# A Review of Prominent Work on Agile Processes Software Process Improvement and Process Tailoring Practices

Rehan Akbar[1], Mohd Fadzil Hassan[1], and Azrai Abdullah[2]

[1] Department of Computer and Information Sciences
[2] Department of Management and Humanities
Universiti Teknologi PETRONAS, Malaysia
rehankb@yahoo.com, mfadzil_hassan@petronas.com.my,
azraia@petronas.com.my

**Abstract.** Global software development has changed the overall software development practices. It has introduced various new software development processes and methodologies. A new generation of processes has increasingly replaced the traditional software engineering practices. Emerging practices such as agile based methodologies, software process tailoring, process improvement and management approaches have gained much attention during the recent years. Software engineering researchers have produced a number of good quality works in such areas. In this paper, a review of major contributions of the researchers on various aspects of software development processes is presented. Specifically, the analysis on different approaches of process improvement and tailoring is critically discussed in the paper. This research provides guidelines to the researchers on future research directions. The research emphasizes on the need of industry oriented practical approaches of software development to meet the challenges of global software development.

**Keywords:** Agile, Globalization, Process Improvement, Management, Tailoring.

## 1 Introduction

The last decade of the twentieth century was the beginning of the advancements in Information Technology (IT). Most advanced tools and technologies for software development were introduced and utilized. Unlike, traditional heavy weight approaches of software development, the software companies started giving preferences to the new alternatives such as light weight agile based methodologies. Such drastic changes occurred as the consequences of IT globalization [1]. Not only IT, the globalization has also affected the overall international socio-economic fronts and political scenarios of the societies [2].

Since the start of the twenty-first century, the effect of IT globalization has become more intense. Information technology was among the fields which were greatly affected by its consequences. Project outsourcing and agile methodologies are

J.M. Zain et al. (Eds.): ICSECS 2011, Part III, CCIS 181, pp. 571–585, 2011.

reported as two major products of IT globalization. Project outsourcing and factors involved in it were the basis of the increase in the use of the agile practices. [3] has discussed the reasons behind this change. It has generally setup the new trends in the software development industry. Preferences of software development teams and their criteria of selecting suitable tools, technologies and processes for software development have been changed due to the wide range of available alternatives. Project outsourcing proved to be the advent of new generation of processes in software engineering [4]. It is believed that traditional software development approaches could not withstand with the newly emerging practices and gradually became outdated.

The consequences of globalization started to appear during 1990s. Till the late 1990s project outsourcing had become the most common practice in the software industry. Mostly projects were being outsourced to the offshore companies. Offshore development started a new debate among the researchers regarding the processes, project management and performance issues. Many researchers have written a lot about it for example [4], [5], [6], [7], [8], [9]. In 2002, [10] presented a conceptual model for offshore development as shown in figure 1. In the model effort, elapsed time and rework were introduced as three factors of project performance measurement. The effect of the variables like quality processes, technical processes and communication & coordination on performance was measured through empirical data.

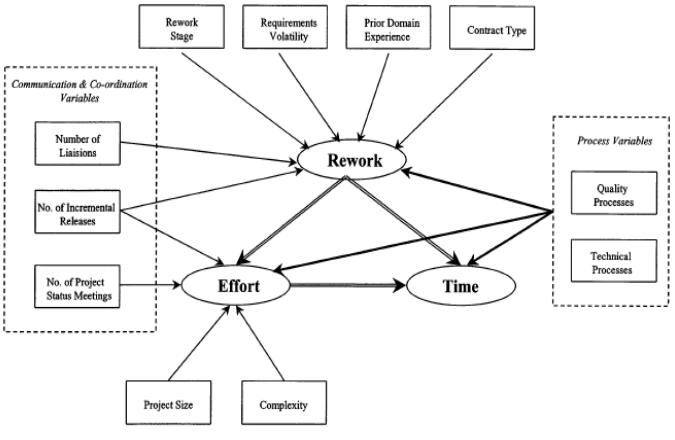


**Fig. 1.** Conceptual model

In another study [11] emphasized on the importance of the role of cross cultural issues in the performance of outsourced projects. Project outsourcing is also attributed to the geographically distributed development. In this regard, a road map to make a governance framework for distributed software development is presented in which characteristics of organizational level standardization, project execution, planning and infrastructure were discussed [12]. For small and medium scale organizations [6] described common issues and their solutions in outsourced projects

such as contract management, demand supply management, documentation, tool support, cultural and team level. These approaches provide significant control over such issues.

In short, as the consequences of IT globalization, a paradigm shift from the traditional heavy weight software development practices to the light weight agile based methodologies has been reported by many researchers. Since late 1990s till to-date the response to agile methodologies is overwhelming. The sole reason behind this was the style of development emerged as a result of outsourcing projects to offshore teams.

The role of software engineering research community has been very important in this regard. They have produced a good quality work, and have introduced the efficient light weight techniques of software development and project management. Agile methodologies, process tailoring, process improvement and management practices were the major areas in which most of the evolutionary work was produced during the last twelve years and further contributions are still being made.

In this paper we have presented the review of the contributory work produced by the researchers in all the three areas mentioned above. The review of the selected good quality papers is made in an analytical way. The papers were selected based on the novelty, key ideas and methodology used. Only a limited number of papers could fulfill these criteria. The agile practices are discussed in general, and a critical review on the process improvement and tailoring models, frameworks and standards is made in particular. Based on the analysis of the reviewed work, we have made conclusion on the future trends and practices of software development and research works.

The structure of the paper is as follows. In section 2, the prominent work on agile based methodologies is presented. Section 3 covers the review on process improvement and management approaches, while important models and frameworks of process tailoring are discussed in section 4. Conclusion is presented in section 5.

Finally, the paper suggests the solutions to the global software development challenges and suggests future research directions. This paper provides a hands-on review of past and current software development practices. It also critically analyzes the applicability and practicability of these approaches in accordance with the software development industry. Based on the analysis and overview, we have recommended the practices which are considered the best for the software industry. This paper emphasizes on the realization of the need of applied industry based solution oriented research works and software processes.

## 2 Agile Methodologies – A paradigm Shift

The clients of software projects prefer to launch their products early in the market to compete with their business rivals. This requirement of the clients keeps the developers under continuous pressure [13], [14]. To compete with the market, the ultimate requirement of the client is only the working code [4]. The support through fast paced development to release the working code early is provided in agile models [15]. Agile models emphasizes on minimum or no documentation. This aspect started debates on agile methodologies among the researchers. Two schools of thoughts in which one supports the agile models and the others to the traditional approaches [16],

[17] have born. Two agile based methodologies Extreme Programming (XP) [18], [19] and Scrum [20], [21] among others are widely used by the developers due to the available support of fast paced development. [16] has discussed agile methodologies in connection with ISO standards. Software engineering researchers have produced a number of models and frameworks on agile based methodologies. In 2001, agile manifesto comprising of twelve principles was formed in order to standardize the agile methodologies [15]. Like other approaches, agile methodologies also have some limitations with respect to the different environments and project requirements. In a study [15] has identified the following limitations of agile for distributed development, subcontracting, reusability, large teams, quality of safety critical products, and large and complex software systems.

The disadvantages and benefits of agile processes that are claimed by [15] are based on a set of assumptions. Therefore, their existence and non-existence in different environments may be doubted. As mentioned earlier that there is always a debate between agile supporters and traditional approaches supporters. In this regard [17] has compared traditional approaches with agile based mainly on the factors such as control, project management, team roles, way of communication and role of client. The control of project, project management and role of client are considered more important and critical in agile as compared to traditional approaches. Other researchers such as [22], [23], [24], [25], [26] have presented the similarities and differences between both approaches. As a matter of fact all software engineering methodologies have limitations [27]. In their framework (CHAPL) to understand the relationship between both approaches, [28] has concluded that traditional software engineering approaches and agile methodologies have "common philosophical origins" and are "technically compatible and complementary" to each others. This arises the need of a reasoning framework to determine the suitability and selection of software engineering methodologies in particular circumstances [27].

In addition to other factors, the selection of SE methodology also depends on the size of the company. Limitations of resources like financial, human etc forces the small companies to adopt light weight methodologies. Large companies with good resources pool prefer more standardized approaches. The web based application development has gain more popularity during recent years. Such kind of web development is being done in almost all the companies irrespective of their sizes, scales and environments. Unlike large organizations, small organizations face the situation of un-decidability many times during the project life cycles. In 2007, [29] discussed the software process models for web based application development in small software development companies. Light weight agile methodologies have proven to be the most result oriented methodologies for small, medium as well as large companies. In developing countries 75% - 80% companies are small and medium sized. Irrespective of the size of the company, agile methodologies are equally beneficial for them because the quick access to the ultimate goal of project success, which is similar for all the companies. Agile methodologies for example Extreme Programming (XP), crystal and Scrum have proved their worth in all kinds of environments. [30] has summarized an overview of XP and have made recommendations on how pair programming, a form of XP, can be implemented. In today's environment, the development time of web based applications has been reduced to few months. In current circumstances, the most important phase of

software development is requirement gathering, analysis and tracking. The agile approaches have also been quite efficient on this phase. In this regard, the agile hypertext design method has been proposed by [31]. In another work, [32] has proposed an agile approach for web systems engineering. In a South African empirical study of 59 projects, results show that agile practices are significantly beneficial in process improvement and project success which ultimately leads to the satisfaction of stakeholders [33].

As mentioned earlier by [27] that all software engineering methodologies have limitations, so there is always a margin for improvement. Software engineering researchers are continuously working on the process improvement and management practices. A number of models, frameworks and standards have been presented by the researchers in this regard. In the next section we have summarized some of the important contributions of the researchers on process improvement and management.

## 3 Process Improvement - Part of Process Management

IT globalization has made the software process improvement and management the key research areas in software engineering today. Each organization irrespective of its size, scale and types of the projects has to rely somehow on a software process for its development and management tasks. This dependency has made the development processes more critical element in the growth and success of projects.

The recent advancements in the field of IT have changed the overall software development scenarios. Organizational structures, preferences and priorities of the clients and the organizations have been changed. From hierarchical structures of departmental based divisions, organizations have shifted to the business process oriented team structures [34]. In a research work [34] has presented a reference framework for process-oriented software development organizations. According to the framework, management and support processes are considered necessary for the output of the project and client's satisfaction. A hierarchy to organize processes is proposed with the process management team at the top and process execution teams & their team leaders at the bottom in the hierarchy. The framework is then examined in the context of a generic as well as a software development organization. The framework provides the basic structure to establish process oriented organizations. Emphasis is given to the processes as critical factor and their continuous improvement and adaptation.

In another study, a review on the roles of the water fall model, capability maturity model (CMM), ISO-9000, SPICE, Trillium and BOOTSTRAP in process improvement are discussed by [35]. The water fall model is considered as the foundation model in organizing the software processes activities. Still this model is widely used in the companies. The Capability Maturity Model (CMM) provides various key process areas (KPAs) for the process maturity, improvement and standardization. The problem with the CMM standardization approach is that it has limitations for small and medium scale organization due to scarcity of the resources. It seems more applicable for large organizations. ISO-9000 is a general quality standard and so far further IS standards have been introduced for software development such as ISO-9001. It is believed that process improvement and

management approaches may vary based on organizational goals, priorities and project requirements [35].

In 2004, [36] discussed process improvement approaches such as capability maturity model, six sigma, lean development and ISO-9001and raised the issues such as :

i. Lack of academic research factor on the efficiency and effectiveness of these approaches.
ii. "They are based on very similar concept and techniques."
iii. They are unable to present best practices specifically for the software development.
iv. They are just the improvement approaches for established processes.

The gap between academic researchers and actual industry practitioners is believed to be the basic reason behind the ineffectiveness of various approaches [37]. The researchers don't find resources to collect the real data from the industry and on the other hand industry people also seem unwilling to cooperate with the researchers and consult their published work to find the solutions of the problems. This has led to a situation where software engineering research has been ineffective to meet the industry issues. Two limitations in software engineering research methodology namely poor understanding of the theory and inability to perform suitable testing have been reported by [38]. Unlike other fields of sciences, research methodologies in software engineering are not well established and structured. In 1995 and later, [39], [40] in their critical study on experimentation in software engineering found the software engineering papers worse among that of 43% papers of computer science in general with poor experimental design and testing methodologies. [41], [42] had also the same kind of observations. In 2002, [43] described the types of research questions, research strategies, types of results and validation techniques in software engineering research. Similarly many other researchers such as [44], [45], [46] have presented a good quality work on software engineering research methodologies. The areas of process improvement and management have always been preferred areas of research. Both areas have become more critical since the start of global software development.

With regards to the size of the organizations, process management and improvement practices have become more problematic and important. In this regard, [47] has proposed an integrated process management system for project management and business processes as a part of a workflow management. The emphasis is given on automation and tool oriented management systems. Enterprise level organizations usually adopt process improvement and management practices to get benefit from the business. Presumably process improvement practices are always considered as beneficial for organizations. Several researchers have presented their point of views on process improvement such as [48], [49], [50], [51]. The managers can better improve a process by avoiding false assumptions such as business improvement is dependent on process improvement, process change leads to process improvement, "software processes are non-lethal," and business process management activities do not need IT processes [52]. [53] has considered commitment as an important factor in software process improvement and has pointed out the four misconceptions in the existing commitment models for software process improvement. These and many other assumptions are considered as hindrances in process improvement and

management practices. Further, [52] has not found any direct relationship between software process change and performance. Process management and improvement processes become the direct responsibility of the project manager. In addition, lack of coordination among team members, and delayed reporting of problems and issues [54] in the project are also considered as one of the factors that affect the management process. In a research work on resource instantiation policies to automate software process management [55] has criticized on the existing process management tools and technologies that they "fall short in their computational capabilities and only provide passive project tracking and reporting aids". The commitment to improve a software process is a basic integral part of the whole process since its beginning. Three forms of commitment such as affective, continuance and normative commitment are introduced by [53]. Many other researchers such as [56], [57], [58], [59], [60], [61], [62], [63] have discussed various other aspects of software process improvements. A prominent work in this regard is done by [56] in which a platform which is based on the CMMI, SPICE, PSP, and 6-Sigma standards is proposed to improve the capabilities of a software process. Emphasis is given to the continuous efforts of process improvement after its beginning. Project managers of small and medium size organizations consider CMMI and ISO standards of process improvement unnecessary [57]. However, [63] considers process standardization and process reuse as the same thing. Extensive documentation, limited resources, training cost, lack of guidance, unnecessary processes, practices and reviews are the limitations of CMMI to be adopted by small organizations [61]. Based on this argument, [63] proposed a meta-model to standardize the reusability of a software process integrating the people, roles, process and infrastructure components. The customer's satisfaction, to the best of our knowledge, for the first time, was regarded as one of the factors in the software process improvement model by [57]. Software quality, cost, project scheduling and organization performance are the motivational factors for a project manager behind the innovative software processes [58], [62]. In addition to software process, skill level of software development team, tools and technologies, software complexity, deadlines, interaction and communication are also important factors that determine the software quality and organization performance [62].

Software development processes, models and frameworks are not the new areas for the researchers. Hence, IT globalization has given the new direction to it. As a consequence, new trends in software development processes have been emerged. Software process tailoring is one of the new emerging practices. Research works on software process tailoring is not available in quite a good number. In the next section we have discussed the prominent work on software process tailoring.

## 4 Software Process Tailoring – An Emerging Practice

The software process tailoring has emerged as a part of process improvement strategy. Though a number good quality research works on process tailoring have been presented by the researchers since 2000, but efforts had started back in 1980s. Software engineering researchers have realized the importance of this important practice and have produced some models and frameworks in recent years, but still a concrete effort is required from the software engineering research community.

In 1987, [64] presented a framework to tailor the software process as an evolutionary software improvement practice. In the framework process tailoring was performed based on the project goals and environments. The quantitative characterization through defect profile (analysis of errors, faults, and failures) classification scheme was made. The framework is supposed to be the beneficial contribution.

The process tailoring is of two types i.e. product tailoring and activity tailoring [65]. The tailoring approach formalized by [65] does not identify other tailoring activities except deletion and modification. The model provides very limited details on software tailoring process and is based on GV-model which is most likely not used by majority of the organizations. A software process is the key to success for a software development project. Software processes may be light weight or complex heavy weight approaches. Process tailoring is a standardized practice which is directly related with the tailoring of the activities of software development phases. During the process tailoring activity adoption and maintenance of a process standard and its reusability is necessary [66]. In the same work [66] has applied addition, deletion, splitting and merging process tailoring activities on a process module. The techniques such as correctness checking, conformance checking and compliance evaluation were used to verify the tailored process. This was considered as a good work on tailoring as four process tailoring activities were formally applied to a process. The need of a systematic approach for process tailoring and its verification was highlighted.

As mentioned earlier that IT globalization increased the use of agile based methodologies because of their fast paced development and light weight processes of software development. Due to this factor, most of the researchers focused on agile methodologies such as XP and scrum in their research works. In 2002, [67] applied their own tailored version of XP on a project and compared the differences, advantages and disadvantages with the standard XP approach. The small release plan, continuous integration of the components, use of pair programming, managing requirements only for the current build/milestone, planning and prioritizing by the developers, code refactoring, and testing were the practices adopted in their tailored version of XP. Except few issues, an improvement in the quality of their software was reported. In the same year [68] presented a framework to use process knowledge to tailor a software process and proposed a prototype tool to help to capture and use the process knowledge. The framework does not provide guidelines on types of information, and level and amount of information required to tailor a process. In a process tailoring framework [69] has defined two types of knowledge which are: 1) generalized knowledge that refers to the general rules, policies, standards and formulated information, 2) contextual knowledge that refers to the organizational decisions, events, time etc. [69] has beautifully expressed the effectiveness and efficiency of knowledge management in software process tailoring. In another work on knowledge based process tailoring [70] retrieved the most similar cases on the basis of existence of similar elements between past projects and a new project, and is applied to a new project. [70] retrieved the most similar cases (structural similarity)

through this approach and concluded that least modifications would be required in the process which would be generated through this approach. In 2004, [71] proposed a process tailoring framework that was also based on gathering information on existing techniques and practices and maintaining a knowledge base. Based on the knowledge repository activities such as problems analysis, finding the solution, proposed process implementation and its evaluation were recommended to perform. The model was more likely a similar kind of work as presented by [68], [69], [70].

The software process tailoring is directly related to the size of the organization and project's size, scale and scope. Software engineering research community is agreed that small and medium scale organizations prefer agile based methodologies for software development. Extreme programming (XP) is the most common approach being followed by majority of the companies. Therefore, varying sized companies and projects need to adapt XP or any other agile methodology according to their requirements. Authors in [72] used rule description practices (RDP) technique, similar to CRC cards, to tailor XP. Two type of rules were defined which are rules of engagement and rules of play. Rules of engagement were agile based and rules of play were based on XP. The RDP technique proved to be applicable in almost all types of environments. Other researchers such as [73], [74], [75], [76], [77], [78] as cited by [72] have presented very practical solutions on adopting XP according to different environments. [79], [80] in their research work claimed that it is a very common practice to partially adopt the XP.

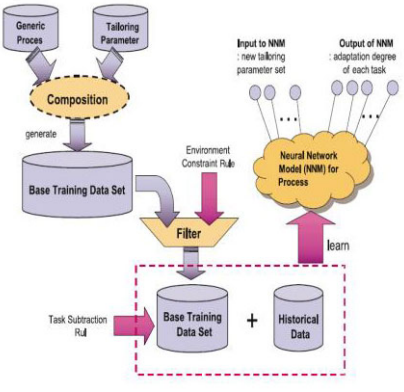


**Fig. 2.** Process Filtering Phase[84]

As a general procedure [81] performed tailoring of the software process meta-model [82]. Analysis of organizational environment, process life cycle, roles, activities and artifacts, documentation, training, and testing were the key elements of the procedure. The model did not provide any details of tailoring activities except presenting a general procedure to follow. The software engineering researchers have performed process tailoring from different aspects. The fundamental part of the process tailoring is the identification of reason or problem for which solution is provided through process tailoring. It deals with the analysis of various elements at

organizational level and project level. In a similar kind of work on process tailoring [83] have presented four types of strategies to analyze a system. The categorization is made on the basis of immediate needs and long term goals for each type. Based on the information achieved from the analysis of each strategy, the selection criteria are proposed. Similar to other works, the model provides another way of creating a knowledge base for process tailoring. A different approach for process tailoring using neural networks was presented by [84]. The process was accomplished through three phases namely process filtering, reconfiguration and feedback. During first phase tasks were selected from a generic process, then precedence was set during the reconfiguration phase and finally a tailored process was produced. During feedback phase the performance of the tailored process was evaluated. The filtering technique was the uniqueness of this approach and is shown in figure 2.

The authors have used more systematic but complex approach during the research.

The quality of a process as a result of process tailoring is very important for an organization. Bad process tailoring practices may affect the project budget, development time, quality of the software, compliance with the standards, and satisfaction of the employees. Also, the addition of unnecessary activities and omission of necessary activities are considered as wastage of time and money [85]. As a result of review of literature available on the software process tailoring [85] has reported 1) focusing on project level or organizational level, 2) case study in real environment, 3) size of the company, and 4) supporting tools etc as the major issues in process tailoring. The authors in this study have presented the review of existing software process tailoring approaches from almost all aspects, thus providing a guideline for future purposes. In a research work, [86] has performed four process tailoring operations namely addition, deletion, splitting and merging based on structured Petri Net without initial marks denoted as basic block. All four operations were based on four basic blocks namely, sequence block, selection block, iteration block and concurrency block. The approach followed in the study is more systematic as compared to other works but is unable to answer the questions raised by [85]. For project level process tailoring, a four step iterative approach was presented by [87]. Evaluation of the project's goals and environment, assessment of the challenges faced by projects, finding the suitable process tailoring strategy and lastly validation and evaluation of the tailored software process operations were performed to tailor a process. The authors have proposed an iterative approach to make it more flexible because they believe that processes gets more refined with the progress of the project. On contrary, they are also not in the favor of excessive and repeating process tailoring during a project lifecycle.

Software process tailoring is a part of process improvement approaches. All the organizations irrespective of their sizes adapt a process model according to their needs. Size of the project and organization, project scale, its complexity vary from one project and organization to other project and organization. Even a process model suitable for a project, might not be beneficial for the other project in the same organization.

Software process tailoring is an emerging trend because of the un-decidability factor involved in the selection of a suitable process according to the requirement of the project and organization. During our analysis we faced the limitation of the availability of literature on process tailoring. Though work on process tailoring had

started in 1980s but software engineering research community could not pay deserved attention to this practice. Now a days majority of the software development companies are relying on software process tailoring approaches and it has found a prominent place in software engineering research works.

## 5 Conclusion

Process improvement practices are the continuous part of project management and standardization processes. The work presented by the researchers in the areas of agile, process improvement and tailoring is very general. It neither addresses the industry issues nor meets their requirements. Most of the work has poor methodology and is not properly validated. Results do not show the applicability of these models and frameworks. Further, it is found that among all approaches, the process tailoring is the most smart and efficient practice to solve the process and projects related issues. It has emerged as a lightweight and faster approach. Almost all types of companies adopt process tailoring techniques. In support of agile methodologies, they have become the new generation of processes. As a newly emerging trend, software process tailoring needs more contribution from the researchers and practitioners. There is a need of more formal and applied approaches, methodologies, frameworks and standards in this regards. The involvement of the actual practitioners in the research process is required. In its current form, it is unable to provide solutions to the IT industry. More concrete efforts both by the software engineering researchers and actual practitioners are required through joint projects. During the coming years, process tailoring would be the ultimate choice of the companies. It has been realized that smart processes are being preferred by the industry and process tailoring is just the beginning.

## References

1. Ramasubbu, N., Ballan, R.K.: Globally Distributed Software Development Project Performance: An Empirical Analysis. In: ESEC-FSE 2007, Cavtat near Dubrovnik, Croatia, September 3–7, pp. 125–134 (2007)
2. Cho, J.: Globalization and Global Software Development, Issues in Information Systems, vol. VIII (2), pp. 287–290 (2007)
3. Ktata, O., Levesque, G.: Agile development: Issues and Avenues Requiring a Substantial Enhancement of the Business Perspective in Large Projects. In: C3S2E 2009, Montreal Qc, Canada, May 19-21, pp. 59–66 (2009)
4. Akbar, R., Hassan, M.F., Safdar, S., Qureshi, M.A.: Client's Perspective: Realization as a New Generation Process for Software Project Development and Management. In: ICCSN 2010, pp. 191–195 (2010)
5. Akbar, R., Hassan, M.F.: A Collaborative-Interaction Model of Software Project Development: An Extension to Agile Based Methodologies. In: ITSIM 2010, pp. 1–6 (2010)
6. Rao, N.M.: Challenges in Execution of Outsourcing Contracts. In: ACM ISEC (2009)
7. Sterba, C., Grechenig, T., Pazderka, M.: Outsourcing as a Strategy for IT Harmonization – A Public Sector Case Study Proposing an Approach in Independent Stakeholder Scenarios. In: ICEGOV 2008, Cairo, Egypt, pp. 245–250 (2008)
8. Taylor, H.: The Move to Outsourced IT Projects: Key Risks From the Provider Perspective. In: SIGMIS-CPR 2005, pp. 149–154 (2005)

9. Kolawa, A.: Out Sourcing Devising a Game Plan. Queue (2004)
10. Gopal, A., Mukhopadhyay, T., Krishnan, M.S.: The Role of Software Processes and Communication in Offshore Software Development. Communications of the ACM 45(4ve), 193–200 (2002)
11. Narayanaswamy, R., Henry, R.M.: Effects of Culture on Control Mechanisms in Offshore Ourtsourced IT Projects. In: SIGMIS-CPR 2005, Atlanta, Georiga, USA, pp. 139–145 (2005)
12. Ramasubbu, N., Balan, R.K.: Towards Governance Schemes for Distributed Software Development Projects. In: SDG 2008, Leipzig, Germany. pp. 11-14 (2008)
13. Aoyama, M.: Web-Based Agile Software Development. IEEE Software 15(6), 56–65 (1998)
14. Cusumano, M.A., Yoffie, D.B.: Software Development on Internet Time. Computer IEEE 32(10), 60–69 (1999)
15. Turk, D., France, R., Rumpe, B.: Limitations of Agile Software Processes. In: Proceedings of 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering. ACM, New York (2002)
16. Theunissen, W.H.M., Kourie, D.G., Watson, B.W.: Standards and Agile Software Development. In: Proceedings of SAICSIT 2003, pp. 1–11 (2003)
17. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of Migrating to Agile Methodologies. Communications of the ACM 48(5), 73–78 (2005)
18. Beck, K., Fowler, M.: Planning Extreme Programming. Addison-Wesley, Reading (2000)
19. Fraser, S., Beck, K., Cunningham, W., Crocker, R., Fowler, M., Rising, L., Williams, L.: Hacker or Hero? - Extreme Programming Today. In: Proceedings of the 2000 ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2000), MIneapolis, MN, USA, pp. 5–7 (2000)
20. Schwaber, K., Beedle, M.: Agile Software Development with Scrum, 1st edn. Prentice Hall, New Jersey (2001)
21. Rising, L., Janoff, N.S.: The Scrum Software Development Process for Small Teams. IEEE Software, 2–8 (2000)
22. Glass, R.L.: Agile versus Traditional: Make Love Not War, Cutter IT Journal 14(12), 12–18 (2001)
23. Turner, R., Jain, A.: Agile Meets CMMI: Culture Clash or Common Cause? In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 153–165. Springer, Heidelberg (2002)
24. Fritzsche, M., Keil, P.: Agile Methods and CMMI: Compatibility or Conflict. e-Informatica Software Engineering Journal 1(1), 9–26 (2007)
25. Paetsch, F., Eberlein, A., Maurer, F.: Requirements Engineering and Agile Software Development. In: Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, p. 308 (2003)
26. Eberlein, A.: Requirements Engineering and Agile Methods: Can they benefit from each other? In: Position Statement in the Proceedings of Canadian Invited Workshop on Scaling XP/AgileMethods, Banff, Canada (2003)
27. Basili, V.R.: The Role of Experimentation in Software Engineering: Past, Current, and Future. In: Proceedings of ICSE, vol. 18, pp. 442–449 (1996)
28. Jiang, L., Eberlein, A.: Towards A Framework for Understanding the Relationship between Classical Software Engineering and Agile Methodologies. In: APSO 2008, Leipzig, Germany, pp. 9–14. ACM, New York (2008)
29. Tarawneh, H., Elsheikh, A., Lahawiah, S.: Web-Based Applications Development in Small Firms. In: Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, Corfu Island, Greece, February 16-19, pp. 69–74 (2007)

30. Aiken, J.: Technical and Human Perspective on Pair Programming. ACM SIGSOFT Software Engineering Notes 29(5), 1–14 (2004)
31. Wills, G.B., Abbas, N., Chandrasekharan, R., Crowder, R.M., Gilbert, L., Howard, Y.M., Millard, D.E., Wong, S.C., Walters, R.J.: An Agile Hypertext Design Methodology. In: HT 2007, Manchester, England, UK, September 10–12, pp. 181–184. ACM, New York (2007) ISBN: 978-1-59593-820-6/07/0009
32. Souza, V.E.S., Felbo, R.d.A.: An Agile Approach for Web Systems Engineering. In: WebMedia 2005, Poços de Caldas, Minas Gerais, Brazil, December 5-7, pp. 1–3 (2005)
33. Ferreira, C., Cohen, J.: Agile Systems Development and Stakeholder Satisfaction: A South African Empirical Study. In: Proceedings of SAICSIT 2008, October 6-8, pp. 48–55 (2008)
34. Fernandes, J.M., Duarte, F.J.: A Reference Framework for Process-Oriented Software Development Organizations. Software and Systems Modeling 4(1), 94–105 (2005), doi:10.1007/s10270-004-0063-0
35. Krishnan, M.S., Mukhopadhyay, T., Zubrow, D.: Software Process Models and Project Performance. Information Systems Frontiers 1(3), 267–277 (1999)
36. Card, D.N.: Research Directions in Software Process Improvement. In: Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004 (2004)
37. Akbar, R., Hassan, M.F., Abdullah, A., Safdar, S., Qureshi, M.A.: An Insight into Real Software Industry Paradigms and Software Engineering Research. In: International Symposium on Computers & Informatics (March 2011)
38. Xia, F.: What's Wrong with Software Engineering Research Methodology. ACM SIGSOFT Software Engineering Notes 23(1), 62–64 (1998)
39. Luckowicz, P., Heinz, E.A., Prechelt, L., Tichy, W.F.: Experimental Evaluation in Computer Science: A Quantitative Study. Journal of Systems and Software 28(1), 9–18 (1995)
40. Tichy, W.F.: Should Computer Scientist Experiment More? 16 Excuses to Avoid Experimentation. IEEE Computer 31(5) (1997)
41. Zelkowitz, M.V., Wallace, D.: Experimental Validation in Software Engineering. Information and Software Technology 39(11), 735–744 (1997)
42. Zelkowitz, M.V., Wallace, D.R.: Experimental Model for Validating Technology. IEEE Computer 31(5), 23–31 (1998)
43. Shaw, M.: What Makes Good Research in Software Engineering? International Journal of Software Tools for Technoogy Transfer 4(1), 1–7 (2002)
44. Marcos, E.: Software Engineering Research versus Software Development. ACM SIGSOFT Software Engineering Notes 30(4) (2005)
45. Seaman, C.B.: Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transactions on Software Engineering 25(4), 557–572 (1999)
46. Runeson, P., Host, M.: Guideline for Conducting and Reporting Case Study Research in Software Engineering. Empirical Software Engineering 14(14), 131–164 (2009)
47. Bahrami, A.: Integrated Process Management: From Planning to Work Execution. In: BSN 2005 Proceedings of the IEEE EEE 2005 International Workshop on Business Services Networks, pp. 11–11 (2005)
48. Hansen, B., Rose, J., Tjornehoj, G.: Prescription, Description, Reflection: the Shape of the Software Process Improvement. International Journal of Information Management 24(6), 457–472 (2004)
49. Herbsleb, J.D., Goldenson, D.R.: A Systematic Survey of CMM Experience and Results. In: Proceedings of the 18th International Conference on Software Engineering, pp. 323–330 (1996)
50. Stelzer, D., Mellis, W.: Success Factors of Organizational Change in Software Process Improvement. Software Process - Improvement and Practice 4(4), 227–250 (1999)

51. Rainer, A., Hall, T.: Key Success Factors for Implementing Software Process Improvement: A maturity-Based Analysis. Journal of Systems and Software 62(2), 71–84 (2002)
52. Bannerman, P.L.: Capturing Business Benefits from Process Improvement: Four Fallacies and What to Do About Them. In: BIPI 2008, Leipzig, Germany, May 13, pp. 1–8 (2008)
53. Abrahamsson, P.: Commitment Development in Software Process Improvement: Critical Misconceptions. In: 23rd International Conference on Software Engineering, ICSE 2001, pp. 71–80 (2001)
54. Keil, M., Smith, H.J., Pawlowski, S., Jin, L.: Why Didn't Somebody Tell Me?: Climate, Information Asymmetry, and Bad News About Troubled Projects. ACM SIGMIS 35(2), 65–84 (2004)
55. Reis, C.A.L., Reis, R.Q., Schlebbe, H., Nunes, D.J.: A Policy-Based Resource Instantiation Mechanism to Automate Software Process Management. In: SEKE 2002, Ischia, Italy, July 15-19, pp. 795–802. ACM, New York (2002) ISBN: 1-58113-556-4/02/0700$5.00
56. Kim, J.A., Choi, S.Y., Kim, T.H.: Management Environment for Software Process Improvement. In: International Symposium on Computer Science and its Applications. IEEE, NY (2008)
57. Xiaoguang, Y., Xiaogang, W., Linpin, L., Zhuoning, C.: Research on Organizational-level Software Process Improvement Model and Its Implementation. In: International Symposium on Computer Science and Computational Technology, pp. 285–289 (2008)
58. Guo, Y., Seaman, C.B.: A Survey of Software Project Managers on Software Process Change. In: ESEM 2008, pp. 263–269 (2008)
59. Morgan, B., Lowry, G.: Software Process Assessment and Improvement, Discussion Summary. IEEE Explore, 497–499 (1996)
60. Bueno, P.M.S., Crespo, A.N., Jino, M.: Analysis of an artifact oriented test process model and of testing aspects of CMMI. In: Münch, J., Vierimaa, M. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 263–277. Springer, Heidelberg (2006)
61. Brodman, J.G., Johnson, D.L.: A Software Process Improvement Approach Tailored for Small Organizations and Small Projects. In: Proceedings of International Conference on Software Engineering (ICSE 1997), pp. 661–662 (1997)
62. Paulish, D.J., Carleton, A.D.: Case Studies of Software Process Improvement Measurement. IEEE Computer 27(9), 50–57 (1994)
63. Succi, G., Benedicenti, L., Predonzani, P., Vernazza, T.: Standardizing the Reuse of Software Processes. Standard View - Supporting Article 5(2), 74–82 (1997)
64. Basili, V.R., Rombach, H.D.: Tailoring The Software Process To Project Goals and Environments. In: ICSE 1987. ACM, New York (1987)
65. Welzel, D., Hausen, H.-L., Schmidt, W.: Tailoring and Conformance Testing of Software Processes: the ProcePT Approach. In: Proceedings of the 2nd IEEE Software Engineering Standards Symposium, pp. 41–49 (1995)
66. Yoon, I. C., Min, S.Y., Bae, D.H.: Tailoring and Verifying Software Process. In: APSEC 200. IEEE Explore (2001)
67. Bowers, J., May, J., Melander, E., Baarman, M., Ayoob, A.: Tailoring XP for Large System Mission Critical Software Development. In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 269–301. Springer, Heidelberg (2002)
68. Xu, P., Ramesh, B.: A Tool for the Capture and Use of Process knowledge in Process Tailoring. In: Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS 2003), IEEE Computer Society, Los Alamitos (2002), 0-7695-1874-5/03 $17.00 ©
69. Xu, P.: Knowledge Support in Software Process Tailoring. In: Proceedings of the 38th Hawaii International Conference on System Sciences (2005)

70. Kang, D., Song, I.G., Park, S., Bae, D.H., Kim, H.K., Lee, N.: A Case Retrieval Method for Knowledge-Based Software Process Tailoring Using Structural Similarity. In: 15th Asia-Pacific Software Engineering Conference. IEEE, Los Alamitos (2008) doi: 1530-1362/08 $25.00 ©
71. Keenan, F.: Agile Process Tailoring and probLem analYsis (APTLY). In: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), pp. 45–47 (2004), 0270-5257/04 $20.00 © 2004 IEEE
72. Mirakhorli, M., Rad, A.K., Shams, F., Pazoki, M., Mirakhorli, A.: RDP Technique: a Practice to Customize XP. In: APOS 2008, Leipzig, Germany, pp. 23–32 (2008)
73. Murru, O., Deias, R., Mugheddo, G.: Assessing XP at a Europian Internet Company. IEEE Software 20(3), 37–43 (2003)
74. Drobka, J., Noftz, D., Raghu, R.: Piloting XP on Four Mission-Critical Projects. IEEE Software 21(6), 70–75 (2004)
75. Grenning, J.: Launching Extreme Programming at a Process Intensive Company. IEEE Software 18(6), 27–33 (2001)
76. Cao, L., Mohan, K., Xu, P., Ramesh, B.: How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 3, vol. 3 (2004)
77. Elssamadisy, A.: XP on a large project - A Developer's view. In: Proceedings of XP/Agile Universe, Raleigh, NC (2001)
78. Lui, K.M., Chan, K.C.C.: A road map for implementing eXtreme programming. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) SPW 2005. LNCS, vol. 3840, pp. 474–481. Springer, Heidelberg (2006)
79. Reifer, D.J.: How to get the most out of extreme programming/Agile methods. In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 185–196. Springer, Heidelberg (2002)
80. Aveling, B.: XP lite considered harmful? In: Eckstein, J., Baumeister, H. (eds.) XP 2004. LNCS, vol. 3092, pp. 94–103. Springer, Heidelberg (2004)
81. Ibraguengoitia, G., Salazar, J.A., Sanchez, M.G., Ramirez, A.Y.: A Procedure for Customizing a Software Process. In: Proceedings of the Fourth Mexican International Conference on Computer Science (ENC 2003), p. 68 (2003)
82. Oktaba, H., Gonzalez, G.I.: Software Process Modeled with Objects: Static View. Computacion y Sistemas 1(4), 228–238 (1998)
83. Bustard, D.W., Keenan, F.: Strategies for systems analysis: groundwork for process tailoring. In: ECBS 2005, pp. 357–362 (2005)
84. Park, S., Na, H., Park, S., Sugumaran, V.: A Semi-Automated Filtering Technique for Software Process Tailoring Using Neural Network. Expert Systems with Applications 30(2), 179–189 (2006)
85. Pedreira, O., Piattini, M., Luaces, M.R., Brisaboa, N.R.: A Systematic Review of Software Process Tailoring. ACM SIGSOFT Software Engineering Notes 32(3) (2007)
86. Dai, F., Li, T.: Tailoring Software Evolution Process. In: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 782–787. IEEE Computer Society, Los Alamitos (2007), doi:10.1109/SNPD.2007.25, 0-7695-2909-7/07 $25.00 ©
87. Xu, P., Ramesh, B.: Using Process Tailoring to Manage Software Development Challenges. IEEE Computer Society ITPro 10(4), 39–45 (2008)