Predicting Acceptance of Software Process Improvement

Medha Umarji
Department of Information Systems
Univ. of Maryland Baltimore County
Baltimore, Maryland, USA.
medha1@umbc.edu

Carolyn Seaman
Department of Information Systems,
Univ. of Maryland Baltimore County
Baltimore, Maryland, USA.
cseaman@umbc.edu

ABSTRACT

Software Process Improvement (SPI) initiatives induce organizational change, by introducing new tools, techniques and work practices. Organizations have to address acceptance issues such as resistance to change, compatibility and fear of adverse consequences. Social psychology literature includes the Technology Acceptance Model (TAM) and the Theory of Planned Behavior (TPB), which study such adoption issues and predict intention to use and actual usage of workplace technology. Some constructs of these models could be applied to software organizations to make it easier for them to counter the initial resistance and to assimilate process improvement into the work culture. To increase applicability of these models to the SPI context, some additional constructs are proposed, by taking into account organizational culture, the impact of changes caused by SPI and the unique characteristics of software developers.

Keywords

Technology acceptance; social psychology; software process improvement.

1. INTRODUCTION

Organizations implement SPI through code inspections and reviews, use of CASE tools [20], new process models such as agile methods, measurement programs and many other initiatives. SPI is very important for a software organization as it helps to reduce redundancies, increase productivity and create a more manageable software process [4]. However it has several acceptance issues because it often involves learning new technology, changes in work practices and an additional workload. Also, SPI involves collecting data about projects, resources and deliverables and often practitioners are not keen on sharing this type of data. Therefore, SPI efforts made by management are often met with a lot of resistance.

Social psychology literature includes the Technology Acceptance Model (TAM) [6], Theory of Planned Behavior (TPB) [1], and Perceived Characteristics of Innovating (PCI) [17], which address the various problems that arise when a new Information System (IS) or technological innovation in the Information Technology (IT) field is introduced in the workplace. These models have received significant attention in the research literature. The TAM, especially, has been applied in many usage scenarios and has performed consistently well.

Introducing an IS is analogous to SPI since the reasons for undertaking these initiatives are similar i.e., budget and schedule considerations, market pressures and continuously evolving industry standards. Also, both are resource-intensive and cause organizational change. The differences between the two are that SPI has a greater impact on the daily work practices and the benefits are more intangible compared to use of an IT innovation.

Therefore, we argue that it is possible to study SPI from a technology acceptance perspective, since most of the issues are similar. However, to predict acceptance of SPI, in a software organization, these models need to be modified to account for the users (i.e., software developers) and the intrusive nature of organizational change. We propose to combine constructs from the TAM and TPB, add extensions like "*fear of adverse consequences*", "*degree of control*" and "*self-efficacy*" and create an SPI acceptance model that will have better predictability in this context.

The sections 2 through 4 describe the IS usage models in detail, address the types of acceptance issues software organizations are faced with and describe the user group i.e., software developers. In section 5, the factors that should be added to existing constructs in order to make them more applicable to the SPI context are discussed. Section 6 describes some ongoing work and ideas for future work and Section 7 presents the conclusion.

2. MODELS FOR IS USAGE

The Technology Acceptance Model (TAM) is a parsimonious and powerful model, based in Social Psychology, for predicting the usage of an Information System (IS). TAM (refer to Figure 1) claims that Ease of Use and Usefulness can predict usage of a system. Attitude and Behavioral Intention act as the mediating variables in TAM. Ease of Use is "the degree to which the … user expects the target systems to be free of effort" [6, p.985]. Usefulness is "the user's subjective probability that using a specific application system will increase his or her job performance within an organizational context" [6, p.985].

Human and Social Factors of Software Engineering (HSSE) May 16, 2005, St. Louis, Missouri, USA
Copyright 2005 ACM ISBN # 1-59593-120-1/05/05

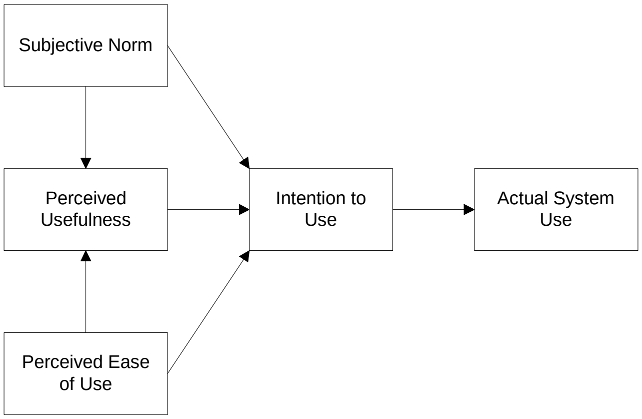

Figure 1 Technology Acceptance Model

Attitude is defined as the user's evaluation of desirability of using the system [27]. In the TAM, Usefulness is influenced by Ease of Use and both have an effect on Attitude. Also, Attitude and Usefulness together influence Intention to use the system. Intention has a direct impact on the actual system usage. TAM has been tested in numerous situations with different types of information systems such as word processing applications, office automation packages, spreadsheet packages and electronic mail and voice mail usages [6], [30].

Over time, TAM has been extended to account for the influence of perceived user resources, affective and cognitive dimensions of attitude, social influence processes (subjective norm, voluntariness and image) and cognitive instrumental processes (job relevance, output quality, result demonstrability and perceived ease of use). All these factors have also been found to significantly influence user acceptance.

The TPB (refer to Figure 2) is a more general theory that has been applied to many domains including IS, to predict actual behavior based on the Behavioral Intention. The TPB suggests that Intention is directly determined by three factors: Attitude toward the behavior, the Subjective Norm and Perceived Behavioral Control.

Previous research has found that when constructs of these or similar models are combined, they show better predictability compared to that of the each model alone [6], [11], [24]. If the model can be made context-specific, its predictive power may improve. This is because each situation in which usage is to be evaluated is different, due to differences in the organizational context, nature of the IS and nature of the people who have to assimilate the changes caused by new technology. Recently, empirical studies have been conducted to test these models in software organizations to determine how well they can predict software developer acceptance of methodologies such as the object-oriented paradigm [11], [26]. Hardgrave et al, [11] could explain 60% of a developer's acceptance of object-oriented systems development, through their model which is a combination of Subjective Norm, Usefulness and Perceived Behavioral control constructs from the TAM and TPB, Thus it follows that combining two or more models can lead to more powerful explanatory models.

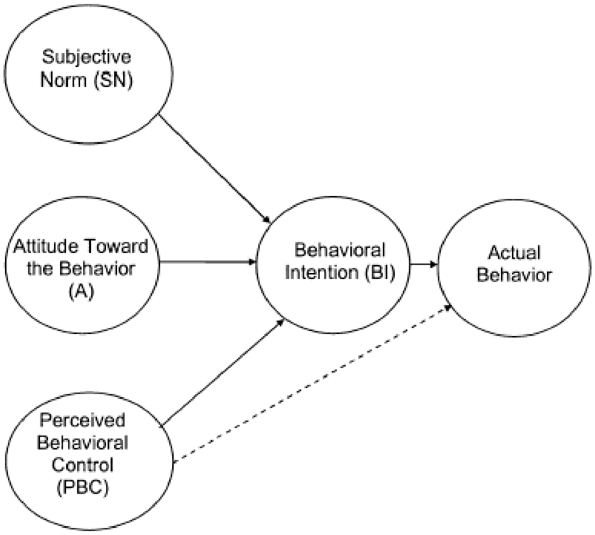
Subjective Norm (SN)
Attitude Toward the Behavior (A)
Behavioral Intention (BI)
Actual Behavior
Perceived Behavioral Control (PBC)

Figure 2 Theory of Planned Behavior

3. ACCEPTANCE ISSUES IN SOFTWARE ORGANIZATIONS

Technology changes rapidly, and software development has to keep up with the latest changes in technology. In addition to this, software organizations have to continuously improve and optimize the software processes to make them more effective and efficient.

Process improvement involves use of new tools and techniques and collecting data about the existing processes, people and products. The next step is analyzing this data, comparing it with pre-set standards and planned values and finding areas of redundancy and inefficiency. These initiatives are met with resistance because schedules may not permit the additional work like code reviews and extra meetings. Also employees tend to assume that the initiatives are a commentary on their performance. Some organizations do not justify the need for introducing SPI to their employees. Employees have a difficult time dealing with the organizational change brought about by these programs.

However, the merit of process improvement cannot be undermined, and taking into consideration that *the life span of 50-80% of these initiatives is only two years* [29], it is necessary to study what can be done to reduce resistance and sustain process improvement in these organizations.

Applying the theory of technology acceptance to process improvement in software organizations seems to suggest itself as a solution to these problems. The predictive capability of these models will help the organization to recognize key areas that need extra focus, so that potential problems with acceptance and adaptation can be foreseen and forestalled. However, it is important to note that TAM, TPB and similar models cannot be applied directly, since they are designed for studying the diffusion and adoption of "technology".

Nevertheless, there are some inherent similarities such as: 1) both (SPI and technology) *induce organizational change*, 2) both are *cost and resource-intensive*, 3) *reasons for introducing them in an* *organization may be similar*. The main differences are that when technology in general is introduced in a workplace, the effects are immediate and visible. For SPI the costs are immediate but the rewards are long-term and many of the benefits are intangible. Also, SPI can be more intrusive and judgmental in nature as compared to a new workplace technology. Therefore, it follows that we have to make changes to these models before we can use them *to study the acceptance of process improvement by software developers in a software organization.*

4. SOFTWARE PERSONNEL

For the purpose of this study, we use the terms software personnel, practitioners and software developers interchangeably to refer to people who are actively involved in developing and maintaining software systems and who participate in project planning, estimation, and management activities.

Software personnel usually have good technical skills and have to learn and apply new technologies quite often. They work under a lot of pressure caused by pressing deadlines and mercurial changes in software specifications. Also, almost all software projects require teamwork and intense meetings and discussions. Communication and inter-personal skills are essential. Also, individual performance tends to get sidelined in a team setting, and this may be frustrating for some people. Programmers also have to be very creative in their work, but within the constraints of time, budget and quality.

The software industry is customer-driven and there is often a lack of alignment between business goals and personal goals. This makes it difficult for the developers to perceive SPI initiatives as useful and makes them skeptical and resistant to new methodologies and work practices. Each developer has to play multiple roles*: that of a learner, an employee, a developer and a team member.* Also, each developer believes that his project is unique and collecting data about it will not help any future projects. The software industry experiences a high rate of staff turnover and this can be stressful to software personnel [5].

These are some of the issues that impact developer thought and account for some of the resistance toward process improvement programs.

5. FACTORS TO CONSIDER

With the same rationale as discussed in the previous sections, we have developed a model to predict acceptance of software metrics programs in particular, as opposed to SPI in general. This model is currently being empirically validated, the results of which will be published soon. However, the context and the user group for the metrics acceptance model are similar to that of an acceptance model for SPI in general. Therefore, we present below the factors that we have included in the metrics acceptance model (in addition to those in the original TAM and TPB models). We believe these factors would be very relevant to an acceptance model for SPI, as support for these factors is also found in the related literature such as adoption of CASE tools. Hence, we discuss each of these factors and their ramifications on the acceptance of SPI in organizations. We divide them into four sub-categories: organizational, personal, SPI-related factors and factors borrowed from social/organizational psychology domain.

5.1 Organizational Issues

5.1.1 Visibility

In software organizations, the corporate goals are not immediately visible, and practitioners tend to take a “narrow, functional view” [20, p.328] of the organizational changes brought about by initiatives such as SPI. Jeffery et al., [19] emphasize that developers should be made aware of the objectives and (tangible or intangible) costs and benefits of measurement programs. We hypothesize that an assessment of visibility of the objectives during the initial phases of an SPI initiative will lend some insight into how well it will be accepted.

5.1.2 Transparency of a process

Transparency facilitates understanding and traceability and consequently, willingness to adapt. Pfleeger [22] discusses the issue of keeping metrics “close” to the developers so that they have access to the measurements, analyses and feedback. Accordingly, we suggest that in order to predict acceptance of process improvement, it is important to gauge how transparent the developers conceive the initiative to be.

5.1.3 Reward structure/incentives

Orlikowski [20] observes that an immediate return on investment is not afforded by CASE tool adoption. Premkumar and Potter [24] also identify that CASE tools have few short-term payoffs and the long-term benefits are often intangible. According to Dekkers [8], one of the secrets of metrics program success is to tie the incentive structure to the metrics program. Hence we adopt the claim that reward structures are part of the organizational policy and if defined and communicated effectively, can aid the motivation to adopt SPI methodologies in general.

5.2 Personal Issues

5.2.1 Fear of adverse consequences

If process improvement activities are not factored into the planned schedule, developers may fear that spending time on these activities will affect their job performance. Also, developers may be hesitant to report inefficiencies of their co-workers and anything that conflicts with the interests of their team and managers. Pfleeger [22] presents a set of lessons learned for metrics programs and asserts that practitioners felt threatened by metrics because metrics can be used as a commentary on their performance; and so, if developers do not want to collect metrics, it is risky to make them do so. Also, it is recommended that the product or the process should be criticized not the people. Jeffery et al., [19] address the concern that metrics should not be used to assess individuals, as this can spark doubt and uncertainty about the usefulness of such initiatives. Based on this discussion, we gather that unless these apprehensions are addressed at the start of the implementation, they would eventually affect the assimilation process.

5.2.2 Communication

Organizations should have well-defined channels of communication, so that all policies can be communicated clearly and effectively. All the studies conducted on metrics programs and CASE tools have highlighted this [8], [10], [14], [19], [20], [24], [25]. Also, it is essential for developers to have good interpersonal and communication skills. Developers usually are technical people who may or may not have very good leadership and management skills. Chilton et al., [5] have created a behavioral rating scale for IT personnel and this can help to gauge the communication effectiveness of software practitioners as well. Communication in an organization can be a very good determinant of whether SPI or other initiatives can function effectively and be successful.

5.2.3 Self-efficacy

Self-efficacy is defined formally as "the belief in one's capabilities to organize and execute the sources of action required to manage prospective situations" [3]. It relates to an individual's assessment of competence to perform a specific task in a given domain based on his or her experience, knowledge, opinions of colleagues and physiological conditions Therefore, we hypothesize that a measure of developer self-efficacy would be a good indicator of the capability to perform SPI activities. Self-efficacy is also a significant determinant of Intention to perform a task [27].

5.2.4 Degree of control

Control signifies the extent to which a developer can make suggestions for changes and become involved in SPI activities. The control that a developer has over planning and implementation of SPI can impact developer self-efficacy and also adoption. All the studies on metrics programs prescribe that developers should be actively involved in analyzing data, making decisions for improvement and implementing the changes [15], [19].

5.3 SPI-related Issues

5.3.1 Amount of learning required

Some of the SPI techniques necessitate use of tools, e.g., tools for diagramming, procedural code generation and reverse engineering. Rai et al.[25], posit that providing adequate training can cultivate the right skills and reduce the knowledge barriers that impede acceptance of new technology. Jeffery et al.,[14] also support this claim. In the case of metrics programs, Pfleeger [22] however, states that "developers need not become experts in measurement theory, statistics or other techniques", it is important to keep everything simple and easy to understand. We believe that a judicious amount of training should be provided so that the migration path is easy for practitioners.

5.3.2 Compatibility of work practices

SPI activities should be technically as well as behaviorally compatible with the existing work practices. Rogers [28] defines compatibility of an IT innovation as "degree to which an innovation is perceived as being consistent with the existing values, past experiences and needs of the potential adopter." Rai et al.[25] express the concern that changes caused by CASE tools may alter the political and social dynamics of the workplace, and cause resistance; therefore the methodology should be compatible with existing practices. Based on the literature on metrics programs and innovation adoption we postulate that compatibility of a process or tool is tied to its smooth acceptance.

5.3.3 Champions/ advocates

Marketing the initiative within the company is a good strategy to reduce resistance of practitioners. Premkumar et al., [24] stressed the importance of having a product champion explain the needs and benefits of CASE tools. At Contel Corporation [22], metrics advocates were groomed to spread success stories to motivate people, a technique that proved very successful. Jeffery et al.,[14] also discuss the promotion of a metrics program though publication of success stories, in their measurement success factor framework. Based on this, we generalize that advocates/champions increase the likelihood of buy-in at the practitioner level.

5.4 Factors borrowed from Social Psychology

The other factors that we have borrowed from the existing models are proven to be strong determinants of Intention to use and actual usage. We interpret them in the SPI context and provide a brief description below.

5.4.1 Perceived Usefulness

The Usefulness construct from TAM can be viewed from two perspectives: organizational Usefulness and personal Usefulness. Organizational Usefulness is the developer's perception of how their involvement with SPI will be beneficial to the organization (and indirectly to the developer). Personal Usefulness is the developer's perception of how performing SPI activities will enhance their job satisfaction and career prospects. We adopt the claim that Usefulness will be a significant determinant of Intention to perform SPI activities.

5.4.2 Attitude

Attitude (borrowed from TPB) is an assessment of the desirability of performing a behavior, by an individual [1], [2]. Attitude has two sub-components: affective (e.g., happy-sad) and cognitive (e.g., beneficial-harmful). Fenton and Hall discuss practitioner attitude in detail and claim that "if you fail to generate positive feelings towards the program, you seriously undermine your likelihood of success." Attitude is also influenced by job seniority, integrity of metrics data and reality of the metrics programs [10].

5.4.3 Perceived behavioral control

This is borrowed from the TPB and refers to both personal and situational impediments to performing a behavior. It addresses the situation where performance may depend on non-motivational factors such as availability of requisite opportunities and resources [2]. PBC includes both external and internal factors. External factors are those things outside one's person that may affect behavior; therefore, we include here the user resources such as relevant documentation, availability of a help facility, reliable and experienced personnel, adequate time and financial stability. Internal factors refer to the personal ease or difficulty of performing the behavior, based on one's prior experiences, skills gained through learning, and intelligence. PBC also has an impact on the overall feeling of control and will eventually affect the Intention to use/actual usage of the system [2], [3].

5.4.4 Subjective Norm

Subjective Norm is defined in the TPB, and suggests that social pressure can influence an individual's intention to use a system [27]. Also, Subjective Norm, which has to do with the attitudes of a group of co-workers, plays a significant role in SPI programs. If one developer feels that the SPI is not effective or worthwhile, he or she may influence people around and this attitude begins to spread through the group. This would potentially cause a slack in the general energy and interest. The descriptive norm is one component of Subjective Norm; it is the notion that if people within the social network of a subject perform a certain behavior, then that subject's motivation to perform that behavior is positively enhanced. Injunctive norm is the other component of Subjective Norm, and it is the notion that if people within the social network of a person *want* him/her to perform a certain behavior that they consider to be useful, eventually his/her motivation for performing that behavior will be positively influenced. Also, people tend to believe their managers, their superiors, and experienced colleagues more than theory or books [27].

5.4.5 Ease-of-Use

Ease of Use refers to the level of complexity of SPI tools and activities and ease with which developers can adapt to changes in work practices caused by SPI and use the tool or set of tools required in the implementation of SPI. All the studies on metrics and CASE tools strongly support the claim that Ease-of-Use facilitates acceptance of the SPI technique.

6. PROPOSED METHODOLOGY

All the factors described above are either borrowed from the literature on social psychology or are derived from anecdotal evidence and case studies in the SPI literature. The inter-relationships between these factors and their individual and combined effect on the Intention to perform SPI activities, needs to be investigated through empirical research. The ultimate aim of this research is to predict the likelihood of acceptance of an SPI initiative and to highlight areas that may need improvement. Therefore, we propose to create a predictive model that will take as input all the factors discussed above and relate them to the probability of success of an SPI effort. From the model, an instrument (e.g. a survey) could be designed to collect data to evaluate the factors.

The sample for validating such an instrument would consist of software practitioners across more than one organization that is in the initial phases of implementing an SPI technique. Test items (e.g. survey questions) could be designed to measure each of the factors. For factors relating to technology acceptance, test items can be borrowed from the literature as they have been tried and tested. However, we would have to design the items to test the other factors.

The collected data could be analyzed by exploratory factor analysis so that the factors that are highly correlated can be reduced to constructs that together determine the dependent variable, Intention. This model will then have to be tested and validated so that it can serve as a tool for future implementations of SPI.

As an initial step in this line of research, we have developed a metrics acceptance model, along the lines of the model proposed above for SPI initiatives, for predicting the acceptance success of a software metrics program. The metrics acceptance model is currently being validated. Early results are promising and are forthcoming in the literature.

7. CONCLUSIONS

These factors capture some of the organizational and personal issues and possible ways in which developers can perceive SPI. This knowledge, when added to the model created by combining constructs from the TAM and TPB as described above, will help in predicting specifically what could go wrong and what areas the organization should focus on while implementing an SPI technique. This will ensure that the large investments in terms of time, resources and cost made in such efforts are not wasted. Empirical validation of this model could be done in a longitudinal study, using the survey methodology.

REFERENCES

[1] Ajzen, I. The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50,* (1991), 179-211.

[2] Ajzen, I. Perceived behavioral control, self-efficacy, locus of control, and the theory of planned behavior. *Journal of Applied Social Psychology*, *32,* (2002), 665-683.

[3] Bandura, A. Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review*, *84,* (1977), 191-215.

[4] Briand, L.C., Differding, C.M., and Rombach, H.D. Practical guidelines for measurement-based process improvement. *Software Process - Improvement and Practice*, 2, (1996), 253-280.

[5] Chilton, M. and Hardgrave, B. Assessing information technology personnel: toward a behavioral rating scale. *DATA BASE Adv. Inform. Syst., 35, 3,* (2004), 88-104.

[6] Davis, F. Perceived usefulness, perceived ease of use, and end user acceptance of information technology. *MIS Quarterly*, *13,* (1989), 318-339.

[7] Dasakalantonakis, M. A practical view of software measurement and implementation experiences within Motorola." *IEEE Trans. Software Eng.*., 18, (Nov. 1992), 998-1010.

[8] Dekkers, C.A. The secrets of highly successful measurement programs. *Cutter IT Journal, 12, 4,* (1999), 29-35.

[9] Gopal, A., Krishnan, M.S., Mukhopadhyay, T., and Goldenson, D. Measurement programs in software development: determinants of success. *IEEE Trans. on Software Eng.*, *28, 9,* (Sep. 2002), 863-875.

[10] Hall, T. and Fenton, N. Implementing effective software metrics programs. *IEEE Software*, (March/April 1997), 55-65.

[11] Hardgrave, B. and Johnson, R. Toward an information systems development acceptance model: the case of object-oriented systems development. *IEEE Trans. On Eng. Management, 50, 3,* (Aug. 2003), 322-336.

[12] Hardgrave, B., Davis, F., and Riemenscheneider, C., Investigating determinants of software developers' intentions to follow methodologies. *J. Manage. Inform. Syst.,20,* (2003), 123-151.

[13] Iversen, J., and Mathiassen, L. Lessons from implementing a software metrics program. In *Proc. Hawaii Int’l Conf. System Sciences (HICSS-33)*, 2000.

[14] Jeffery, D.R. and Berry, M. A framework for evaluation and prediction of metrics program success,” In *Proc. 1st Int’l Software Metrics Symp.* (Los Alamitos, Calif., 1993).IEEE Computer Society Press, 1993, 28-39.

[15] Jeffery, D.R., Basili, V., and Berry, M. Establishing successful measurement for software quality improvement. In *Proc. TC8AUS IFIP Int’l Working Conf*. (North-Holland, Amsterdam, 1994), 339-350.

[16] Mathieson, K. Predicting user intentions: comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research*, *2*, (1991), 173-191.

[17] Moore, G., and Benbasat, I. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, *2*, (1991), 192-222.

[18] Niessink, F., and van Vilet, H. Towards mature measurement programs. In *Proc. Second Euromicro Conf. on Software Maintenance and Reengineering.* (Florence, Italy, March 8-11, 1998). IEEE Computer Society, 82-88.

[19] Offen, R.J. and Jeffrey, R. Establishing software measurement programs. *IEEE Software*, (Mar./Apr. 1997), 45-53.

[20] Orlikowski, W. J. CASE tools as organizational change: investigating incremental and radical changes in systems development. *MIS Quarterly*, *17, 3,* (1993), 309–340.

[21] Paulish, D., and Carleton, A. Case studies of software process improvement measurement. *IEEE Computer*, 27, 9, (Sep. 1994), 50-57.

[22] S. L. Pfleeger, “Lessons Learned in Building a Corporate Metrics Program,” *IEEE Software*, pp. 67-74, May 1993.

[23] Pfleeger, S. L. Understanding and improving technology transfer in software engineering. *J. Systems and Software*, *47, 2-3,* (Jul. 1999), 111-124.

[24] Premkumar, G., and Potter, M. Adoption of computer aided software engineering (CASE) technology: An innovation adoption perspective. *DATA BASE Adv. Inform. Syst.,26, 2&3,* (1995), 105-123.

[25] Rai, A., and Patnayakuni, R. A structural model for CASE adoption behavior. *J. Manage. Inform. Syst*., *13, .2,* (1996), 205-234.

[26] Reimenschneider, C., Hardgrave, B., and Davis, F. D. Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Trans. on Software Eng*., *28, 12*, (Dec. 2002), 1135-1145.

[27] Rhodes, R., and Courneya, K. Investigating multiple components of attitude, subjective norm, and perceived control: An examination of the theory of planned behavior in the exercise domain. *British Journal of Social Psychology*, *42,* (2003), 129-146.

[28] Rogers, E. *Diffusion of Innovation.* 4th ed. Free Press, New York, 1995.

[29] Rubin, H. Measuring 'Rigor' and putting measurement into action. *American Programmer*, 4, 9, (Sep. 1991), 9-23.

[30] Venkatesh, V., and Davis, F. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Manage. Sci*., *46, 2, (2000),* 186–204.