IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB12846

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Cultural and Political Issues in Implementing Software Process Improvement

Dana Edberg & Lisa A. Anderson
University of Nevada, Reno, Mail Stop 026, Dept of ACC/IS, Reno, NV 89557, P 775-784-6179, F 775-784-8044, dte@unr.edu

INTRODUCTION

Software process improvement (SPI) initiatives encompass a broad range of potential changes to the software development and maintenance process. An SPI initiative can be as simple as implementing a new package for software version control or as complex as changing the way developers work with system stakeholders to identify necessary software requirements. While the scope of the actual initiative may vary, the key issue is that SPI means *change* of some kind to the way people conduct the work of software development and maintenance. Changing people's work habits is frequently perilous, as has been confirmed by the reported 70% failure rate of SPI projects (SEI, 2002). Software development organizations continue to implement SPI with the hope that such initiatives will reduce defects, improve overall productivity, lower costs and enhance the repeatability of project results (Humphrey, 1998).

While current research has extensively explored the reasons for success and failure of SPI through case studies, observation and surveys of project participants, research has been unable to produce a clear formula for successful SPI implementation. What has become clear is that managing SPI is similar to managing any other form of profound organizational change and requires broader understanding of the effects of organizational culture and politics on that change. According to Peter Senge et al. (1999), if we believe that organizations are living organisms capable of learning and growth, then we should stop asking why people resist change and instead attempt to understand the natural response of the enterprise, given its traditions, norms and assumptions. This paper explores the cultural and political issues affecting work process change within a software development enterprise.

BACKGROUND

Much research has been performed to describe and understand the reasons for the failure of SPI initiatives. This section of the paper describes the current research into software process improvement initiatives to understand the motivation for our study.

Why Do Organizations Undertake SPI Initiatives?

Organizations undertake SPI initiatives for a broad set of reasons that can be divided into three basic categories: To address a particular problem in the results of the process (e.g. too many software defects, low personnel productivity, incorrect specifications); to gain outside validation of the development process (CMM or ISO certification); or to align more fully with a recently acquired (or acquiring) organization's processes (Baddoo and Hall, 2002).

What Motivates Software Practitioners During SPI?

Research indicates that the primary reason people are willing to change software processes is evidence of the benefits of the change (Baddoo and Hall, 2003). Other reasons are dependent upon the role of the person within the organization but include such factors as (Baddoo and Hall, 2002):

- Visible senior management support, input into the type of process changes, and sufficient resources available to maintain the initiative (for software developers);
- Ability to make decisions about the changing processes, sufficient resources available to maintain the initiative, and ability to change processes as needed (for project managers); and
- Ability to meet financial/managerial targets more effectively and relative cost benefits of the initiatives (for senior managers).

A key issue with the reasons above is that most are dependent upon clear evidence of the success of SPI. Unfortunately, research has shown that few organizations have historical metrics to use for comparison or a reliable new metrics program in place to evaluate the effectiveness of SPI initiatives (Iversen and Mathiassen, 2003). In addition, it has been demonstrated that process changes may require several evolutionary iterations to become successful (Borjesson and Mathiassen, 2004) implying that some organizations abandon the initiative before it has a chance to become successful.

Why Do Software Process Improvement Projects Fail?

Through case studies and surveys (Borjesson and Mathiassen, 2004; Baddoo and Hall, 2003; Humphrey, 1998; Humphrey, 2002; Iversen, et. al, 2004), research has shown the following factors are related to the failure of SPI initiatives: Lack of a fully-defined problem to be solved through SPI; lack of management support; mismatched accountability measures among levels of project participants; lack of acceptance by software professionals; lack of sufficient resources available to the project; lack of ongoing support and management commitment to the changes; miscommunication among project participants; and poor and/or incomplete change management processes.

Most of the factors above relate to the management of change and are common to comprehensive new policies or procedures introduced in an organization. The factors themselves do not provide great insight into how to help an organization sustain the type of profound process change triggered through SPI.

What Are The Prescriptions for SPI Success?

Studies of both successful and unsuccessful SPI initiatives have shown some patterns for success, but none with guaranteed steps to follow for every project. Some of the dominant prescriptions for success include (Kim, 2004; Borjesson and Mathiassen, 2004; Rainer and Hall 2003):

- Establish a separate process management group with authority and responsibility to create and implement process changes.
- Define and communicate the problem to be solved through the SPI initiative.
- Gain commitment from key stakeholders.
- Be prepared to begin slowly and implement changes with a phased approach.
- Publish evidence of SPI success.

At the same time, however, research into organizational change has shown that even following the advice above it may not be possible to achieve sustained change in an organization (Senge et al., 1999).

Is SPI-Induced Change Different From Other Organizational Changes?

None of the above findings is substantially different from those discovered during any other form of change seeking to modify an individual's work methods. Studies have documented the continuing failure of change initiatives in organizations, attributing the problems to issues very similar to those found in the SPI literature (Senge et al., 1999; Argyris and Schon, 1996).

Much of the current research has studied the personal and organizational changes produced from the implementation of SPI initiatives as if it were separate from other types of change in organizations. Very little of the current research references existing literature on organizational change. Two notable exceptions are a study that evaluates the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) from an organizational culture viewpoint (Ngwenyama and Nielsen, 2003) and a study that uses potential independent variables from the organizational learning and change literature to create a model of the key factors correlated with success in software process improvement (Dyba, 2005).

By not including the current literature on organizational change, the existing studies of SPI neglect the effects of organizational culture and political interactions on the potential success of an SPI initiative. We posit that the basic culture of a software development organization can provide insight into the relative success or failure of an SPI initiative. Culture can be likened to the "personality" of an organization. Culture is the tacit, shared and coherent understandings about who and what matters; how, what and why things get done (Schein, 1985).

It is possible that the culture of a software development organization affects the way that major organizational change is implemented. For instance, the managerial culture of a software development organization may encourage greater collaboration from participants at all levels of the organization (Ngwenyama and Nielsen, 2003). This managerial culture would make it difficult to enforce substantial change to the work processes of employees. Our study explores this and other organizational change issues by examining the culture and political interactions of two divisions of a company undergoing SPI initiatives.

DESCRIPTION OF THE STUDY

The objective of this study is to develop theory grounded in empirical evidence. The research design for this study is what Yin (1994) terms an "embedded multiple case" design. The primary unit of analysis for this study was the division. Embedded within each division was a set of three groups. A group consisted of people with the same (or very similar) job responsibilities. The groups were the same across the two cases. We studied three groups for each of the two cases. The groups were Software Engineers, Project Managers and Senior Managers.

Two divisions served as the cases for this project. They were divisions of the same company, IT Develop Co., which provides software and services for business management, has annual revenue of $1 billion and employs about 7,000 people. Division A serves as an application service provider for payroll applications for small to medium businesses. It employs approximately 700 people. The software engineers in Division A produce software to support the application service business. The customers for the software engineers are ultimately the clients who purchase the service, but the software engineers actually work with business analysts who then work directly with the customers. Division B produces shrink-wrapped software for accountants. It employs approximately 600 people. The software engineers in Division B work with marketing analysts to produce software suitable for the final end-users who are accounting firms.

Both divisions had implemented SPI initiatives in the year prior to this study. Division A had implemented a new version control system for software development and had started using Fagan inspections for documenting initial specifications and testing. Division B had implemented a version control system for testing (different system than Division A), Fagan inspections (similar to Division A), and a workload balancing system. Both divisions were contemplating additional SPI initiatives during our study, but the focus for data collection was on the initiatives already implemented. Management told us before beginning the study that Division A had implemented the initiatives partially, while Division B had completely implemented all SPI initiatives. We were also told that the company considered the initiatives at both divisions to be a success.

Table 1. Focus group participants

	Software Engineers	Project Managers	Senior Managers
Quantity in focus group	Division A: 7 Division B: 9 (two groups)	Division A: 4 Division B: 6	Division A: 6 Division B: 3
Job Responsibilities	Requirements definition with clients, design, coding, testing, documentation	Project planning, risk assessment, estimation, control, management, reporting	Strategic planning, project definition, initiation, overall management
Range of Experience	5 – 30 years	3 – 16 years	3 – 20 years
Range of Educational Level	no college to masters degree	bachelors to masters degree	bachelors to masters degree

Data Collection and Analysis

Data were collected through interviews and study of documentation. Face-to-face, semi-structured, focus group interviews formed the primary method of data gathering. Table 1 summarizes the size and demographic information for the focus groups.

FINDINGS AND ANALYSIS

Each separate focus group was asked to discuss explicitly the topics shown in Table 2. Many of the results shown in Table 2 are in alignment with other studies of SPI. Each of the groups had a clear reason for implementing SPI and those reasons were the same across the three groups within a given division. The reasons given for the success and/or failure of SPI were also similar to existing studies.

The results that were striking became clear during the analysis of Division A. In Division A, the three groups had substantially different views of the relative success of SPI. As shown in Table 2, the Software Engineers (SE's) had abandoned the new processes three months before our study began, but the Project Managers (PM's) and Senior Managers (SM's) believed that the initiatives were a resounding success. We explored this finding again in Division B. We found a similar, but less striking result. The SE's in Division B felt that some of the initiatives were successful and others were not. These SE's were unable to identify any actual changes to their work processes, but they still felt that some were successful. They defined success in this case as "not having to work overtime during the peak period. Work is now spread more evenly over the whole year." We found the same perception with the PM's in Division B. Unlike the PM's in Division A, the PM's in Division B were aware that some of the initiatives were not successful. However, the SM's in Division B were oblivious to the level of implementation of SPI; they felt that the initiatives were completely implemented, should be considered a success and looked forward to additional SPI's.

We found substantial similarities among the three groups between the two divisions. The SE's in Division A had the same attitudes and issues as those expressed in Division B. The SM's in the two divisions also shared ideas. The PM's were a little different between the two divisions. The PM's in Division A were very optimistic about SPI, felt that it was a great success and believed that increased customer satisfaction was due to new processes (processes abandoned by the SE's in Division A). The PM's in Division B, on the other hand, were far more aware of the actual status of implementation and were actively discussing among themselves ways to customize SPI so that it would work better within their division. There were far more similarities among the people in the three groups, however, than there were between the divisions. Thus, the

Table 2. Summarization of results from groups in divisions A and B

Topic	Div.	Software Engineers	Project Managers	Senior Managers
Reasons for implementing SPI	A	Need to better capture user requirements. Need to more completely test products before releasing to customers.	Need to better satisfy user requirements. Newly acquired by a company that required more defined processes.	Need to improve existing processes to better satisfy customer demand. General improvement of development efforts for better cost control.
	B	Loss of market share due to defective software. Newly acquired by a company that required more defined processes. Balance workload. Avoid overtime.	Loss of market share due to defective software. Newly acquired by a company that required more defined processes. Balance workload. Avoid overtime.	Opportunity to enhance productivity. Loss of market share due to defective software.
Personal role in SPI	A	Learn the process.	Encourage SE's to use the process Create method to evaluate use.	Identify coalition of willing participants. Communicate. Create method to evaluate use
	B	Learn the process. Figure out how to incorporate process into work habits	Decide what aspects of a given SPI might benefit a particular project team.	Decide what processes need to be improved. Choose appropriate process initiative. Communicate
Perception of relative success of SPI	A	Failed; new processes abandoned.	Successful; new processes are being used.	Very successful.
	B	Some successful, most abandoned.	Some successful, most abandoned.	Very successful.
Reasons for success and/or failure of SPI	A	No support from management. Too much required documentation.	Processes help elicit customer requirements. Processes enhance testing methods.	Good communication.
	B	No support from management.	No clear definition of process implementation requirements.	Good communication. Good SPI choice.
Signs of improvement from SPI	A	Used some process suggestions to improve personal testing methods.	New processes have increased client satisfaction with software.	Fewer complaints from clients about their satisfaction with software.
	B	Used some process suggestions to improve personal development methods.	Fewer defects in software.	Slight increase in market share. Fewer defects in software.

findings of the cultural and political interactions will focus on the three groups, rather than the divisions.

INDUCED FINDINGS FROM INTERVIEW DISCUSSIONS

We did not ask direct questions about the culture and political interactions of the divisions. Past research has shown that individuals may answer such questions with their "desired" beliefs rather than their "actual" attitudes and actions (Argyris and Schon, 1996). Based on the discussions with the groups and a review of documentation about the SPI initiatives we derived the set of questions and answers shown in Table 3. These questions and answers show clear differences among the three groups in their attitudes towards personal responsibility for SPI. For example, SE's were not sure who chose a particular SPI initiative (such as Fagan inspections), while PM's and SM's viewed it as a responsibility of senior management. SE's were also unsure why the organization wanted to generally make changes in the software development process, but the PM's and SM's had clear views of the need for SPI. Most interesting in Table 3 are the differences in perception of the responsibility for enforcing SPI changes. Both SE's and PM's felt strongly that senior management was responsible for devising a way to enforce the use of SPI, while SM's felt that mandated use of SPI was contrary to the structure of the organization. The SM's believed that they hired professionals (SE's and PM's) who were capable of deciding their own optimal work methods. If SE's and PM's chose not to use certain SPI initiatives, then senior management would not force them. On the other hand, both the SE's and PM's thought that their use of the SPI processes should be part of their evaluation.

Table 4 displays findings about the culture and political interactions of the groups. We were especially interested in discovering the communication methods and management structure perceptions. As shown in Table 4, there is agreement among the groups about the use of informal communication about work activities. The groups diverged when asked

Table 3. Findings from groups about SPI cultural and political issues

SPI Issue	Software Engineers	Project Managers	Senior Managers
Who is responsible for choosing a particular SPI initiative?	Mixed responses, but essentially didn't know.	Senior management and/or central software engineering process group.	Senior management.
Why are SPI initiatives pursued by the organization?	Mixed responses, but essentially didn't know.	To solve a particular problem.	To generally produce a better product and satisfy customer requirements with greater cost control.
Who is responsible for enforcing SPI changes?	Senior management.	Senior management.	No enforcement – process use should should be the choice of the participants.
Who is responsible for customizing SPI changes?	Senior management.	Project management and/or central software engineering process group.	Project management and/or central software engineering process group.
How is the need for SPI communicated?	General meetings talking about the problems in the organization.	Project meetings and email.	General meetings.
How are SPI initiatives introduced?	Through initial training classes. Don't find out until classes are scheduled.	Project meetings. Don't always know when an initiative is introduced.	General meetings and announcements.
How are SPI changes incorporated into work activities?	Personal choice. Select those changes that make the most sense and discard the rest.	This is a problem. Changes need to be better customized for a given project or situation.	Mixed responses, but essentially didn't know.

Table 4. Findings from groups about organizational cultural and political issues

Organizational Issue	Software Engineers	Project Managers	Senior Managers
How do people generally communicate about activities?	Informally – small meetings, lunch conversations, hall talk	Informally – small meetings, lunch conversations, hall talk	Informally – small meetings, lunch conversations, hall talk
How is bad news communicated to management?	Try not to do it. Just allow the news to become "known."	Try not to do it. Small meetings if absolutely necessary.	Open to all news. Open door policy and skip-level meetings.
How does management react to bad news?	Kill the messenger.	Kill the messenger	Positively – all news is important to open communication.
What type of managerial structure is used?	Somewhat hierarchical – management decides what to do and SEs implement the plans.	Collaborative – management and engineers work together to decide on projects and implementation.	Collaborative – management sets the agenda in collaboration with other company stakeholders, including SEs and PMs.
Who sets estimated dates for projects?	Management. Sometimes the estimates are very optimistic.	Management. Estimates are optimistic.	Management in collaboration with other company stakeholders, including SEs and PMs. Estimates reflect all input.

about the communication of bad news. In the focus groups, the SE's uniformly hunched over and expressed through body language their unwillingness to deliver bad news. They stated that they "don't say anything" in those situations. The perception was that management would meet bad news with either disbelief or disapproval. The PM's agreed with this perception. The SM's sharply contradicted this perception. They believed that they welcomed all news and provided many methods for open communication. This difference in perception made it clear why the SE's in Division A had not told management that they had abandoned the new SPI processes.

Study Propositions

As prior research has shown, and as our own interviews have corroborated, affecting long-term significant change is a complex and difficult undertaking with no guarantee of success. The collaborative culture that exists in professional organizations makes any kind of dictatorial change unwelcome and hard to maintain. In the software industry, engineers are hired for their skills, experience and ability to think for themselves. Our data showed that engineers want to know what needs

to be done, not how to do it, and management is willing to allow them to accept or reject new processes as they see fit. While collaborative management may be the norm in the software industry, it does not support SPI induced change. Thus, collaborative management makes it difficult to sustain profound change throughout an organization unless all participants are willing to make and retain the changes. This finding leads us to propose:

Proposition 1: Senior management should be willing to enforce employee adherence to the new processes to sustain SPI induced change.

Proposition 2: All participants should determine how changes are made to their personal work processes and agree to make those changes to sustain SPI induced change.

Another issue that may be unique to the culture of software development organizations is the "engineering" mindset. Engineers prefer to have rational reasons to change. They want to have evidence and analysis performed before they are willing to invest the time and effort to change. To create this level of evidence requires metrics both before and after the change has occurred. This may be especially problematic for organizations that have not gathered metrics before change. To use only metrics after the change lessens the likelihood that a person with a rational mindset will be willing to accept the efficacy of the process change. This observation leads us to propose:

Proposition 3: SPI induced change is best sustained in organizations that documented key metrics before and after the process change.

A third issue that is important in the culture of a software development organization is the level and type of communication that occurs. Engineers tend to shy away from substantial, detailed communication. They do not tend to discuss their work habits or processes with others. The lack of easy communication among members was clear in the interviews with the software engineers – few had discussed the process changes and it was very difficult to gather information about their level of understanding of the differing initiatives. While the engineers were very willing to discuss the company, they did not want to discuss individual work processes or whether they fully understood a given initiative. This finding leads us to our final proposition:

Proposition 4: Personal, indirect communication methods, such as email, will provide more information for SPI participants than meetings or general announcements.

CONCLUSION

We remain concerned about the long-term sustainability of SPI initiatives. Past work and our research has shown that such initiatives may not be aligned with the management style of a software development organization. We have also corroborated the necessity identified in prior research for open and candid communication among project participants of actual performance during and results from a SPI initiative. We believe that SPI-induced changes will be sustained only when they are accepted by software engineers and incorporated into their personal work processes.

REFERENCES

Argyris, C. and Schon, D.A. (1996) Organizational Learning II, Reading Massachusetts: Addison-Wesley.

Baddoo, N. and Hall, T. (2002)*"Motivators for Software Process Improvement: An Analysis of Practitioner's Views,"* The Journal of Systems and Software, Vol. 62, 85-96.

Baddoo, N. and Hall, T. (2003)*"De-Motivators for Software Process Improvement: An Analysis of Practitioner's Views,"* The Journal of Systems and Software, Vol. 66, 23-33.

Borjesson, A. and Mathiassen, L. (2004) *"Successful Process Implementation,"* IEEE Software, July/August, 36-40.

Dyba, T. (2005) "An Empirical Investigation of the Key Factors for Success in Software Process Improvement," IEEE Transactions on Software Engineering, Vol. 31, No. 5, 410-424.

Humphrey, W.S. (1998) *"Why Don't They Practice What we Preach?,"* Annals of Software Engineering, Vol. 6, 201-222.

Humphrey, W.S. (2002) *"Three Process Perspectives: Organizations, Teams, and People,"* Annals of Software Engineering, Vol. 14, 39-72.

Iverson, J. and Mathiassen, L. (2003) *"Cultivation and Engineering of a Software Metrics Program,"* Information Systems Journal, Vol. 13, 3-19.

Iverson, J., Mathiassen, L., Nielsen, P. (2004) *"Managing Risk in Software Process Improvement: An Action Research Approach,"* MIS Quarterly, Vol. 28, 395-433.

Kim, H. (2004) *"A Process Model for Successful CRM System Development,"* IEEE Software, July/August, 22-28.

Ngwenyama, O. and Nielsen, P.A. (2003)*"Competing Values in Software Process Improvement: An Assumption Analysis of CMM from an Organizational Culture Perspective,"* IEEE Transactions on Engineering Management, Vol. 50, February, 100-112.

Rainer, A. and Hall, T. (2003)*"A Quantitative and Qualitative Analysis of Factors Affecting Software Processes,"* The Journal of Systems and Software, Vol. 66, 7-21.

Schein, E. (1985) Organizational Culture and Leadership. San Franciso : Jossey-Bass Publishers.

SEI (2002), *"Process Maturity Profile of the Software Community,"* White Paper, Software Engineering Institute, Carnegie-Mellon University.

Senge, P., Roberts, C., Ross, R., and Smith, B. (1999) The Dance of Change. New York: Random House, Inc.

Yin, R.K. (1994) Case Study Research, Sage Publications - Applied Social Research Methods Series.