Adoption of Usability Engineering Methods: A Measurement-Based Strategy

Eduard Metzker
Vector Informatik, Stuggart Germany
eduard.metzker@gmx.net

Ahmed Seffah
EHL-LHR, Lausanne Switzerland
seffah.ahmed@ehl.ch

ABSTRACT

In the context of a software development organization, two strategies are possible for introducing and institutionalizing new usability engineering methods. The first one, expert-based institutionalization, require to resort to third party companies or experts that can, based its previous expertise, assist the team in selecting, implementing and institutionalizing usability engineering methods and tools. The second one, a measurement-based strategy, is based on empirical evidence for learning and assessing the appropriateness, usefulness of a usability engineering method. This paper proposed to combine these two approaches in a single process metrics support environment for selecting and institutionalizing usability engineering methods. The proposed approach has been validated via in a cross-organizational empirical study involving several software engineers from five mediums to large sized software development companies.

Keywords

Metrics, usability, usability engineering methods, institutionalization, adoption, software developments organization

1. INTRODUCTION

Within the scope of this research, by adoption we refer to the process and the related tools for selecting the appropriate new software development technology while assessing their suitability to the project needs and size as well as the capability of the personnel to use effectively and efficiently the new established technology. Adoption has been always a key challenge for software development organizations [28]. It was reported that the management staff commitment and the involvement of the personnel represent the top factors that impact on the success with a new technology when first introduced [27, 29, and 30].

However, despite management efforts made by organizations to render the transition more "user-friendly", the associated help and training material, although precise and perfectly describing the new method are often delivered in an esoteric and unreadable language. Another important factor is that organizations and managers are usually overly optimist in their employees' ability to quickly master a new technology. The reality is that understanding how to apply the technology is a long and arduous process.

Furthermore, there is little hard evidence backing up new technology to be adopted, and their costs and benefits are rarely understood [1]. Without this data, choosing a particular technology or methodology for a project at hand essentially is a random act with many consequences [2]. The findings from a very large survey made by Standish group, new technology is one of the top ten reasons for projects failure or success [27].

In order to support and effective adoption, a new metrics-based approach comprising a process model and a support environment are presented in this paper. A large case study was developed to assess the acceptance of the approach by development teams. The evaluation involved 44 professional software engineers from five medium to large-sized organizations. The evaluation method is discussed including context, method, subjects, procedure and results. Implications of the results on the design of metrics-based strategy are discussed for adopting new technology and assessing their acceptance by project teams. Based on the results, a set of guidelines is derived to optimize the acceptance of metrics exploitation approaches by project personnel.

2. THE PROPOSED METRICS SUPPORT ENVIRONMENT

The overall goal of the proposed approach, called Adoption-Centric Usability Engineering (ACUE), is to facilitate the adoption of UE methods by software engineering practitioners and thereby improve their integration into existing software development methodologies and practices. ACUE is designed to support project teams in institutionalizing this abstract knowledge about UE methods and to help them transfer this knowledge into their development processes. UE methods are perceived as integrated into an existing software development process when they are adopted by the project team, i.e. when they are accepted and performed by the project team.

ACUE exploits empirical data collected in different projects to yield stronger evidence on how the method works in a certain context. The data form an empirical base to guide the improvement of UE methods and to facilitate the informed selection and deployment of UE methods in future projects. If this approach is applied repeatedly in a number of projects over time, it leads to an incremental construction of a body of evidence to guide usability engineering method selection (Figure 1).

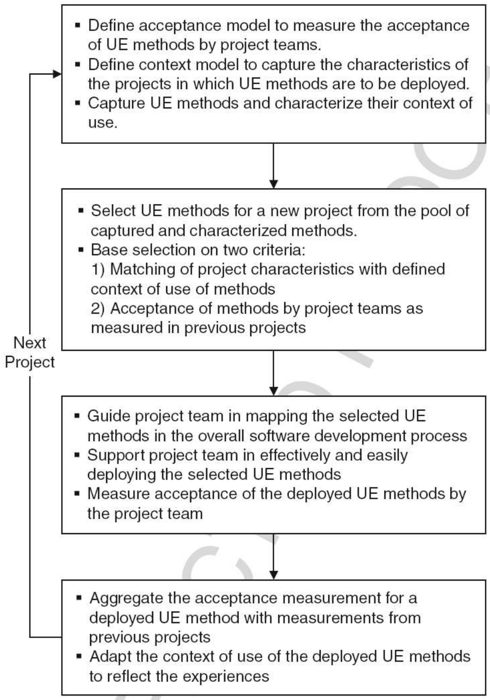

Figure 1: The overall view of ACUE approach

The support environment is called ESPrEE (Evidence-based Software PRocess Evolution Environment). The components of ESPrEE are integrated via a web portal and they be remotely accessed using any web browser. Its core functionalities are:

- At the beginning of a new project, the metrics-based method selection component of the environment is used to configure the set of usability engineering methods that will be applied in the project
- After the method selection has been completed, the environment generates a project-specific hyper-media workspace in which the methods selected are graphically visualized according to the project phases in which their usage is intended
- At the completion of major project phases or in post mortem sessions [13], the quality of the methods employed is assessed by the project team against a quality model. For this purpose quality models contain a set of quality factors and carefully defined rating scales

The core of the system is a fuzzy multi-criteria decision-making engine. The characteristics of the usability methods and projects are defined as sets of fuzzy sets. Based on these models the engine is able to compute similarity measures for projects and methods to facilitate decisions based on analogy. The engine is coupled with and assessment component. If a method receives a poor assessment in a certain project context, the method's characteristics are automatically adapted to reduce the probability of the method being selected in similar projects. On the other hand, if a method has successfully been applied in a certain project, its characteristics are adapted to increase its probability of selection in similar projects in the future.

The characteristics of the project are specified using the context models stored in the repository. A context model includes context factors to describe various project constraints such as the resources available in the project or the type of product to be developed. Each context model consists of a set of factors that can have nominal, ordinal or interval scale measures [11]. An example for an ordinal factor that describes a property of a product to be developed is 'user interface interaction complexity'. This factor may have three characteristics 'text-based interface', 'graphical user interface' or 'multi-modal interface'. Depending on the project characteristics, appropriate methods are suggested by the system. Candidate methods are ranked according to two different criteria: (1) similarity between the method and the project characteristics, (2) results of assessments from project teams that used the methods in previous projects.

Within, the system the usability engineering methods are provided in the format of method packages. Each package contains a textual description of a method that is structured according to the pyramid principle [12]. Auxiliary material such as templates, checklists and tools is linked to each package. This material facilitates easy compliance of the method described. The process guidance remains passive and does not enforce the performance of the methods proposed.

The constraints of industrial software development projects often enforce the invention of new methods or the adaptation and streamlining of existing methods. For these reasons the environments provides means for capturing methods and integrating them into the repository.

3. EVALUATION

3.1. Specific purpose of the evaluation

A large number of measurement programs are suspended or - in the worst case – failed. This is because the measurement program is not accepted by stakeholders for the following reasons [3, 4, 5, 6 and 7]:

1. The measurement process is perceived as tedious and time consuming
2. Effort and benefits of the program are poorly distributed
3. The impact on daily practice is perceived as being too low to justify sustained effort
4. Metrics support tools and/or processes are difficult to use

To examine if the proposed approach addresses these issues, the evaluation study was designed with the following questions in mind:

- Does each member of the project team understand the basic principles and structure of the method without extensive training?

- How do project managers assess the potential quantitative effects of the approach on their practices? Would they use the approach in their setting?
- Which problems the project personnel may face when applying the metrics support tool underlying the proposed framework?

3.2. Context of the evaluation

We used a set of five medium- to large-size software engineering companies developing advanced next-generation home entertainment systems, driver assistance technology for passenger cars and military support systems. The usability of these systems has been recognized by the organizations as a crucial quality factor. While usability engineering methods [10] are well-know by these companies to ensure the development of software with high usability, no experience with usability engineering was available in the engineering teams. ESPrEE was configured for this environment. Appropriate context and quality models were defined and usability engineering methods were captured in method packages. Resources included successful methods invented in previous industrial engineering projects such as reported in [16], methods distilled from literature on usability engineering [10, 17, 18], and recent research results such as Spencer's 'streamlined cognitive walkthrough'[19]. This initial population of the support tool was performed by a team of professional usability engineering experts and took about 2.5 man-months of effort. The participating organizations were part of a government-supported software engineering research consortium. However, no organization committed to adopt the approach prior to the evaluation.

3.3 The Subjects

All 44 subjects participated in the study on a voluntary basis. Of them, 39 are full-time employees working as software engineers for the companies described in section 3.2. Five subjects were graduate students working for the companies on a part-time basis. The subjects were involved in developing highly interactive software systems in a variety of domains, e.g. driver assistance systems, home entertainment, and military defense systems. Based on their experience with the development of highly interactive systems, the subjects were classified into three groups: new employees (NE, 10), software engineers (SE, 21), and usability engineers (UE, 13) [10]. In the following, these groups are referred to as user groups for reasons of simplicity.

3.4 Method

In this study, Davis' technology acceptance model (TAM) [14] was used. TAM's assessment dimensions 'perceived utility' and 'perceived ease of use' were extended while adding 'understandability' as a third dimension. TAM postulates that tool acceptance can be predicted by measuring two dimensions: the perceived usefulness (PU) and the perceived ease-of-use (PEU) of a system.

The perceived usefulness of the system expresses the "subjective probability that using a specific application system will increase (the user's) job performance within an organizational context", i.e. it is a measure for the perceived utility of the system.

The perceived ease-of-use is the main factor that influences the acceptance of a system. Davis defines perceived ease-of-use as "the degree to which the user expects the target system to be free of effort", i.e. it is a measure for the usability of a system.

Together, perceived ease-of-use and perceived usefulness constitute the person's attitude toward using a system. The attitude (A) and the perceived ease-of-use (PEU) influence the behavioral intention (BI) which can be used to predict the actual use of the system. The technology acceptance model (TAM)

TAM's dimension of perceived utility was further divided into the following sub-dimensions:

- Perceived compatibility with daily practice
- Perceived increase in efficiency
- Perceived usefulness
- Perceived support of working tasks

The sub-dimension of perceived utility was measured by qualitative effects analysis while usability was examined by studying user behavior during the user's interaction with the metrics support environment [10]. Understandability was examined via a knowledge test in which the subjects answered questions on the concepts of the overall approach and the metrics support environment. The knowledge test was performed before and after the interaction with the tool to study if and how the understanding of the concepts by the subjects changes.

To implement the study two basic techniques were deployed: questionnaires and scenario-based task solution. The purpose of the two questionnaires deployed (q1 and q2) are summarized in table 1.

Table 1: Purpose of the questionnaires deployed

	Data collected
q1	• Subject characteristics (age, qualification, professional experience) • Typical role of subject in the software engineering process • The subjects knowledge on the concepts of the approach and the metrics environment (pre-test, based on the introductory material supplied)

q2	• The subjects knowledge on the concepts of the approach and the metrics environment (post-test, based on the scenario-guided interaction with the metrics environment) • The subjects assessment of the utility and usability of the metrics environment

The scenario-based tasks that were specific for each user group forms the core of the evaluation. While the subjects were working on a task, behavioral data and any comments made while thinking out loud were captured as a basis for improving the metrics environment. Section 4.5 describes the tasks that were performed by the subjects while the exact evaluation procedure and deployment of the methods is described in section 4.6.

4.5 Tasks

To identify potential problems and study the acceptance of the metrics support environment under realistic conditions, specific usage scenarios were developed for each user group. Each scenario reflects the role of the subject and details the tasks to be solved.

All scenarios were embedded in a cover story that set a common background for all scenarios. Scenario S0 is equivalent for all user groups. In S0, the subjects were allowed to freely explore all components of the metrics environment. The other tasks to be solved in each scenario are different among the user groups (Table 2).

Table 2: Tasks to be solved in scenarios

	Tasks
NE-S1	Find an explanation of the term usability engineering. Mark the section for later exploration.
NE-S2	Find an introductory article about evaluation and tests. Review the material supplied.
NE-S3	Open the hypermedia workspace for the project DIGital. Find and open the method package on heuristic evaluation.
SE-S1	Browse all method packages available for project DIGital. Find and display a package on heuristic evaluation. Assess the quality of the method heuristic evaluation.
SE-S2	Comment the method ‚heuristic evaluation‘. Edit the method ‚heuristic evaluation‘. Extend the method package with a checklist to be used in ‚heuristic evaluation‘.
SE-S3	Create a new method package. Fill the package with given raw input material. Specify the meta-data of the methods context model. Link the package to related packages.
UE-S1	Create a new project PORTAL. Choose a context model and specify the project characteristics via the context model. Choose appropriate method packages based on the project characteristics specified. Trigger generation of hypermedia workspace for the project PORTAL.
UE-S2	Manage access to the hypermedia workspace for the project PORTAL. Invite project team members. Manage access levels of project team members.

3.6. Test procedure and scripts

Prior to the evaluation sessions, all the subjects received introductory material. It described the concepts of the metrics approach and the components of the related environment. Single subjects, who, for some reason, had no access to the material prior to the evaluation, were given the opportunity to study printouts of the material. Each subject had an individual session, no group sessions were performed.

The evaluation session started with a short introduction of the participants, the procedure, and the objectives of the study. The subjects were explicitly informed that the goal of the evaluation was to assess the utility of the approach and not the capabilities of the participants and that all data would be treated confidentially. First questionnaire q1 was handed out. Next the tasks to be solved were handed out in form of scenarios. Scenario S0 was performed by each participant to promote a free exploration of the system. The time for S0 was limited to 20 minutes. Next, the group-specific scenarios were handed out to the subjects. No time limit was set for task completion. The subjects were encouraged to articulate impressions and problems and think aloud while performing the tasks. After the tasks of all scenarios were completed, questionnaire q2 was handed out to the subject. The evaluation is concluded with a brief free discussion.

Two observers were involved in each evaluation session. One observer recorded the behavioral data, while the other was responsible for writing down comments from the subjects were thinking aloud. During the session, the subjects worked with a laptop with each evaluation lasting of roughly two hours.

4. Results and Findings

The qualitative effects analysis shows that the proposed metrics-based strategy is strongly accepted by the main target group. However the support environment receives higher-than-average ratings from all subject groups.

4.1 Understandability of the approach

The understanding of the metrics collection approach and the metrics support environment by the subjects was measured before and after usage of the support system via the questionnaires q1 and q2. After reading the introductory material, the average percentage of correct answers was 36%. Subsequent to performing the scenario-based tasks, this value doubled, being up to 63%. The performance of the groups and the overall performance of the subjects are depicted in figure 2. It shows that even the relatively short time of usage of the metrics environment led to a significant increase in the understanding of the concepts and components of the approach. The increased understanding of the overall approach was lowest in the group of new employees (NE). However this can be easily explained since their scenarios (NE-S1-S3) did not comprise the usage of all components of the metrics support system.

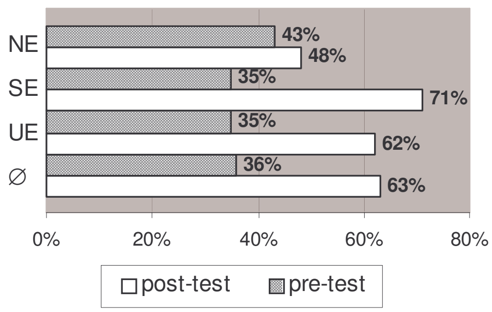

Figure 2: Pre- and post-test scores in knowledge tests with respect to subject groups (in percentage of correctly answered questions)

4.2 Usefulness of the approach

For the qualitative effects analysis, the subjects were asked to assess the properties of the metrics support environments along with the utility dimensions defined in section 3.4Each subject filled out questionnaire q2 after performing the scenario-specific tasks. For this reason questionnaire q2 includes a number of items to be rated on a five-level Likert scale [20] for each dimension. Figure 3 sets out the results of the qualitative effects analysis. The bars represent ratings of the assessment dimensions. The mean ratings were calculated for each dimension and grouped according to the subject groups.

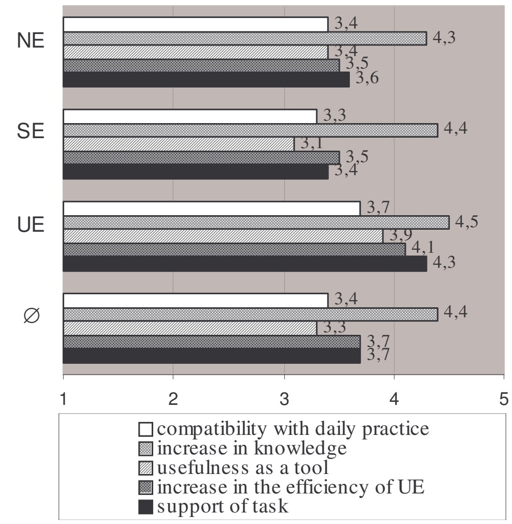

Figure 3: Results of the qualitative effects analysis (assessment scores for utility dimensions with respect to user group, 1: very low, 5: very high)

The results indicate that the approach and its support environment were generally assessed by all groups as higher-than-average useful and reasonable. All subjects seem to highly appreciate the potential of the approach for increasing their knowledge of usability engineering methods. The dimension 'task support' receives the highest scores from the UE group. This corresponds with the pre-configuration of the support environment with usability engineering methods and the subjects role as usability engineers. It could be concluded that the assessment of this dimension by the other groups could be further enhanced, if also usability engineering methods for other areas such as 'requirements engineering' or methods for enhancing programmer productivity were integrated into the method repository of the metrics environment. This result underpins the necessity to provide benefits for all groups of project personnel involved in metrics collection and exploitation.

4.3 Recommendations for usability improvements

The behavioral data and user comments recorded during task performance suggest that there is potential for improving the usability of the metrics support environment. The distribution of usability issues identified by subjects across the client components of the metrics support environment are presented in figure 4.

Most improvement suggestions are related to the components for process guidance and metrics-based decision support. The high number of usability issues identified for the process guidance component can be partially explained by the fact, that the scenarios of all user groups (NE, SE, UE) included interaction with the process guidance component.

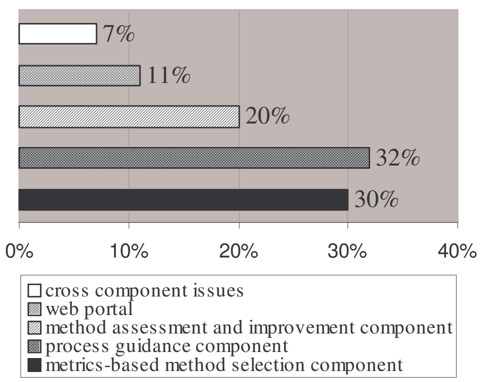

Figure 4: Distribution of the usability issues identified over client components of metrics support environment

One issue is that more assistance in working with the UE methods is appreciated by the subjects. In particular novice users could profit from concepts such as wizards to augment metrics capturing. Moreover the consistency between the applications should be increased. Finally the parts of the terminology used in the graphical user interfaces of the metrics environment should be revised for more comprehensible terms. One example given was that subjects suggested changing the term "project context model" to "project characteristics".

5. A Concluding Remark

In this paper, an approach for adopting UE methods was proposed. It consists of a three-phased process. First, usability engineering methods are selected for a new project based on the projects constraints. Then the project team is supported in deploying the methods in the project. Finally the project assesses the quality of the methods deployed. The approach is supported by a tool, an intranet that offers a web-based support system. The approach has been successfully implemented in industry.

Instead of evaluating the approach in an isolated longitudinal case study, a study was performed to examine the acceptance of the approach by practitioners from various organizations. The acceptance was measured in scenario-driven evaluation sessions, by capturing the understandability, perceived utility and perceived usability of the approach. The results indicate that the approach is accepted by the subject groups examined. We recommend using the study described as a template to estimate the initial acceptance when introducing tool supported measurement programs into organizations. Such studies can be useful for early identification of acceptance problems that hamper the log-term success of metrics programs. The usability of the metrics environment will be further improved using the feedback gathered.

6. Acknowledgements

Part of the empirical study presented in this paper was originally conducted at Daimler Chrysler. We would like to thank Elke Wetzstein and Gerd Jan Tschoepe from the Institute for Industrial Psychology of the Humboldt University Berlin for their support in preparing and conducting the evaluation sessions. Part of this research work was funded by the BMBF (German Ministry for Research and Education) under the project EMBASSI (01IL904I). We would like to thank also the National Science and Engineering Research Council of Canada and Daimler Chrysler for their financial support to the human-centered software engineering group at Concordia University.

References

[1] D. E. Perry, A. A. Porter, and L. G. Votta, "Empirical Studies of Software Engineering: A Roadmap," in Proc. International Conference on Software Engineering (ICSE2000), 2000, pp. 345 - 355.

[2] V. R. Basili, F. Shull, and F. Lanubile, "Building Knowledge Through Families of Experiments," IEEE Transactions on Software Engineering, vol. 25, pp. 456-473, 1999.

[3] D. R. Goldenson, A. Gopal, and T. Mukhopadhyay, "Determinants of Success in Software Measurement Programs: Initial Results," in Proc. Sixth IEEE International Symposium on Software Metrics, 1999, pp. 10-21.

[4] B. G. Silverman, "Software Cost and Productivity Improvements: An Analogical View," IEEE Computer, vol. May 1985, pp. 86-96, 1985.

[5] R. T. Hughes, "Expert Judgment as an Estimating Method," Information and Software Technology, pp. 67-75, 1996.

[6] F. Niessink and H. Van Vliet, "Measurements Should Generate Value, Rather than Data.," in Proc. Sixth IEEE International Symposium on Software Metrics, 1999, pp. 31-39.

[7] O. Laitenberger and H. M. Dreyer, "Evaluating the Usefulness and the Ease of Use of a Web-based Inspection Data Collection Tool," in Proc. Fifth IEEE International Symposium on Software Metrics, 1998.

[8] S. Komi-Sirviö, P. Parviainen, and J. Ronkainen, "Measurement Automation: Methodological Background and Practical Solutions - A Multiple Case Study," in Proc. 7th IEEE International Software Metrics Symposium, 2001, pp. 306-316.

[9] L. Rosenberg and L. Hyatt, "Developing a Successful Metrics Program," in Proc. 8th Annual Software Technology Conference, 1996.

[10] J. Nielsen, Usability Engineering: Morgan Kaufman Publishers, 1994.

[11] L. Briand, K. El Emam, and S. Morasca, "On the Application of Measurement Theory in Software Engineering," Empirical Software Engineering, vol. 1, pp. 61-88, 1996.

[12] B. Minto, The Pyramid Principle - Logic in Writing and Thinking, 3rd ed. London: Minto International Inc., 1987.

[13] A. Birk, T. Dingsøyr, and T. Stålhane, "Postmortem: Never Leave a Project without it," IEEE Software, vol. 19, pp. 43-45, 2002.

[14] F. D. Davis, "A Technology Acceptance Model for Empirically Testing New End-User Information Systems: Theory and Results,," in MIT Sloan School of Management. Cambridge, MA, USA: MIT Sloan School of Management, 1986.

[15] B. A. Kitchenham, "Evaluating Software Engineering Methods and Tools," ACM SIGSoft Software Engineering Notes, pp. 11-15, 1996.

[16] E. Metzker and M. Offergeld, "An Interdisciplinary Approach for Successfully Integrating Human-Centered Design Methods Into Development Processes Practiced by Industrial Software Development Organizations," in Engineering for Human Computer Interaction: 8th IFIP International Conference, EHCI 2001(EHCI'01), Lecture Notes in Computer Science, R. Little and L. Nigay, Eds. Toronto, Canada: Springer, 2001, pp. 21-36.

[17] L. L. Constantine and L. A. D. Lockwood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design: Addison-Wesley, 1999.

[18] D. J. Mayhew, The Usability Engineering Lifecycle: A Practioner's Handbook for User Interface Design: Morgan Kaufman Publishers, 1999.

[19] R. Spencer, "The Streamlined Cognitive Walkthrough Method: Working Around Social Constraints Encountered in a Software Development Company," in Proc. Conference on Human Factors in Computing Systems (CHI'00), 2000, pp. 353-359.

[20] C. M. Judd, E. R. Smith, and L. H. Kidder, Research Methods in Social Relations, 6 ed: Harcourt Brace Jovanovich College Publishers, 1991.

[21] G. F. Smith and S. T. March, "Design and Natural Science Research on Information Technology," Decision Support Systems, vol. 15, pp. 251-266, 1995.

[22] R. D. Galliers and F. F. Land, "Choosing Appropriate Information Systems Research Methodologies," Communications of the ACM, vol. 30, pp. 900-902, 1987.

[23] T. DeMarco and T. Lister, Peopleware: Productive Projects and Teams, 2. ed. New York: Dorset House Publishing, 1999.

[24] Y. Malhotra and D. F. Galletta, "Extending the Technology Acceptance Model to Account for Social Influence: Theoretical Bases and Empirical Validation," in Proc. Thirty-Second Annual Hawaii International Conference on System Sciences, 1999, pp. pp. 1006.

[25] A. Cockburn, Agile Software Development: Addison Wesley Longman, 2001.

[26] N. Juristo and A. M. Moreno, Basics of Software Engineering Experimentation. Dordrecht: Kluwer Academic Publishers, 2001.

[27] Standish Group. "CHAOS Chronicles or CHAOS: A Recipe For Success". 1995.

[28] Bayer, J. and Melone, N. Adoption of Software Engineering Innovations in Organizations. Technical Report CMU/SEI-89-TR-017, Software Engineering Institute, Carnegie Mellon University.

[29] Desmarais M.C., Leclair R., Fiset J.Y., Talbi H. "Cost-Justifying Electronic Performance Support Systems." Communications of the ACM, Vol. 40, No. 7, July 1997.

[30] Howard R. "Software Process Maturity: Measuring Its Impact on Productivity and Quality." IEEE International Software Metrics Symposium, May 1993.