standardized process that includes the activities, methods, best practices, deliverables, and automated tools to be used for information systems development" (p. 70).

In both the third and fourth editions of their popular textbook, Information Systems Development: Methodologies, Techniques, and Tools, Avison and Fitzgerald [1] (p. 21) and [2] (p. 24) defined methodology as follows.

> A methodology is a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system. A methodology will consist of phases that in turn consist of sub-phases, which will guide systems developers in their choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control, and evaluate information systems projects.

Vessey et al. [26] defined methodology as a systematic approach to building a system, with tools to facilitate the following of the steps and rules of the methodology. These definitions have many commonalities, based on which we identified the three major components of a systems development methodology: tools, techniques and processes, for the development of the framework proposed in this paper.

4. Stages of ISDM adoption

Early studies based on the classical theory of technology diffusion treat adoption as a discrete process, that is, either complete adoption or rejection of a technology [4]. However, as the tools, techniques and processes of systems development continue to improve, technology diffusion is now considered to be a continuous process. In addition, to succeed in the adoption of an innovative technology in systems development, an organizational strategy for providing constant support to the developer is as important as taking care in its selection. In his model of the innovation-decision process, Rogers [24] (pp. 169-192) developed a multi-stage process for individuals making the adoption decision: (i) knowledge, (ii) persuasion, (iii) decision, (iv) implementation and (v) confirmation stages. Rogers [24] (pp. 420-430) also advocated five stages for the implementation of the innovation process: (i) agenda setting, (ii) matching, (iii) redefining/restructuring, (iv) clarifying and (v) routinizing.

The present study reports an ISDM adoption approach in which organizations combine the decision and adoption processes. After the evaluation of innovative technologies, organizations often pilot a technology on a small scale before making a full organization-wide commitment to adopt the new technology.

5. Hypotheses of ISDM Adoption

Efficiency and effectiveness are major dimensions of an evaluation framework. A methodology is not efficient if it cannot enhance system development productivity and not effective if it is not adopted by organizations or cannot improve the quality and reusability of the system deliverables. Efficiency is the internal performance view of the system development process, and includes factors such as ease of use, repeatability, adaptability, integration with other tools, amount of uncertainty, responsiveness to change, accountability, traceability and manageability. In contrast, effectiveness is the external performance view, and involves the outcomes of the system development process. In addition to the quality of the resultant software, outcomes include functionality, correctness, robustness, reliability, extendibility and maintainability of the system product. Reusability is addressed in terms of, but not limited to, reuse objectives, impact on developer productivity and problems in reuse. A framework must consider all of these factors and present a consolidated view of them for management to make informed decisions.

Based on an analysis of the existing methodologies, models and frameworks of the tools, techniques, and processes of systems development along with their effects on the reusability of system deliverables, quality of software products and productivity by deploying system resources, a preliminary framework was derived with reference to Kaplan and Norton [16] and Jacobson et al [13].

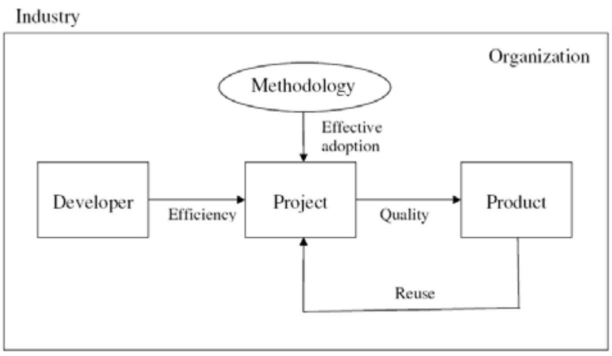

Figure 1: The preliminary ISDM adoption framework.

Figure 1 shows that a methodology can be adopted by an organization prior to being used in system development projects. The effective adoption of a methodology can lead to the efficient use of resources and improve the efficiency of developers [12]. The effective adoption can then result in both improvement of the quality of the system product and facilitation of system reuse. In the diagram, “industry” is outside and “organization” is inside the main box because the existing literature shows that although ISDM adoption is operationally managed within the organization [29], adoption is also heavily dependent on the availability and maturity of the ISDMs in the industry. With reference to the work of Kaplan and Norton on the Balanced Scorecard [15], the framework developed in this research can be considered from the perspectives of (i) industry, (ii) organization, (iii) project, (iv) product and (v) developer.

The relationships of the factors contributed by the different perspectives to the success of the adoption of a systems development methodology have been tested and validated in this study using questionnaire surveys. Drawing on the findings of prior research, the survey in the present study classified organizations into one of the following categories based on ISDM adoption status: (1) using a traditional ISDM, or (2) using an OOM or updated ISDM.

5.1 The impact of project and product characteristics on ISDM adoption

Three classes of entities can be measured in software technology metrics: processes, products and resources. According to Fenton [6], internal attributes are used to measure the processes, products and resources themselves, and external attributes are used to measure how processes, products and resources relate to their environment. Product metrics include codes, documentation and other deliverables. Attributes include size, functionality, robustness, extendibility, maintainability, quality, complexity, usability and reusability. Quality is an external attribute by which a product is measured. The term “quality” can have different meanings, and numerous research efforts have been made on the subject of quality in systems and systems development. The present research measured satisfaction with the product from the standpoint of developers, users and management.

5.2 The impact of developer characteristics on ISDM adoption

Diffusion is the process through which an innovation is communicated along certain channels in the course of time among the members of a social system. It is a special type of communication and persuasion, of which the content of messages is concerned with new ideas and concepts. However, new ideas and concepts are not acquired by system developers naturally and without effort. Diffusion is a mode of social change, and professional development opportunities are paramount in affecting social change. In ISDM diffusion, the social system is the systems development community and ISDMs are sets of complex ideas. Rogers [24] noted that the communication and use of complex innovative technology require new sets of skills. In this regard, system developers need considerable training and frequent retraining.

6. Research Method

The research design involved three distinct phases. Initially, a list of open-ended questions relating to the adoption of systems development methodologies in organizations was prepared based on issues identified from the related literature including [1, 9]. The validity of these questions was pilot tested with interviews with 10 senior software developers, each with over 15 years of system development experience and in possession of a master’s or doctoral degree. Using the descriptive comments of these interviewees and information collected from further analysis of the literature, the questions were then organized into an interview script. Twelve senior system development professionals who were advisory committee members for the curriculum for a degree in information technology at a publicly funded university in Canada were invited to comment and verify the interview scripts through a focus group discussion.

After conducting 10 interviews using the open-ended questions, the interview scripts and focus group discussions results were organized into a survey questionnaire. Between the summer of 2006 and early 2007, the revised questionnaire survey was uploaded onto a Web site and e-mail invitations were sent to 2,568 organizations. Anonymous responses were obtained from 223 respondents, out of which 202 sets of data were determined to be usable.

The survey subjects were mainly selected from among individuals holding the job title of project manager, information systems manager, IT manager or the equivalent in the membership directories available from IT professional bodies. The collected data were used to test the hypotheses proposed in this study. As the ISDM adoption decision is implemented by the project team of an organization, the responses were considered to provide the appropriate level of support to the research. Most empirical studies of technological impacts rely on subjective and perceptual data, and the survey approach is well supported. Moore and Benbasat [20] indicated that subjective perceptions provide a good basis for developing a theoretical framework to measure the perceptions of adopting an information technology innovation.

7. Survey Findings

One can assume that ISDM adoption decisions are based on rational decisions about the impact of such adoption on organizations, projects, system products and developers. Based on the work of Fitzgerald [8] and Goldberg and Rubin [9], the hypotheses for the reasons not to adopt an ISDM are shown in Table 2, while the hypotheses for the reasons to adopt an ISDM are shown in Table 3.

The survey findings of this research showed that despite the advantages claimed for the use of new methodologies such as OOMs, their adoption in organizations remains a challenge. Not only are the majority of systems in operation developed using traditional methods, the systems currently being developed and under consideration are still using such methods (see Table 1). Many organizations use both OO and non-OO methods, and are found to still be in the pilot testing stage in their adoption of an OOM. It is evident that most organizations take an evolutionary approach, customizing an OOM and integrating it into their existing systems development method. In fact, in-house methodologies that are tailor-made to fit system developers work well in system development organizations. These methodologies are typically derived from a combination of methodologies, and use tools that fit the characteristics of the development teams. The survey findings support the observation that organizations should first evaluate and pilot the methodology before customizing and institutionalizing it.

The questionnaire asked respondents to name the methodologies that they were using for a project currently being developed. Table 1 summarizes the total number of projects developed by the organizations of respondents and the types of methodologies used. The results show that even organizations that adopted OOMs customized the methods to suit the needs of their software developers.

Table 1: Comparing the types of systems development methodologies

Use of ISDM in organizations	No. (percent)
Not using any formal methodology	14 (14%)
Using a formal process-oriented methodology	31 (23%)
Using a formal object-oriented methodology	49 (36%)
Using an in-house customized methodology that utilizes both process-oriented and object-oriented methodologies	43 (31%)

7.1 Relationship of adoption with reasons not to use an ISDM

Table 2 summarizes the results of using an independent t-test to compare the means of the perception of ISDM adoption between the two groups of survey respondents. One group included organizations that had already adopted a new ISDM, while the other included those that had not yet adopted such a methodology. The critical value found from the t-distribution at a 95% confidence level with a degree of freedom of 200 was 1.645. Most p values were less than 0.000, except that of H1.11, which was marginal (p = 0.077). None of the others was larger than 0.050, indicating that the results were statistically significant. The null hypotheses of equal means thus could not be established for hypotheses H1.1-H1.13. It was concluded therefore that the perceptions of the reasons not to adopt an ISDM differed between the two groups.

Table 2: t-test for reasons not to use an ISDM

Ref.	Critical factor	t	df	p
H1.1	Have never considered adopting an ISDM	-6.272	200	.000
H1.2	Immature technology	-4.511	200	.000
H1.3	Lack of industrial standards	-4.309	200	.000
H1.4	Lack of training in new methodologies	-4.439	200	.000
H1.5	Resistance from staff	-6.877	200	.000
H1.6	Perceived as economically infeasible	-3.183	200	.002
H1.7	Satisfied with existing methods	-6.193	200	.000
H1.8	Techniques are not well defined	-5.330	200	.000
H1.9	Tools are difficult to use	-6.823	200	.000
H1.10	Methods are not well defined	-2.696	200	.008
H1.11	Processes are incomplete	-1.778	200	.077
H1.12	Lack of support for developers	-5.060	200	.000
H1.13	Developers too busy to change	-7.838	200	.000

7.2 Relationship of adoption with reasons to use an ISDM

Table 3 summarizes the results of conducting an independent t-test to compare the means of the perception of ISDM adoption between the two groups of survey respondents in the same manner as that outlined in the previous section. The critical value found from the t-distribution at a 95% confidence level with a degree of freedom of 200 was 1.645. Most p values were equal or close to 0.000, and none of them was larger than 0.005, indicating that the results were statistically significant. The null hypotheses of equal means could not be established for hypotheses H2.1-H2.14. It was concluded therefore that the perceptions of the reasons to adopt an ISDM differed between the two groups.

Table 3: t-test for reasons to use an ISDM

Ref.	Critical factor	t	df	p
H2.1	Improve the development time	11.376	200	.000
H2.2	Keep up with current technology	3.107	200	.002
H2.3	Improve maintenance	5.669	200	.000
H2.4	Consistent with prior systems	6.471	200	.000
H2.5	Best choice for user interface	5.091	200	.000
H2.6	Build a reuse library for the future	2.983	200	.003
H2.7	Facilitate accurate system requirements	13.030	200	.000
H2.8	Facilitate correct system specifications	13.030	200	.000
H2.9	Have clearly defined development responsibilities	7.409	200	.000
H2.10	Develop best-in-class systems	5.321	200	.000
H2.11	Have a repeatable system development process	8.577	200	.000
H2.12	Reduce system development uncertainty	7.923	200	.000
H2.13	Have traceable system events	9.319	200	.000
H2.14	Be more resilient to change	8.654	200	.000

During the interviews and focus group meetings, IS managers indicated that organizations often had to customize the chosen systems development methodology to meet their specific needs, such as the skills and culture of their developers, or the nature of the systems they were developing. To properly deploy the technology, they also had to provide support so that the customized technology could be institutionally internalized. In essence, the empirically observed stages of systems development methodology adoption, in common IT terms, should include (i) evaluation, (ii) pilot testing, (iii) customization and (iv) institutionalization. The process can be matched with Rogers's innovation decision process and innovation process model, as shown in Table 4.

Table 4: Empirically observed stages in the selection and adoption of systems development methodologies

Rogers' stages of the innovation decision process	**Rogers' stages of the innovation process**	**Empirically observed stages in the selection and adoption of an ISDM**
Knowledge, Persuasion	Agenda setting	Evaluation
Decision	Matching	Pilot testing
Implementation	Redefining/ Restructuring, Clarifying	Customization
Confirmation	Routinizing	Institutionalization

8. Discussion

Before taking a closer look at how the independent variables worked in relation to ISDM adoption, a correlation matrix of these variables was produced. It was discovered that the independent variables shown in Tables 6 and 7 were significantly correlated with each other. One of the requirements of regression analysis is that the independent variables be mutually independent of each other. When the variables are highly correlated (i.e., > 0.7), it is not worthwhile to run regression analysis (in this case, on ISDM adoption) with this group of variables alone. Ways to overcome this problem include making predictions using other statistical procedures such as discriminant function or factor analysis. Discriminant function analysis, however, cannot distinguish the sign, whereas factor analysis reduces the number of independent variables to a few important factors that are independent of each other.

Hence, to extract the important components from the 27 negative and positive reasons, factor analysis was deployed. The process is effective not only for condensing many variables into a few important components but also for determining the relative contribution of the variables to each component. Table 5 shows the results of factor analysis using the method of principal component analysis. To reduce the complexity of the table, only the factor loading coefficients with the highest value in the row, which are the most important ones, are shown in the table.

On close examination, variables in the first component (column two of Table 5) are related to the performance of projects and thus can be regarded as attributing to the "project perspective". Variables in the second component (column three of Table 5) are related to the availability and status of the systems development methodologies in the industry and thus can be regarded as attributing to the "industry perspective". Variables in the third component (column four in Table 5) are related to the internal policies, strategies and operations of the organization and thus can be regarded as attributing to the "organization perspective". Variables in the fourth component (column five of Table 5) are related to product development and thus can be regarded as attributing to the "product perspective". Variables in the fifth component (column six of Table 5) are related to the perceptions and attitudes of developers and thus can be regarded as attributing to the "developer perspective". Table 6 summarizes the list of critical variables contributing to each perspective.

Table 5: Results of factor analysis

	Component				
	1	2	3	4	5
H1.1: Have never considered adopting an ISDM			.471		
H1.2: Immature technology		.429			
H1.3: Lack of industrial standards		.508			
H1.4: Lack of training in new methodologies			.539		
H1.5: Resistance from staff					.366
H1.6: Perceived as economically infeasible			.423		
H1.7: Satisfied with existing methods					.330
H1.8: Techniques are not well defined		.343			
H1.9: Tools are difficult to use		.270			
H1.10: Methods are not well defined		.513			
H1.11: Processes in methodology are incomplete		.542			
H1.12: Lack of support for developers			.521		
H1.13: Developers too busy to change					.515
H2.1: Improve the development time	.746				
H2.2: Keep up with current technology			.503		
H2.3: Improve maintenance	.580				
H2.4: Consistent with prior systems				.521	
H2.5: Best choice for user interface				.523	
H2.6: Build a reuse library for the future			.607		
H2.7: Facilitate accurate system requirements	.838				
H2.8: Facilitate correct system specifications	.835				
H2.9: Have clearly defined development responsibilities	.611				
H2.10: Develop best-in-class systems	.615				
H2.11: Have a repeatable system development process	.651				
H2.12: Reduce system development uncertainty	.650				
H2.13: Have traceable system events		.702			
H2.14: Be more resilient to change		.699			

Table 6: Critical variables of each perspective

Perspective	Critical variables
Industry	• Maturity of methodologies in the industry • Existence of standards • Techniques are well known and defined • Tools are available and easy to use • Methods are well defined • Methodologies cover all major processes • Methodologies facilitate traceable system events • Methodologies are resilient to change
Organization	• Due consideration given to systems development methodologies • Training provided to software developers • Economically feasible in the organization • Support provided to software developers • Strategy to keep up with current technology • Building a reuse library for the organization
Project	• Improvement in system development time • Improvement in system maintenance • Facilitation for the collection of accurate system requirements • Facilitation for a correct system specification • Facilitation for clearly defined responsibilities in the project • Facilitation for a best-in-class system development environment • Provision for repeatable system development processes in the project • Reduction in system development uncertainty in the project
Product	• Compatible with prior systems • Best choice for user interface
Developer	• Resistance of staff • Developers satisfied with existing methods • Developers too busy to make changes

9. The Adoption Framework

A comprehensive framework for the effective selection and adoption of systems development methodologies by organizations is proposed in this paper. In this framework, the critical factors are classified based on industry, organization, project, product and developer perspectives. The stages of adoption and implementation include evaluation, pilot testing, customization, and institutionalization. The methodology elements that are considered include tools, techniques and processes.

9.1 Illustration of the Three-Dimensional Adoption Framework

The abovementioned considerations can be represented using a three-dimensional diagram, as illustrated in Figure 2.

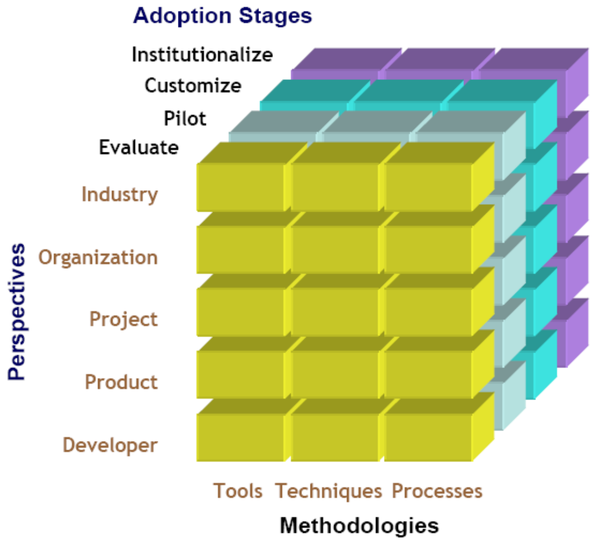

Figure 2: A framework for the effective adoption of an ISDM.

The proposed framework for ISDM adoption provides an overall picture of the different stakeholders and considerations. Each perspective can be broken down and explored further to meet the specific requirements of an organization. Thus, the proposed framework can be used as a generic guideline for the effective adoption of systems development methodologies.

9.2 Applicability of the Three-Dimensional Adoption Framework

According to Fettke and Loos [7], the beginnings of framework formulation and reference modeling can be traced back to at least the 1930s, and studies in this area have recently aroused considerable interest. A well-formulated framework should be reusable by various organizations under different situations. The use and reuse of a framework will lead to decreases in cost, implementation time, and degree of risk in the adoption of a systems development methodology by an organization, while at the same time the quality of development will increase. As it is difficult, if not impossible, to have a universal framework that can be used by all organizations operating under all situations, the objective of the present research was to propose a framework that is sufficiently generic to model various types of systems development methodologies for organizations that are developing different types of systems in specific environments. The framework can be applied in organizations using one or more of the following scenarios.

1. Analogy – an organization can apply the proposed framework in its adoption decision by matching the perceived similarities of the framework in some

aspects, including but not limited to the industrial environment and organizational characteristics, to the projects, products and developers of the organization.

2. Specialization – an organization can derive a new framework from the one proposed, with the aim of making it more suitable for a specific situation. For example, the revised framework could be used to match a particular class of methodologies to specific types of projects or products.
3. Aggregation – an organization can combine or regroup one or more original items in the framework before applying it. For instance, the categorization of some decision variables may not be readily distinguishable in the framework so these may be combined or regrouped. Examples of such decision variables include "support for reuse" and "technical training," where the organization is the supplier of the support/training but the developer is the consumer.
4. Instantiation – an organization can create and apply another generic framework by replacing items in the proposed framework with a set of matching items that are appropriate to the actual situation. For example, it can replace the references to tools, techniques and processes in the decision variables with a set of actual known and specific tools, techniques, and processes.
5. Configuration – an organization can choose and apply a subset of the proposed framework that is relevant to the specific characteristics and environment of the organization from the larger number of items and alternatives available in the framework.

Empirically observed data indicate that the abovementioned situations are common, but are not the only ways to apply the framework. In reality, an organization will typically combine two or more of these scenarios and select the combination of methodologies that is most suitable to its situation.

10. Conclusion

In this research, the issues that were found to be important to the effective selection and adoption of systems development methodologies were formulated into a framework. A framework should be able to resolve problems and be generally applicable in the field. To meet this requirement, rather than proposing a model that is specific to a particular situation, a framework that can be customized by way of analogy, specialization, aggregation, instantiation and/or configuration to meet the needs of different organizations is proposed.

When making a decision about the adoption of a systems development methodology, which will have a far-reaching impact on the organization, a framework that can consider all aspects of the methodology including the tools, techniques and processes, as well as the different responsibilities and actions of concerned parties, is very important. The proposed framework provides a strategic map that covers both external industry-related factors and internal (organization, project, product and developer) ones. In addition, this framework is believed to provide a generic solution to the selection and adoption of systems development methodologies.

Bridging the gap between planning and realization and ensuring that outcomes are measurable are important factors that contribute to the success of ISDM adoption. The framework proposed in this paper can be used to communicate critical factors to all concerned parties during the evaluation, pilot testing, customization and institutionalization stages. The framework can be used to document the implementation, measure the outcomes and monitor the results of adoption.

References

[1] Avison D. and Fitzgerald G., *Information Systems Development: Methodologies, Techniques, and Tools*, Third Edition, McGraw-Hill, London, 2003.

[2] Avison D. and Fitzgerald G., *Information Systems Development: Methodologies, Techniques, and Tools*, Fourth Edition (International Edition), McGraw-Hill, London, 2006.

[3] Aydin M., Harmsen F., Slooten K. and Stegwee R., "A model for a method adaptation process", *Information Systems Development: Advances in Theory, Practice, and Education*, Edited by Vasilecas V., Caplinskas A., Wojtkowski W., Wojtkowski G., Zupancic J. and Wrycza S., Springer, New York, 2005, pp. 477-486.

[4] Bayer J. and Melone N., "A critique of diffusion theory as a managerial framework for understanding adoption of software engineering innovations", *Journal of Systems and Software*, Vol. 9, Feb. 1989, pp. 161-166.

[5] Champeaux D. and Faure P., "A comparative study of object-oriented analysis methods", *Journal of Object-Oriented Programming*, March/April 1992, pp. 21-33.

[6] Fenton N., *Software Metrics: A Rigorous Approach*, Chapman & Hall, London, 1991.

[7] Fettke P. and Loos P., *Reference Modeling for Business Systems Analysis*, Idea Group Publishing, Hershey, PA, 2007.

[8] Fitzgerald, B., "An investigation of the use of systems development methodologies in practice", *Proceedings of the 4th European Conference on Information Systems*, Lisbon, 1996, pp. 143-162.

[9] Goldberg A. and Rubin K.S., *Succeeding with Objects: Decision Frameworks for Project Management*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1995.

[10] Grass R., "The realities of software technology payoffs", *Transaction for Computing Machinery, Communications of the ACM*, Vol. 42, No. 2, Feb. 1999, pp. 74-79.

[11] Iivari J., "Object-oriented as structural, functional and behavioural modelling: A comparison of six methods for oriented-oriented analysis", *Information and Software Technology*, Vol. 37, No. 3, 1995, pp. 155-163.

[12] Iivari J., "Object-oriented information systems analysis: A framework for object identification", *IEEE Transactions*, Vol. 2, 1991, pp. 205-218.

[13] Jacobson I., Cristerson M., Jonsson P. and Overguard G., *Object-Oriented Software Engineering: A Use Case Driven Approach*, Revised Printing, Addison-Wesley Longman, Redwood City, CA, 2004.

[14] Johnson R.A., "The ups and downs of object-oriented systems development", *Communications of the ACM*, Vol. 43, No. 10, 2000, pp. 69-73.

[15] Kaplan R. and Norton D., *The Balanced Scorecard: Translating Strategy into Action*, Harvard Business School Press, Boston, MA, 1999.

[16] Kaplan R. and Norton D., *Strategy Maps: Converting Intangible Assets into Tangible Outcomes*, Harvard Business School Press, Boston, MA, 2004.

[17] Ladden R.M., "A survey of issues to be considered in the development of an object-oriented development methodology in Ada", *Ada Letters*, Vol. 9, No. 2, March/April 1989, pp. 78-88.

[18] Losavio F., Matteo A. and Schlienger F., "Object-oriented methodologies of Coad and Yourdon and Booch: Comparison of graphical notations", *Information and Software Technology*, Vol. 36, No. 8, 1994, pp. 503-514.

[19] Monarchi D. and Puhr G., "A research typology for object-oriented analysis and design", *Communications of the ACM*, Vol. 35, No. 9, Sept. 1992, pp. 35-47.

[20] Moore G.C. and Benbasat I., "Development of an instrument to measure the perceptions of adopting an information technology innovation", *Information Systems Research*, Vol. 2, No. 3, 1991, pp. 192-222.

[21] Mrdalj S., "Bibliography of object-oriented system development", *ACM SIGSOFT Software Engineering Notice*, Vol. 15, No. 5, 1990, pp. 60-63.

[22] Ovaska P., "Working with methods: Observations on the role of methods in systems development", *Information Systems Development: Advances in Theory, Practice, and Education*, Edited by Vasilecas V., Caplinskas A., Wojtkowski W., Wojtkowski G., Zupancic J. and Wrycza S., Springer, New York, 2005, pp. 185-197.

[23] Rifkin S., "Why software process innovations are not adopted", *IEEE Software*, Vol. 18, No. 4, 2001, pp. 110-112.

[24] Rogers E.M., *Diffusion of Innovations*, Fifth Edition, Free Press, New York, 2003.

[25] Tolvanen J.P., Incremental Method Engineering with Modeling Tools – Theory Principles and Evidence, Department of Computer Science and Information Systems, University of Jyvaskyla, Finland, 1998.

[26] Vessey L., Jarvenpaa S.L. and Tractinsky N., "Evaluation of vendor products: Case tools as methodology companions", *Communications of the ACM*, Vol. 35, No. 4, April 1992, pp. 90-105.

[27] Whitten J. and Bentley L., *Systems Analysis and Design Methods; special contributions by Kevin Dittman*, McGraw-Hill, New York, 1998.

[28] Whitten J. and Bentley L., *Systems Analysis and Design Methods; with contributions by Gary Randolph*, Seventh Edition, McGraw-Hill, New York, 2007.

[29] Woo F., Mikusauskas R., Bartlett D. and Law R., "A framework for the effective selection and adoption of software development methodologies", *Proceedings of the ACMSE Conference*, 10-12 March 2006, Melbourne, Florida, US, 2006b, pp. 198-203.

[30] Woo F., Mikusauskas R., Bartlett D. and Law R., "Is OO the Systems Development Technology for Your Organization?", *Proceedings of the 4th ACIS International Conference on Software Engineering Research, Management and Applications*, published by the IEEE, 9-11 August 2006, Seattle, Washington, US, 2006c, pp. 354-361.

[31] Wynekoop J. and Russo N., "Systems development methodologies: Unanswered questions", *Journal of Information Technologies*, Vol. 10, 1995, pp. 65-73.

Dr. Fleming Woo is an IT professional with over 35 years of experience in systems development, e-learning technologies, infrastructure systems and data centre management, including 12 years as Department Chair at the Kwantlen Polytechnic University in Canada. In the past 9 years, he has taken on the position of Associate Director of Information Technology Services at The Hong Kong Polytechnic University and has also been acting as Director. He holds a BSc degree in Mathematics, an MSc degree in Engineering and a PhD in Computing from the Hong Kong and UK universities. He is also Chartered Fellow of the British Computer Society and Fellow of the Hong Kong Institution of Engineers.

Dr. Romas Mikusauskas received MSc in Mathematics from Vilnius State University (Lithuania) in 1974 and PhD in Mathematics and Physics from Joint Institute for Nuclear Research (Dubna, Russia) in 1986. During his professional career since 1974 he used to work in academy, IT industry and education as well as participated in a number of projects. Currently he is a Senior Lecturer in the Faculty of Computing at London Metropolitan University. Teaching and research interests are in 3D modelling, computer graphics, image processing, systems analysis and design, IS development methodologies, and software engineering.

Professor Dean Bartlett is a Chartered Occupational Psychologist and Professor of Organizational Psychology. He obtained his doctorate from the University of Southampton and also has Bachelor's and Master's degrees from the University of Southampton and the University of London. He is the Director of the Management Research Centre, based in London Metropolitan Business School and is a Fellow of the Higher Education Academy. He specialises in the application of psychology to the workplace. He has published widely in a number of peer-reviewed journals and has particular research interests in the management of change and innovation.

Professor Rob Law received his PhD (1994), MSc (1990), and BASc (1988) in Computing Science from universities in Canada. He is presently a Professor of Information Technology at the Hong Kong Polytechnic University's School of Hotel & Tourism Management.