A Framework for the Effective Adoption of Software Development Methodologies

Fleming Woo*, Romas Mikusauskas**, Dean Bartlett**, Rob Law*

*The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{+852 2766 4565, +852 2766 6349}
{itfwoo, hmroblaw}@polyu.edu.hk

**London Metropolitan University
London, England
{+44 20 7973 4822, +44 (0) 20 7133 3215}
{r.mikusauskas, d.bartlett}@londonmet.ac.uk

ABSTRACT

Significant developer productivity gains and software quality improvement are expected with the use of software development tools and methods. However, recent studies reveal that the adoption of innovative technologies in software development has been undersubscribed. As different types of systems may need different kinds of software processes, programming paradigms and tools, no single set of solutions is best for all situations. The purpose of this study is to propose a framework for organizations to effectively select and adopt systems development methodologies so that they can decide on the most suitable tools and methods for their specific environment.

General Terms

Management, Measurement, Performance, Design, and Standardization.

Keywords

Systems development methodologies, software selection, technology adoption, diffusion of innovations, software process improvement, software reuse, quality and productivity.

1. INTRODUCTION

Many tools and methods have emerged in response to the huge demand for software development and the increased complexity of information systems. Some of these include: fourth generation languages (4GL), computer-aided software engineering (CASE), structured methodologies, object-oriented methodologies (OOM), formal methods, personal software process (PSP), eXtreme programming, component programming, relational database management systems (RDBMS) and object-oriented database management systems (ODBMS). However, literature reveals that the rate of adoption of innovative technologies in software development is slow [15], there is a lack of study on the evaluations and selections of technologies, and often many claimed benefits cannot be realized [4].

In particular, the object-oriented methodologies have been cited in many publications as having clear advantages over the structured methodologies, such as more traceable to system requirements, better transition between requirement analysis and design, more resilient to system changes, producing more reusable components and delivering more stable resulting systems. However, many systems developers do not view object-orientation as a way to solve their problems and keep using the structured methodologies for their planning, requirement analysis and design of software systems, even though the system may be finally constructed in Java which is an object-oriented tool. As a web programming tool, Java is often adopted because increasingly more business applications are web-based and deploying extensive amount of graphical user interface. Developers are also staying on using relational database management systems in preference to object-oriented database management systems.

Apparently, there should be some reasons for the slow adoption rate and a framework is required to facilitate organizations to evaluate, select and adopt the most suitable set of tools, techniques and methods for their specific environment. A framework that can identify factors affecting the selection and adoption and the responsibility and action from different parties will help management to make sound decisions on systems development methodologies.

The framework in this research will involve a critical examination of industry wide factors, organization factors, project characteristics, outcome of the software product and individual developer behaviors. The framework will consider the adoption of tools and methods, the efficiency in the deployment of resources and the effectiveness, usability and reusability of the system deliverables as a result of the adoption.

ACM SE'06, March, 10-12, 2006, Melbourne, Florida, USA
Copyright 2006 1-59593-315-8/06/0004…$5.00.

2. BACKGROUND

The existing literature on evaluation frameworks for software development methodologies focuses on comparing and evaluating methodologies from the perspectives of methods [2], graphical notations [11], and structures, functions and behaviors [5]. Iivari [6] observed that there is a lack of frameworks for evaluating the work done in OOMs. Further, Mrdalj [14] identified a need for the collection, classification, and evaluation of the available materials. Ladden [10] analysed the factors that should be considered in developing an OOM.

Monarchi and Puhr [12] recognized the need for an evaluative model for OOMs rather than a prescriptive model. They evaluated 23 different methodologies based on the completeness and cohesiveness of processes and representations, and then identified their strengths and weaknesses. Surprisingly, very little research has been conducted to further the methods for evaluating software development methodology in recent years.

The evaluation framework developed in this research is based on the perceived outcomes from adopting the methodology, including the productivity of deploying development resources, the quality and reusability of deliverables. The framework provides a comprehensive examination on the tools, techniques, processes, methods and productivity improvement as well as the quality of software deliverables resulting from the adoption of the chosen methodologies.

3. 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 and projects was prepared based on issues identified from literature including [1] and [3]. The validity of these questions was pilot tested with interviews of ten senior software developers each with over 15 years of software development experiences and in possession of masters and doctoral degrees. Using the descriptive comments of these interviewees and information collected from further literature analysis, the questions were then organized into an interview script. Twelve senior software development professionals who were Advisory Committee Members of a degree programme in Information Technology at a publicly funded university in Canada were invited to comment and verify the interview scripts during a panel discussion.

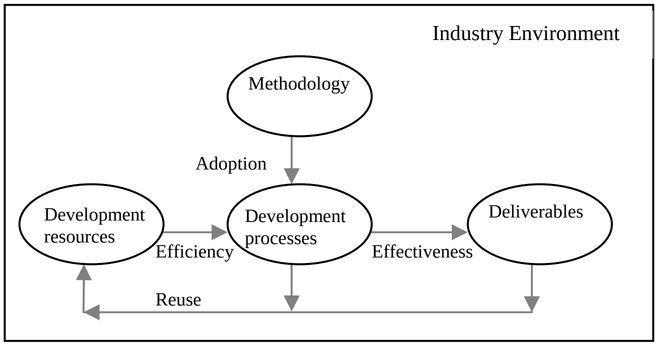

Figure 1: The Initial Framework

Based on the interviews and panel discussions, an initial framework shown in Figure 1 was laid out as a starting point.

Further studies in literature revealed that an adoption framework is not available but it is required to provide a comprehensive picture on different considerations for the effective adoption of systems development methodologies. In this regard, literature on the studying of management strategies was referenced.

Questions used for the interviews and panel discussions were also organized into a survey questionnaire and copies of which were distributed to 1,109 organizations engaged in software development in Canada. The survey subjects were mainly selected from individuals holding the job titles of project manager, information systems manager and above in the membership list published by the Canadian Information Processing Society (CIPS).

Anonymous responses were obtained from 82 organizations encompassing 85 system development projects across ten Provinces in Canada. The data collected were used to test the hypothesis supporting the proposed framework. As systems development methodology adoption decision is implemented at the project team of organizations, the responses were considered supporting the research at the appropriate level.

In summary, the framework for the adoption of systems development methodologies was firstly formulated from interviews, panel discussions and literature. Quantitative data collected from the survey were used to validate the adoption framework. As most empirical studies on technological impacts rely on subjective and perceptual data, the approach of using a survey was well supported by the work of Moore and Benbasat [13] who indicated that subjective perceptions provide a good basis for developing a theoretical framework to measure the perceptions of adopting an information technology innovation.

4. THE ADOPTION FRAMEWORK

Efficiency, effectiveness and reusability are major dimensions of the evaluation framework. A methodology is not efficient if it cannot enhance the productivity in systems development. It is 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 software development process. In addition to developers' productivity, it also includes other factors such as ease of use, repeatability, adaptability, integration with other tools, amount of uncertainties, responsiveness to changes, accountability, traceability and manageability. On the other hand, effectiveness is the external performance view focusing on the outcomes of the software development process. Besides the quality of the resultant software, effectiveness includes functionality, correctness, robustness, reliability, extendibility and maintainability of the system product. Finally, reusability can be addressed in terms of but not limited to its reuse objectives, impact on developer productivity and problems in reuse. An important value of a framework is to consider all these factors and present a consolidated view for the management to make informed decisions.

Although some models exist for the evaluation and selection of systems development methodologies, there is no well-recognized framework exists for the adoption process. In view of the rapid changing business and technology environment, there is a growing need for a framework that can be used to guide the selection and adoption of the most suitable systems development methodologies for a specific organization. This paper proposes a general framework that can be customized to specific organizational environment.

From a search of current literature, there is no framework available for the adoption of system development methodologies. One possible solution is to cross-reference a framework with similar structure and mechanisms. The Balanced Scorecard (BSC) introduced by Norton and Kaplan [7] in the early 1990s is a management tool to formulate strategies. Subsequent development of BSC has shifted to focus on measurement of strategic achievements [8] and implementation and monitoring [9] in organizational performance improvement to ensure that a strategy function is objective-oriented.

In the basic model of BSC, a strategy is examined from four perspectives: financial, customer, internal, and learning and growth. These perspectives can vary from one organization to the other depending on the scope of the problem under investigation. For each of the perspectives, a set of critical success factors and metrics can be formulated according to the vision and strategy, strategic emphases and operational policies.

The proposed adoption framework for systems development methodologies has five strategic perspectives: industry, organization, project, product, and developer. Different from the BSC framework, the critical factors are constructed on the basis of the organization's efficiency, effectiveness and reuse objectives. These factors can be translated into criteria that can be either quantitative or qualitative. Typically these factors include:

Industry perspective:

- Known techniques
- Available tools
- Available methods
- Existing standards

Organization perspective

- Quality objectives
- Best-in-class systems
- Cost-effective production
- Efficient use of resources
- Gaps in skill inventory
- Reuse strategy
- Support for reuse

Project perspective

- Accurate system requirements
- Correct system specification
- User-friendly system interface
- Repeatable processes
- Reduced uncertainties
- Responsive to changes
- Traceable system events
- Improved project manageability
- Adoption of a well-chosen methodology
- Reuse from prior projects
- Reuse for subsequent projects

Product perspective

- Reduced defect density
- Functionally correct
- Secure
- Robust
- Reliable
- Extendible
- Maintainable

Developer perspective

- Technical competence
- Cross functional skills
- Domain knowledge
- Adaptable to changes
- Ability to learn
- Communications skills
- Team work skills

The relations among these factors are illustrated in Figure 2.

Even though BSC involves different parties and encompasses different perspectives, by looking at the structure of information exchange, it parallels the requirements of those in the adoption of system development methodologies. The BSC framework is chosen in this paper also because it is widely known and well recognized for its applicability in businesses.

The proposed adoption framework for system development methodologies can provide an overall picture of different stakeholders and considerations. To avoid the impression that different perspectives are strictly hierarchical in structure and to facilitate the looping back in relationship among factors, different perspectives of the adoption framework are laid out horizontally in one layer. Each perspective can actually be broken down and explored further to meet the specific requirements of an organization. Thus, the proposed framework can provide a generic guideline for the effective adoption of system development methodologies.

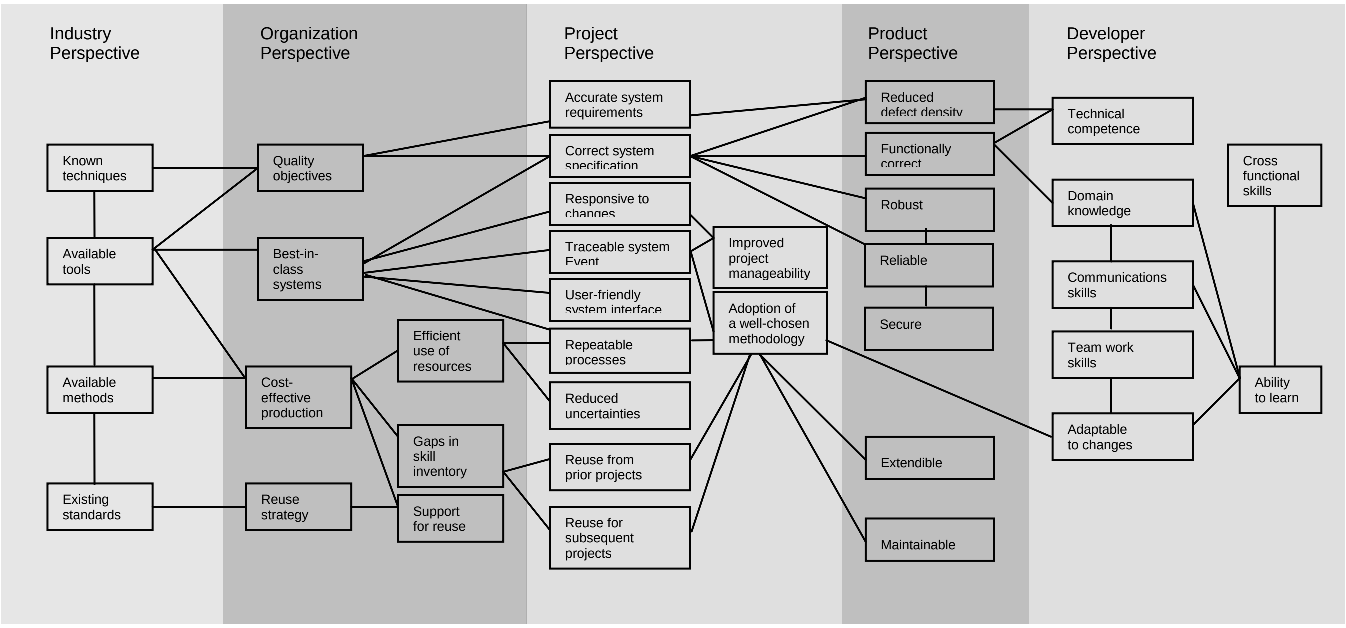

Figure 2: An Illustration of the Proposed Adoption Framework for Systems Development Methodologies

5. TEST OF HYPOTHESES

It is most ideal to have a universal and perpetual framework. In reality, every framework needs to be fine-tuned in an ongoing way. The most obvious challenge to any framework is the measurement of its validity with real-life projects. It is especially important to try fitting in different factors affecting the adoption of systems development methodologies according to different industries, organizations, projects, products, and developing environments. Besides the technological aspects, cultural and skill level differences should be considered when evaluating different perspectives.

To test the applicability of the framework, it is possible to put forward a series of hypotheses in the form of cause-and-effect relationships based on the factors affecting the adoption decision in the real world. In practice, the number of critical factors should not be infinitely many. For instance, the hypotheses statement could be:

"The existence of industry standards and competence of developer in systems development methodology, are important considerations in the adoption of a methodology, will also contribute to fulfill the business goal of making profits. If we can deploy suitable tools, the defect density of the system deliverables will be decreased that contributes to the favorable consideration in adopting a particular methodology. The deployment of software development tools will also enhance the user interface of the final software product and increase developer's productivity, which is in turn a good reason for choosing a methodology. Developer productivity may be decreased due to the sophistication of user interface. Reusing software components, although may negatively affect the developer's satisfaction in the development process, will increase development productivity and is an important factor for the adoption of a methodology."

The hypotheses above can be represented using the proposed framework in a diagram shown in Figure 3. The factors affecting the choice of methodology as indicated in the hypothesis are connected in the form of a simple path diagram. Using the data collected from the survey with 85 systems development projects in Canada, the coefficient of correlation (ß) and its significance level (s) between the relationships are indicated on the connections. For a value of zero in the coefficient of correlation, it means the connecting factors are unrelated. For a value of one or negative one, it means the connecting factors are perfectly related. For the significance level, 0.05 means 95% significant or the chance of error due to sampling is 5%. For the significance level of 0.00, the chance of error due to sampling is less than 0.5% or 5 out of 1000.

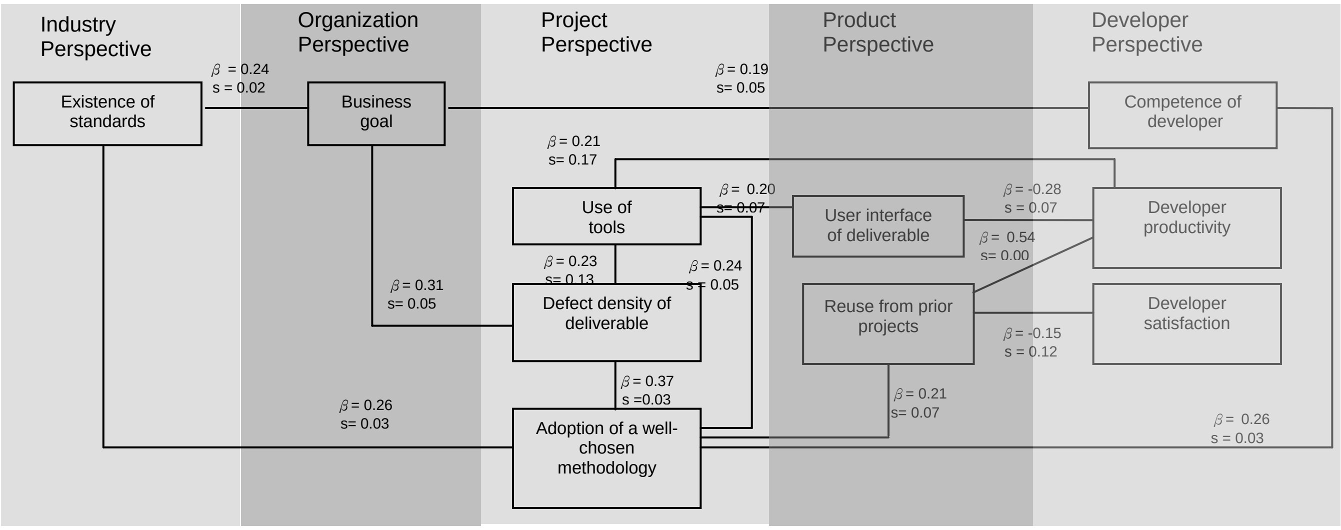

Figure 3: Path Analysis Diagram Used to Test the Hypotheses

For a negative value in ß, it means that the factors have a negative effect on each other. For instance, referring to Figure 3, a high level of requirement in the user interface of deliverable will have a negative effect (ß = -0.28) on the level of developer productivity. Similarly, a high level of software reuse from prior projects will have a slightly negative effect (ß = -0,15) on the developer satisfaction of the project. In the case of factors using a ordinal scale with few categories (i.e. business goal, use of tools, etc.) the sign of ß is dependent on the choice of ascending or descending values of the scale and, in most cases, is quite arbitrary.

To put the framework into practice, besides of using primary data from surveys and secondary data from industry, organizations should collect data regarding their organizational objectives, culture, project characteristics, developer competence, system implementation and technology deployment. The organizational data together with the adoption framework will provide continuous feedback for them to fine-tune, select and adopt innovative technologies pertaining to their systems development environment.

6. CONCLUSION

When making a significant decision, such as the adoption of a system development methodology which will have a far-reaching impact on the organization, a framework that can identify the processes, different responsibilities and action parties is very important. Employing a robust framework can save significant amount of time, money and reduce the number of problems. A well-recognized framework is supposed to be generally applicable in the field.

In this paper, many issues considered important to the adoption of a system development methodology are formulated into a conceptual framework. Instead of proposing a model that can only be applied to a particular situation, a framework that can be customized to meet the needs for different organizations is proposed.

The emphasis on the proposed framework is on the importance of measurability and communication. It can be used to measure and monitor the results of adoption and communicate the critical factors to all parties concerned which are essential for the success of the adoption processes.

Very often, the process of preparing a strategy for adoption is not followed by systematic implementation and monitoring. A key problem was bridging the gap between the adoption decision by management and its implementation. The framework proposed in this paper could be used as a tool to track the implementation of systems development methodologies.

As new IT technologies emerge almost every day in our Internet induced work place, technology adoption should not be regarded as a snap shot in organizations. The framework proposed in this paper can also be used as a tool for the continuous review of past, present and future needs for methodologies in the organization.

7. REFERENCE

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

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

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

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

[5] 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.

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

[7] Kaplan R. and Norton D., The Balanced ScoreCard: Translating strategy into action, Harvard Business School Press, Boston, Massachusetts, 1999.

[8] Kaplan R. and Norton D., Strategy Focus Organization: How Balanced Scorecard Companies Thrive in the New Business Environment, Harvard Business School Press, Boston, Massachusetts, 2001.

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

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

[11] 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.

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

[13] 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.

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

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