



OpenAIR@RGU

The Open Access Institutional Repository at Robert Gordon University

<http://openair.rgu.ac.uk>

This is an author produced version of a paper published in

Proceedings Seventh International Conference on the Quality of Information and Communications Technology (ISBN 9781424485390)

This version may not include final proof corrections and does not include published layout or pagination.

Citation Details

Citation for the version of the work held in 'OpenAIR@RGU':

ALLISON, I., 2010. Organizational factors shaping software process improvement in small-medium sized software teams: a multi-case analysis. Available from <i>OpenAIR@RGU</i>. [online]. Available from: http://openair.rgu.ac.uk

Citation for the publisher's version:

ALLISON, I., 2010. Organizational factors shaping software process improvement in small-medium sized software teams: a multi-case analysis. In: In: F. B. E ABREU, J. P. DE FARIA and R. J. MACHADO, eds. Proceedings Seventh International Conference on the Quality of Information and Communications Technology, 29 September – 2 October 2010. Los Alamitos, CA: IEEE. Pp. 418-423.
--

Copyright

Items in 'OpenAIR@RGU', Robert Gordon University Open Access Institutional Repository, are protected by copyright and intellectual property law. If you believe that any material held in 'OpenAIR@RGU' infringes copyright, please contact openair-help@rgu.ac.uk with details. The item will be removed from the repository while the claim is investigated.

“© © 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Organizational Factors Shaping Software Process Improvement in Small-Medium Sized Software Teams: a Multi-Case Analysis

I. Allison
School of Computing
Robert Gordon University
Aberdeen, UK
i.allison@rgu.ac.uk

Abstract—Previous work looking at software process improvement (SPI) in small organizations has highlighted difficulties faced by small organizations in implementing SPI successfully, but there is little analysis to understand why this is from an organization theory perspective.

This paper presents an analysis of SPI across six software teams in the UK using a framework based on Giddens' Structuration Theory. Using a structurational perspective helps to draw out how the process improvements are enabled and constrained by their context. By comparing these across the six situations the key similarities and differences across the cases are highlighted.

This work extends the existing literature by helping to identify the areas of risk that need managing in small SPI initiatives. The study shows the issues as pertained to the six contexts and actions in each case. The paper highlights how the context influences the outcome.

Keywords-Software Process Improvement; Small-Medium Enterprises; Structuration Theory

I. INTRODUCTION

Software process improvement (SPI) programs support the definition and enhancement of the processes adopted by the software organization. To address these needs the Capability Maturity Model Integrated (CMMI) has been developed by the Software Engineering Institute for organizations to adopt / be evaluated against [2]. Organizations adopted CMMI based SPI mainly to improve their product quality and project performance but also to improve process management [8].

Others question the long-term impact of the SPI activity [10]. Not all efforts have been successful or without issues. In particular, small and medium sized enterprises (SMEs) have shunned or struggled to adopt CMMI and other such maturity models [4, 12]. However despite these problems, there is a lack of studies of failed adoption of CMMI-based SPI [9].

There is a growing body of knowledge developing addressing process engineering in smaller organizations and teams. The primary reason for the attention on smaller organizations is that they face specific difficulties related to the

limited resource they can put towards any initiative. The implementation of software engineering techniques is a difficult task for small organizations as they often operate on limited resources and with strict time constraints [6]. CMMI and other methods appear to overwhelm SMEs as the resources required to implement the detailed processes is too much [11]. And so often before they start small business abandon SPI and never benefit from process capability maturity because "they consider it infeasible to adopt" [9, p.891]. To address these difficulties researchers have proposed alternative approaches for SMEs. A number of lightweight or agile approaches have been suggested [For example: 5, 6, 7, 12].

If we are to understand the problems involved in making process changes in small teams though, we need to understand the organizational factors involved in the change [13]. It is therefore appropriate to examine the organizational issues that arise as smaller software development groups move toward a more structured, process-oriented environment. So, here we will address the following research questions:

- How does the context of the SPI initiative affect the way it is enacted?
- How are the changes undertaken and what are the constraints actors find when instigating changes to processes?

Using six case studies from different sized teams in different contexts, this paper builds on the recognition in recent longitudinal studies that SPI understood from an organizational theory perspective helps to draw out the nuances of the changes. To help to explain the issues arising during the SPI initiative, a theoretical framework is adopted based on Giddens' Structuration Theory to analyze the organizational features that shape process improvement.

This work extends the existing literature by helping to identify the areas of risk that need managing in small SPI initiatives. The study shows the issues as pertained to the six contexts and actions in each case. Drawing from these findings, the paper begins to provide a direction forward for the community in addressing these issues.

II. RESEARCH APPROACH

A. Research Method

This paper presents an analysis of software process improvement (SPI) across six software teams in the UK. Cases were identified from both organizations producing software products and software functions within non-software organizations, and in different size categories of organizations: micro < 10; small < 50; medium < 250 [13].

Six software teams were chosen: one for each size category within the two sets of organizations (see table 1), with the focus on the size of the software unit rather than the whole organization for comparison purposes. The cases have been selected from a set of action research projects undertaken with companies. Due to the nature of the research with the organizations, ready access was given to people and information so that data was captured through active participation and observation. This paper is not reporting the action research findings as such, but reflections across the set of cases to develop a more holistic response to the events.

TABLE I. CASES SELECTED

Size of team	Autonomous unit developing software products/ services	Unit within a larger non-software specific organization
Medium	Pharmaceutical Services [medium 1]	Telecommunications Systems [medium 2]
Small	Bespoke Business systems [small 1]	Market Analysis Packages [small 2]
Micro	E-Commerce Management [micro 1]	Educational Qualifications [micro 2]

B. Theoretical Framework

Software processes can be considered to emerge by means of a structuring process between the context and the content of the action [1]. The contextual factors include those within the organization and those external to it. So through time there is a metamorphosis of the context, the actor's understanding of the situation, and the processes enacted (see Fig. 1).

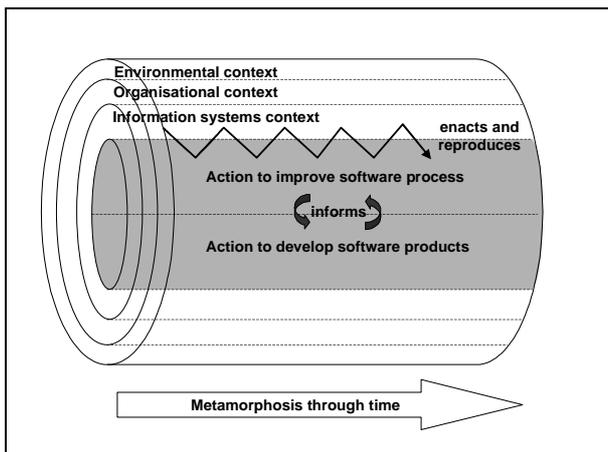


Figure 1. Emergent view of SPI (from [1])

Giddens' Structuration Theory is used to form a theoretical framework showing the intertwining of these facets of organizational emergence (see table 2). Beginning with the current historical context, Giddens' duality of structure weaves together action and social structures through a set of modalities. The enactment of, and changes to, the software processes are seen to embody the modalities of the structuring process. The process of change is understood to occur through the linkage between action of software practice and its context. So through time there is a metamorphosis of the context, the actors' understanding and intentions, and the software process as it is enacted.

Using a structural perspective helps to draw out how the process improvements are enabled and constrained by their context. The process of change can be analyzed through the three modalities: interpretive schemes, facilities, and norms. By comparing these across the six situations common factors are highlighted.

Human communication uses interpretive schemes to make sense of actions, so knowledge is drawn on and changed through action. Developers draw on knowledge, prior software engineering frameworks, and shared experiences discuss and undertake the SPI changes.

Next, human agents also draw on facilities, such as human and technical resources, to maintain or modify structures of domination. To draw on (or not) personal or organizational resources in order to retain or alter existing software approaches is within the control of all practitioners. Each member of the organization has the power to conform or challenge a suggested change. Individuals and groups may exercise power to resist in some circumstances and not others. Managers also can resist initiatives from employees through their disinclination to mobilize resources.

Thirdly, we sanction our actions by drawing upon norms thereby creating or recreating structures of legitimation. Norms are the rules or standards that govern appropriate conduct, constraining and enabling action. So, software process improvement is a constant process of negotiation, communication and establishment of norms through the everyday relationships enacted within the process improvement program and software development process. Software processes and methods used within the group are drawn upon as norms, and in so doing recreate structures of legitimation. The norms that develop are legitimated through the shared language of the community-of-practice developed from mutual knowledge of their traditions.

TABLE II. ANALYTICAL FRAMEWORK

Structural Context
Organizational History; External environment; Organizational Culture and Structures; Software team
Emergence of Process and Products
Dialectic of defined process and process-in-use Planned and unintentional change & consequences
Process of Change
Interpretative Schemes (stocks of knowledge; engineering approaches as frameworks for learning) Facilities (use of personal or organizational resources to change/retain current approaches; trust between managers and practitioners) Norms (the defined process acts as the norm; practised process becomes the norm; language as active process of legitimisation and institution)

III. CASE STUDIES

This section briefly outlines the factors noted in the analysis of the six cases. Each of these cases was analyzed in a longitudinal manner to ascertain the factors as they emerged through time. So whilst the factors are drawn out in a more static fashion here, this does not infer that the dynamic nature of the cases is not important. The above analytical framework is used to discuss the cases: the context and actions involved in the change are outlined in this section, and the following section discussed the process of change across the cases.

A. Medium organisation 1

1) Context

The Medium1 case examines a software engineering team of 60 in a Global Telecommunications company. As a well established organization, the team had clearly defined processes in existence prior to this initiative, but problems remained with getting products to market quick enough. So they introduced Scrum in the belief that it would differentiate them from the competition by producing higher quality solutions quicker.

2) Emergence of Processes and Products

The approach adopted wrapped other engineering practices, including eXtreme Programming and Rational Unified Process. An expert in the Agile methods was employed to assist with the change of working practices and values required to make the new process a success.

To encourage significant behavior change within the organization, the new working practices were established straight away allowing problems with the practices to be identified and dealt with early on. Engineers adopted the new practices with enthusiasm.

Rather than piloting the scheme with a single team on a small project, the process was used on a large-scale project for another division of the organization. The project represented a significant investment that incorporated most of the software engineering department's effort.

The company waterfall approach was retained as an overarching feature with agile being used somewhat covertly. But, the way in which the Scrum framework was initially

designed and implemented, customer buy-in and participation would be a requisite for the success of the process. Thus the conflict between conventional and Agile ways of working caused issues with the expected delivery of software.

Originally, the change was thought to affect just the engineers, not the management. However, the Scrum framework requires management to change from a traditional project management perspective to a more mentor like approach. Once a Sprint backlog has been assigned, the Sprint initiated, the management should not interfere with or interrupt the team, other than to facilitate the removal of any impediment.

Thus the change to having self-managing teams rather than having one person in charge meant that management could not keep up with the dynamics in the teams. One team member expressed his frustration, "Either [the management] have to trust us to get on with task or not."

After nine months, due to increased pressure from the customer, management allowed the process to regress to a traditional phased approach to meet the expectations of the rest of the organization. Although several aspects of the agile process, such as scrums and continuous integration are still being used, the current situation is an ad-hoc mixture of top down structured projects and a bottom up agile process.

B. Medium organisation 2

1) Context

Medium2 is a 13 year old development group working as an independent software group providing services to pharmaceutical companies. A team of 50 developers generated software tools for the industry. As such they were subject to stringent regulation and audit. Business requirements had to be traced into the software for compliance purposes. The tools were developed through a common architecture, but that architecture was compartmentalized, with functional silos. Projects were affect by "scope creep", and often over ran. The motive for the SPI initiative was to address these issues but also the company's aim was to get to level 2 of the Capability Maturity Model as a form of external legitimization of its quality.

2) Emergence of Processes and Products

Following a gap analysis two specific, and independent, change projects were introduced one on project management methodology and one on requirements capture. The Software Development Manger noted that "obtaining internal resources is difficult".

So, both of these projects were undertaken with external resource and expertise. The consultants worked with internal champions and developed new processes for the organization. The issues then came as they were introduced in the team. Training and support was provided but the existing approaches remained the norm with only minor evidence of change initially. The underlying issue of the siloed product architecture was identified as a core issue that required the introduction of a technical architect to address. Overtime the new approaches began to become accepted following their use on specific pilot projects with consultant involvement.

C. *Small organisation 1*

1) *Context*

Small1 is a bespoke software developer for a set of local clients, primarily utilizing the Microsoft product set. It had been a start up company 8 years previously, with the Managing Director forming the initial team. The company has grown to a team of 15, with the owner now acting more in a sales role. The Software Development Manager initiated the SPI project as a way of managing this growth in the team and following significant cost overruns in recent projects. The project had Board approval and was tied into business key performance indicators (KPI). They also desired external benchmarking of their approach for marketing purposes and so were looking to achieve CMMI level 2.

2) *Emergence of Processes and Products*

The previous development approach was a traditional structured approach, but the processes were not formally documented. This approach had been drawn from the Software Development Manager's previous knowledge, but new staff were challenging this perspective – both in terms of improving the robustness of the process but also seeking a more agile approach to managing the client requirements. There was project management in place, but the management was primarily related to measuring the chargeable time for clients.

Throughout the initiative the Software Development Manager was committed to making the changes necessary, putting a lot of his own time into it. It was difficult to obtain other time as one of the key business measures was time sold, so taking people off income generating projects was not welcome.

The processes focused on were project planning and control, technical documentation, and peer review. The latter two as a means of improving team communication about the client project. However, whilst these new approaches were trialed, as immediate benefits were not obvious, the team reverted to what they knew. Eventually the focus for the SPI moved towards looking at forms of project management and life cycles. So a review of external models was undertaken, with an agile approach being preferred.

So despite the approval of the Board, the commitment from senior management, and the close ties with the business KPIs, the demands of client projects eventually lead to the initiative becoming unsustainable – at least in the short term. However, over a longer period the company remains committed to following through on this initiative as they understand the business imperative. The agile approach is now embedded in their standard practice.

D. *Small organisation 2*

1) *Context*

Small2 is a 25 person software unit within a division of a global information services business. The business as a whole has a turnover in excess of £1 billion. At the time of the SPI initiative, the unit had been in existence 10 years and were developing Market Analysis tools. Whilst processes could be seen to have evolved over the whole period, the specific SPI initiative was instigated following a significant upgrade to the

core product. Due to delays in getting this product to the market and defects in the early versions of the system, the company lost market share. In line with the external professional environment, the company also began to adopt a more mixed sourced approach to the development of its portfolio of products, buying in components to form part of the product set.

2) *Emergence of Processes and Products*

Small2 had no desire to get externally assessed for its process capability, and so focused on goals that they felt would result in improvements to their products and time to market. In tying the SPI activity to both the business goals and the motives of the team members they were able to create a clear motive. Utilizing process action teams, the Software Development Manager included everyone in the improvement activity, with frequent reviews of progress.

As with other cases, however, resources were diverted from the task onto development projects and some individuals withheld their resource as they saw the SPI activity as management's responsibility, but significant process improvement occurred as process ideas emerged from the individual developers' software practice. New processes included the development of improved project planning and control, component based development, software review and testing, risk management, and project evaluation. These resulted in a significant reduction in defects and reduction in the variability on the delivery schedules.

As individuals reflexively monitored their own actions they identified actions to change the process. To achieve this change they were prepared to apply their own and the team's resources. By recognizing the relevance of the new approach they were able to recreate the team's norm, and thereby change the interpretive schemes of the unit. Aspects of language, knowledge and communication were all important in shaping the outcome of the SPI activity. Individuals drew on external forms of legitimation to justify their changes, such as their prior experiences or evidence from external professional practice. The personal drive for improvement from the management was evident through comments made in interviews, but an underlying motive was the desire to strengthen the internal position of the development team compared to other parts of the group. So whilst the change reflected a negotiated state, the initiative resulted in improved business performance.

E. *Micro organisation 1*

1) *Context*

Micro1 is a self-contained business of four developers. They had 8 small business clients for whom they provided e-commerce services. The business model was based on commission from the web sites as well as a fee for creating it. Therefore, reduction in cost of development and exploiting cross-sale opportunities was a key driver for the organisation. They were keen to overcome the need develop a new system for each client by creating a meta-system /information architecture. The process improvement was therefore in systems architecture design rather than the priority CMM key process areas of project management. The company initiated

an improvement to their approach to developing a common architecture using semantic web / web services concepts.

2) *Emergence of Processes and Products*

The approach and information architecture concept were developed independently using external expertise and piloted with the organizational data. To change the development approach required the company to restructure its databases. After initiating the change and trialing it successfully, the company regressed to its previous approach. Even though in the long-term this would have brought additional income at reduced development effort and risk, it did this because of the cost and risk involved in making the change.

Eventually a new client project was developed using the new approach, with the intention of migrating other clients later. Training and ongoing support was provided for this development from the external consultant.

One of the key constraints in this case was the lack of knowledge and experience in the team. Despite the strength of software engineering capability the lack of appreciation of the new ideas intensified the concern about their business risk. Also, ongoing need to bring in revenue using the same resources led to the stalling of the change. Only the persistence of the consultant and the desire of the senior business partner brought the project back on track.

F. *Micro organisation 2*

1) *Context*

Micro2 is a consortium of four educational awarding bodies in the UK vocational sector operating as a stand-alone organization with approximately 50 employees across the regional offices. The business has over 1500 customers. Two people formed the Systems team. The organization had no previous internal systems development experience, and lacked confidence in external IT vendors. The business focus was on developing a strategic system to support the regional offices to enable faster time-to-market for the qualification product development.

2) *Emergence of Processes and Products*

To achieve this goal, the Chief Executive employed a Systems Manager who drew on his previous experiences to develop a set of agile methods using known practices, such as aspects of agile development like prioritization of user requirements with the customer delivered in defined time-boxes, pair programming, database refactoring, critical chain project management. As the decision was primarily in his hands there was little need to deal with different views, other than where the processes impacted the user community. Employment of the right person to set up the processes was therefore important. Not having established process norms helped to change the approach, but some initial resistance was noted with the users being involved in requirements development and stage reviews, but once they saw this working conformance was quickly forthcoming – and the culture of partnership with the business was established. These processes were introduced successfully and continue to be utilized and built on as the team grows. The strategic system was

implemented within 12 months as planned with minimal level of defects.

IV. PROCESS OF CHANGE: A STRUCTURAL PERSPECTIVE

This section utilizes the Structural modalities to draw out the features that shaped the process of change in the 6 cases. These features have been highlighted by considering the duality of the context-action interaction as summarized above. The organizational history and social structures are understood to form the context of the actions that changed the processes and products, and in turn the actions reshaped the structures and context. Here the focus is on the key similarities and differences between the cases.

Across the cases interpretative schemes were drawn upon to make sense of the actions and changes as they occurred. Individuals drew on their stocks of knowledge from previous experience to make sense of the current situation and how it should change. In order to shape the change and influence the initiative, the team members explained their position and perceived ambition for change or otherwise using language that incited power structures in the organization. For example, use of organizational strategy terminology, existing shared experiences in the development, and a shared understanding of new approaches, such as agile, were used to convey the position. As new people were employed their prior knowledge and experiences were drawn on to propose alternative approaches, thus changing the shared vocabulary and understanding. We see in other cases external consultants were brought in to assist in the change of understanding and perception.

In the larger organizations we noted a greater degree of conflict and challenge, with team members either willing to put their own effort towards making the change or withholding it to resist this change. In the smaller contexts this ongoing conflict is less prevalent – at least internal to the software team – yet the dialectic of negotiated change was still evident in the development of the solution. In the software groups within organizations each of them had to change the perception of the other areas of the company, such as described above for Medium2 and Micro 2.

Individual motives were a key influencing factor. These motives were linked to organizational imperatives in a number of the cases. So the champion of the initiative drew on organizational “power”, coming from the perceived need to achieve a strategic direction or to react to an external perception, as a means of bringing resources to bear. So when these imperatives were clearly defined and supported it helped to maintain the initiative (e.g. Medium 1, Small 2, Micro 2). In the other organizations this influence changed and began to undermine the individual’s domination and ability to apply the organizational resources for this purpose.

Organizational norms, as evidenced through previous practice, were a damping factor in the change. As described above a number of the organizations reverted to previous approaches. As the processes began to be understood and the revised practices showed benefits the espoused and practiced practices became the norm. Even where we see reversion, this

was not necessarily by the whole group – nor forever. So the ability and willingness to challenge the existing approach was seen to regulate the pace of the change.

V. DISCUSSION AND CONCLUSION

The paper has outlined six cases and utilizing a Structural approach has drawn out some key aspect of how the change was undertaken. The SPI projects were drawn from small software teams of different sizes. Whilst these groupings (by size) showed some differences other factors running across the cases were seen as most important in enabling and constraining the improvement processes.

The common factors identified from the cases that have a constraining (or enabling) impact can be summarized as:

- Business motivation / goals for doing the SPI project and how well these are linked;
- Resources brought to bear – both personal and organizational – and resistance to change; resources for the improvement became more of an issue with reduction in organization size;
- Knowledge and prior experiences that the organization brings to the definition of processes and the process improvement project itself;
- Current norms / structural barriers and how these act to constrain the change;
- Political strength of the champion / other key players.

These points give rise to risks that management need to control in the same way that any other project governance process would suggest controlling risk. Key suggestions arising from the cases are to:

- Ensure the process improvement project is managed as a strategic project with senior management commitment. This commitment was strengthened when the project was intrinsically tied to the corporate / unit goals, but even that did not imply success on its own.
- Select a process improvement champion who has experiences that they can draw on to deliver the improvement [see 3]. Their political strength is important. Aspirations and activities have to be proportionate. Starting small can help deliver results. Involvement of a number of people in a SPI team can help spread this risk, but only if they are committed to the initiative.
- Training for the champion and others does help, but is insufficient on its own. SPI in small teams are project-oriented: their processes are rarely driven by a long-term strategy. Consequently, learning and knowledge management practices can rarely be observed. [5] Ongoing mentoring from outside the organization improves the ongoing knowledge exchange and development.

- Address prior experiences – especially either negative experiences or entrenched positions. Pilot projects can assist in changing the norms.

Using the Structural framework has helped to identify these risks. The analysis presented in this paper is a brief summary of six longitudinal case studies. Further expansion of these cases would help to draw out the nuances of the different projects and scenarios. Ongoing work is therefore required to develop the common factors into a risk management tool to support SPI champions to deliver improvement, especially in smaller development teams.

ACKNOWLEDGMENT

The contribution of the numerous other collaborators and contributors to the original primary research projects is acknowledged. The author is grateful to each of them for their ideas and input to the original research without which this paper would not have been possible.

REFERENCES

- [1] I. Allison and Y. Merali, "Software process improvement as emergent change: A structural analysis," *Information and Software Technology*, vol. 49, pp.668-681, 2007.
- [2] M.B. Chrissis, M. Konrad, S. Shrum, *CMMI: Guidelines for Process Integration and Product Improvement*, Boston,MA:Pearson Education, 2003.
- [3] G. Coleman and R. O'Connor, "Investigating software process in practice: a grounded theory perspective," *The Journal of Systems and Software*, vol. 81, pp. 772-784, 2008.
- [4] R. Conradi and A. Fugetta, "Improving Software Process Improvement," *IEEE Software*, vol.19(4), pp. 92-99, 2002.
- [5] N. Habra, S. Alexandre, J-M. Desharnais, C.Y. Laporte, and A. Renault, "Initiating software process improvement in very small enterprises: Experience with a light assessment tool," *Information and Software Technology*, vol.50, pp.763-771, 2008.
- [6] A. Mishra and D. Mishra, "Software quality assurance models in small and medium organisations: a comparison," *Int. J. Information Technology and Management*, vol 5(1), pp. 4-20, 2006.
- [7] F. Pettersson, M. Ivarsson, T. Gorschek and P. Ohman, "A practitioner's guide to light weight software process assessment and improvement planning," *The Journal of Systems and Software*, vol. 81, pp.972-995, 2008.
- [8] M. Staples and M. Niazi, "Systematic review of organizational motivations for adopting CMM-based SPI," Technical Report PA005957, National ICT Australia, Sydney, Australia, 2006.
- [9] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An exploratory study of why organizations do not adopt CMMI," *Journal of Systems and Software*, vol. 80, pp.883-895, 2007.
- [10] S. Rifkin, "Is Process Improvement Irrelevant to Produce New Era Software?" in *Proc. Software Quality*, Springer-Verlag, Heidelberg, Germany, LNCS 2349, pp.13-16, 2002.
- [11] M Sivashankar, A. M. Kalpana and A.E. Jeyakumar, "A framework approach using CMMI for SPI to Indian SME'S," in *Proc. International Conference Innovative Computing Technologies*, pp.1-5, 12-13 Feb. 2010
- [12] C.G.V. Wangenheim, S. Weber, J.C.R. Hauck, G. Trentin, "Experiences on establishing software processes in small companies," *Information and Software Technology*, vol. 48, pp. 890-900, 2006.
- [13] J. Zavala-Ruiz, "Organizational Analysis of Small Software Organizations: Framework and Case Study" in *Software Process Improvement for Small and Medium Enterprises*, H. Oktaba and M. Piattini, Hersey, PA: Information Science Reference, ch.1, pp 1-41, 2008.