A Systematic Review of Software Process Tailoring

Oscar Pedreira[1], Mario Piattini[2], Miguel R. Luaces[1] and Nieves R. Brisaboa[1]

[1]Database Laboratory, University of A Coruña
email: {opedreira, luaces, brisaboa}@udc.es

[2]Alarcos Research Group, University of Castilla-La Mancha
email: Mario.Piattini@uclm.es

Abstract

Although software process proposals appear continuously, it is difficult to fit any of them into a given company as they are. Thus, some kind of adaptation or tailoring is always necessary. The goal of software process tailoring is to adapt an "off-the-shelf" software process to meet the needs of a specific organization or project. Although process tailoring is a mandatory activity in most software process proposals, it is usually carried out by following an ad-hoc approach, and the amount of research done on this topic to date can be considered small. This paper presents a systematic review of software process tailoring, analyzing the existing approaches towards this activity, discussing the main issues related to the problem, and providing an up-to-date and complete framework in which to position new research activities.

Keywords: Systematic review, software process tailoring

1. Introduction

The software development process has always been one of the most important research topics for the software engineering community. A great number of proposals defining different approaches to successful software development appear continuously. However, it is unlikely that one of these "off-the shelf" approaches will fulfil the requirements of a specific project or organization. Bearing in mind that any two organizations are different, and furthermore, that two projects within the same organization may also be different, a process that can be successfully applied to one project may be a failure in another. Thus, the process must be adapted to the requirements and the specific context of the project.

Software process tailoring is the act of adjusting the definitions and/or of particularizing the terms of a general process description to derive a new process applicable to an alternative (and probably less general) environment [1]. That is, it is the act of adapting a standard software process to meet the needs of a specific organization or project. Software process tailoring can take place at two different levels: the *organizational level* and the *project level*.

The consequences of performing bad process tailoring are very important for the organization. First, the project budget, the project development time, and the product quality depend directly upon the quality of the software process. Second, a bad software process may involve unnecessary activities that lead to a waste of time and money, or the omission of those activities that are necessary, which may affect the product quality. Third, inappropriate process tailoring can cause the software process not to comply with the organizational standard process or with international standards such as ISO or CMM. Finally, the satisfaction of the employees is an important consideration that must be taken into account by process tailoring because nobody likes wasting time on unnecessary activities.

Nowadays, software process tailoring is considered a mandatory task by most process proposals, but this activity is usually done without the proper dedication, following an ad-hoc approach and using neither guidelines nor rules. Even though some systematic and formal methods for software process tailoring have been proposed, to date there has been little research carried out in this area. In the existing literature, one can find studies that propose support tools for software process tailoring. One can also find papers presenting real experiences of software process tailoring to adapt well-known "off-the-shelf" software processes to specific situations, which constitute an important contribution due to the practical nature of this problem. However, one problem with the case-study papers is that most of them focus on large software development companies with a formal and well defined set of software processes in a well organized environment.

In this paper we have performed a systematic review of software process tailoring research. To do this, we have gathered and analyzed the most important existing research carried out to date in an attempt to identify the approaches, the methodologies and the support tools proposed for software tailoring. Furthermore, we have gathered and analyzed case-study reports that describe process tailoring experiences in real software development organizations. Thus, this paper summarizes the most relevant existing evidence, and provides an up-to-date and complete framework in which to position new research activities.

The remaining part of the paper is structured as follows. The following section briefly describes the concept of systematic review in software engineering. Section 3 describes the decisions taken in each step of the systematic review presented in this paper and the results obtained. Section 4 presents the main issues concerning software process tailoring identified in literature. Finally, Section 5 presents our conclusions and future work.

2. Systematic review in software engineering

A systematic review is a research methodology developed to gather, evaluate and analyze all the available research relevant to a particular research question or area of interest [2]. In contrast with a conventional review of literature, a systematic review follows a well defined and strict sequence of methodological steps, which ensure the high scientific value of the results obtained. The main reason for performing a systematic review is to increase the probability of detecting more real results in the subject of interest

than those obtained by a less formal review. A systematic review requires a considerable effort in comparison with a traditional review of literature, but that is the price that must be paid for in depth and complete research in an area of interest.

The concept of systematic review appeared in the field of medicine, and its adaptation to software engineering is presented in [3]. The method proposed consists of three main activities: *planning*, *reviewing* and *publishing*. During the planning activity, the researcher identifies the review needs and develops the review protocol. In the reviewing activity, the most important studies are selected and evaluated. Then, the relevant data found in the papers are extracted and synthesized. Finally, the results of the review are published in the third activity. To make the systematic review easier, a template for the review protocol is proposed in [2], which is that which is followed in this paper.

3. A systematic review of software process tailoring

In this section, we present the decisions taken in each step of the systematic review and the results obtained. All of the assumptions and considerations made during the process are also explained.

3.1. Question formulation

The goal of this systematic review is to gather and analyze all the proposed tools, techniques, approaches and experiences for the tailoring of a software process to a specific purpose. Furthermore, all the research works that deal with questions related to software process tailoring and compliance with software engineering standards are also part of our goal. Finally, it is interesting to gather evidence of real experiences of tailoring well-known software processes (e.g., Rational Unified Process or eXtreme Programming) in software development companies. Previous systematic reviews have concentrated on more particular problems, but due to the small amount of research available in this field, we decided to gather all the available information about software process tailoring.

Unlike the systematic review template described in [2], our review must answer several questions and not only one due to the broad scope it aims to cover. Particularly:

- What are the main existing approaches, methods, and tools for software process tailoring?
- What are the better-known tailoring guidelines for software process tailoring and standard compliance?
- Are there any real case studies of software process tailoring available?

Our expectation when formulating these research questions was to be able to provide a comprehensive and broad report of the state of the art in software process tailoring at the end of this systematic review. We wanted not only to identify the main approaches in the field, but also its strengths and weaknesses, and, of course, future work that can be done to reduce these weaknesses.

From these research questions we can extract the keywords used to query the primary study sources. Some examples are the following: *software*, *process*, *tailoring*, *methodology*, *unified process*, *extreme programming*, *improvement*, *organization*, *practice*, *experience*, *case study*, *standard*, *compliance*, *ISO*, *CMM*. For the search of primary studies we considered it important to check whether synonyms of the word *tailoring* such as *adaptation* or *customization* are also used, as happens with *method* as a synonym of *process*, because this may be a source of problems.

3.2. Sources selection

From the research questions presented above and the list of keywords, we built the search strings used in the review. The basic search string is "*software process tailoring*". We also used additional search strings including the other proposed keywords. The string "*(software AND (process OR method) AND (tailoring OR adaptation OR customization))*" will show whether the synonyms previously identified affect the search. Finally, search strings such as "*software process tailoring AND (practice OR experience OR organization)*" were used with the intention of revealing possible works about case studies in real software development companies.

The search for primary studies was performed in the digital libraries of the most famous publishers and organizations in software engineering. The list of sources is the following: IEEE Computer Society Digital Library, ACM Digital Library, Wiley InterScience (computer science area), Science@Direct (computer science area), and SpringerLink. Of course, before starting the review we already had some studies about software process tailoring. Perhaps two of the most important works we had are [7] and [1]. Both of them present a structured and well-designed framework for software process tailoring without supposing the use of any software process in particular. So, both of them were good starting points and were directly included in the review.

3.3. Primary study selection

For the selection of primary studies, the digital libraries cited above were queried using the search strings presented above. The query "*software process tailoring*" returned most of the studies we selected. The other strings used returned only two additional studies that were not retrieved by the previous search due to the problem of the synonyms 'tailoring' and 'adaptation'. The other strings did not return additional primary studies. This proves that little research has taken place in this area.

Most of the results returned by the sources were not of interest for this review. The inclusion and exclusion criteria used to select the relevant primary studies were applied to the title and abstract of each study. In a few cases this was not enough and the full text of the study had to be partially reviewed to ensure that it was relevant to our research goal. Of course, we also took into account that some results appeared in more than one source, so repeated studies were identified and controlled.

After querying the information sources, 394 non-repeated results were obtained. After evaluating all of these sources by applying the inclusion/exclusion criteria, only 28 were considered relevant primary studies for the research questions of this review. Table 1 shows the total number of results and primary studies obtained from each source. The complete result list is too long to be included in this paper. It can be obtained by contacting any of the authors. The list of selected primary studies is included in the reference section of this paper. The number of primary studies is quite small in comparison with the results obtained in previous

systematic reviews. However, the small number of primary studies is not a bad result for the systematic review because its final goal is to find and analyze all the available research in this area. On the contrary, it can be viewed as evidence that there is little research available in this field to date and that there is still some work to be done in software process tailoring.

Source	Total results	Primary studies
ACM	200	3
IEEE	100	14
Springer	31	6
Science@Direct	10	2
Wiley	50	0
Others	3	3
Total	**394**	**28**

Table 1. Number of results and selected primary studies obtained from each source.

An interesting observation is, perhaps, that two of the studies that passed the inclusion/exclusion criteria during the information extraction phase were not really about software process tailoring. However, only two mistakes in a set of 394 papers can be considered a good result.

3.4. Information extraction

Once the primary studies were identified and obtained, the next step consisted of their review and information extraction. For each of them, a review summary such as that proposed in [2] was written describing the methodology followed by the authors of the primary study, the main results they achieved, possible problems found in the study, and a subjective evaluation of the study. This evaluation usually included a small summary with a comparison of the results obtained in that study with those obtained in others which were similar.

4. Issues related to software process tailoring

This section presents the answers to the questions formulated for the systematic review. During the evaluation and analysis of the primary studies we found the following relevant issues related to software process tailoring:

- Software process tailoring can take place at different levels within the organization (e.g., in the organization, or in a single project).
- The approach for process tailoring can be formal or informal
- Presence of a case study in a real company
- Size of the company where the approach is applied.
- Presence of a study standard compliance in process tailoring.
- Description of support tools for the tailoring activity.

Each study may deal with more than one of these issues. For example, some papers describe a formal approach for software process tailoring and real experience in applying their proposal in a real situation, but focus only on the project level process tailoring.

Figure 1 shows the percentage of papers which focused on formal vs. informal tailoring, company or project level, and small or large companies. This figure gives an idea of the effort devoted to each problem. For example, most of the reviewed papers describing a case study focus on large software development organizations rather than small and medium enterprises.

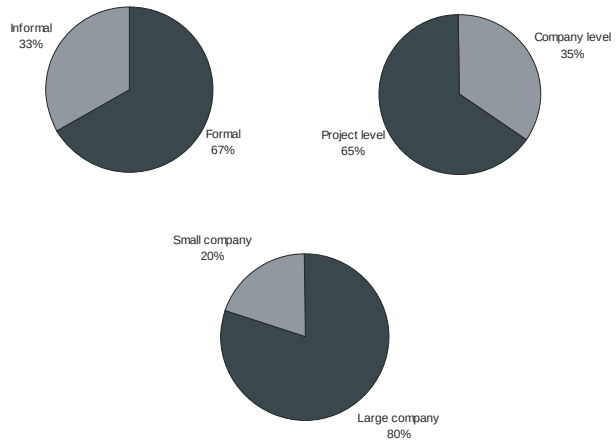

Figure. 1. Number of studies covering each topic.

4.1. Different levels for software process tailoring

Software process tailoring can take place at different levels. For example, some studies [4, 5, 6] distinguish between process tailoring at the organizational level and at the project level. The goal of process tailoring at the organizational level is to adapt an "off-the-shelf" industry standard software process to meet the needs of a specific organization. The resulting process is the Organizational Software Standard Process (OSSP) [4]. This OSSP is completely adapted to the needs and environment of each individual company. Some elements of the standard software process are usually removed, but sometimes some other elements are added due to the type of software that the company develops.

However, this is not usually enough and software process tailoring is also necessary at the project level. That is, the OSSP must be tailored to meet the needs of a specific project. Considering that within any given company one project may be very different from another, a process which is successfully applied to one of them may give bad results in the other. As pointed out by [4], failure to align the OSSP with the project context can result in bad effects on the product development time, cost and quality. This problem is greater for companies in which projects are big, costly, and very different from each other. Hence, in this case, software process tailoring at the project level is a mandatory activity. For this type of organizations [7] is a good example of tailoring guidelines to meet the needs of project plans. This approach also allows the use

of a completely different approach for software development in each project if necessary.

As we can see in Figure 1, most of the studies reviewed consider software process tailoring only at the project level. However, tailoring at the organizational level is also necessary, and the tailoring rules can be quite different in this case. Furthermore, if clearly different project types can be found in a company, another level of process tailoring can be considered. In this case, the OSSP can be tailored to each of these project types. However, only [8] distinguishes this level of tailoring.

As a final conclusion, the detail of tailoring at the organizational and project levels depends on the size of the company and its projects. Software process tailoring at the organizational level is always mandatory. For some companies with clearly different product lines, a tailoring process for each line may be applied if necessary. Finally, as mentioned in [8], tailoring can take place at the project level only if the project is very large and the adaptation represents a small amount of the total development time.

4.2. Formal vs. informal approaches for software process tailoring

The differentiation between formal or systematic process tailoring versus informal tailoring is also important. Most of the papers reviewed propose or use some kind of formal approach, rules or guidelines to systematically tailor a standard process. However, many of the studies reviewed describe only the result of informal tailoring for a specific situation. The degree of formality in the different approaches proposed may be very different from one to another. For example, the approach proposed for process tailoring in [4] is not as formal as those proposed in [7] or [9].

Studies like [4] provide a systematic and ordered method for process tailoring which is very complete. The main advantage of systematic software process tailoring is that the result does not depend so much on the skills of those responsible for this task, nor on his/her own preferences for a software process. It can also be applied by practitioners who are not very experienced. This approach may be a good option for large software organizations which can assume this grade of formality. However, for a small company an approach with a high level of formality may be excessive. For example, an experience in a small company is presented in [8] and concludes that in this situation the adaptation is best done as a simple, pragmatic process and not as a planned and strictly managed process.

4.3. Case studies: experiences in real organizations

Some of the studies reviewed present their proposal with the results obtained after having applied it in a real software organization [4, 10]. Others describe a real experience of tailoring following an informal approach for this activity. Most of the case study papers following an informal approach describe their experience in the adaptation of the Unified Process or eXtreme Programming to specific situations. For example, [11] and [12] report their experiences in adapting XP to large/complex projects in large software development organizations, while [13] presents an experience in the use of a tailored Unified Process.

As is pointed out in [6], the absence of practice-based research in software development, and in method tailoring in particular, is surprising in an applied field. Case study has proved to be a useful tool in other fields. The study of real experiences is a good source of information for the development of new approaches.

Another problem encountered in the practical evidence of process tailoring is that most of the studies devote more attention to the resulting process than to the guidelines and methods for process tailoring.

4.4. Software process tailoring in large vs. small companies

This is, perhaps, one of the most interesting results derived from the review. Most of the primary studies propose a tailoring method defined for large and complex software organizations. For example, [5, 6, 11] describe the software process tailoring approach followed in a large telecommunications company; [9] presents a quantitative methodology for software process tailoring in a high-risk environment such as NASA/SEL; and [14, 15] do the same for the Raytheon laboratories.

Process tailoring is very important for this kind of organizations, but these companies are certified at least as CMM Level 3 and they are able to assimilate complex process tailoring better than a small company. However, the approaches they follow are too extravagant for small and medium companies. Software process tailoring is usually a difficult activity for small companies because they do not have a well-organized software development environment and because the experience of their software engineering practitioners is more limited.

Some works focus on the case of small companies. [8] report that in their experience, process tailoring in small companies is best done as a simple and pragmatic process, and not as one which is over-extravagant and strict. [10] is a really interesting paper which focuses on small companies, proposing a semi-formal but lightweight method for process tailoring, and reporting a real experience in two companies with quantitative results about the resulting process. Perhaps this is one of the important gaps in software process tailoring research and more effort should be devoted to this kind of organizations.

4.5. Software process tailoring and standard compliance

Software process tailoring presents an interesting problem with regard to standard compliance. In organizations certified by ISO or CMM, the software development process and the tailoring activities must ensure that the resulting process used in each project still conforms to the organization process definition.

Process tailoring is a mandatory activity in CMM and [1] provides a complete framework and guidelines for software process tailoring in CMM-certified organizations. This is one of the most complete studies about software process tailoring and one of the few that covers the problem of standard compliance. Of all the papers reviewed only one other covers the problem of standard compliance [16]. This study proposes a strictly formal framework for software process tailoring, and introduces the idea of automatically verifying the resulting process by measuring the number of dependencies between process modules that the tailored process preserves.

4.6. Support tools for software process tailoring

The use of supporting tools can facilitate the activity of software

process tailoring. Studies such as [17] and [18] define software process tailoring as a knowledge-intensive activity, and analyze the benefits of knowledge management in this task. They distinguish between the use of general knowledge about process tailoring and the use of contextualized knowledge about previous experiences in the company. As is shown in these studies, the use of this kind of knowledge can be very helpful, mainly for inexperienced practitioners who have to deal with this important task.

A different tool is proposed in [19] which is based on the use of neural networks for semi-automated software process tailoring using historical data as learning data for their system. Therefore, the tool described in this study uses contextualized knowledge in the process tailoring activity. However, the lack of previous experiences may be a problem for the system's behaviour.

5. Conclusions and future work

This paper presents a systematic review of software process tailoring which gathers and analyzes the most important existing research carried out in this topic. The paper provides an up-to-date framework in which to position new research activities. The systematic approach followed when conducting the review ensures the completeness of the results obtained.

One of the most important problems we discussed is the degree of formality in the process tailoring activity. Some of the primary studies propose a formal approach that proved successful in real experiences. However, all these experiences took place in very large and advanced software development organizations which can cope with this formality. When applied to smaller organizations, this approach may be excessive. As a consequence, these companies usually confront process tailoring by following an ad-hoc approach, which makes the resulting process highly dependent upon the preferences and skills of the person responsible. As a result, the software process may not be adequate for the company's characteristics.

In conclusion, the most important need that we identified in software process tailoring is the small amount of attention paid to small and medium sized companies. More effort must be devoted to the development of a general framework for process tailoring which is applicable in a broader range of organizations.

In our opinion, standard compliance is another area in which more work must be done. The problem of standard compliance is vitally important for ISO or CMM certified organizations. As deviation from the defined process may result in grave problems for their certification, the lack of a good process tailoring framework may sometimes force them to follow an excessive process containing unnecessary activities. The research available in this area is extremely limited, so it is still an open problem.

One of the problems we found is that the existing approaches for software process tailoring were defined for specific environments. Thus, our future work in this line of research involves the development of a complete and general framework for process tailoring, which is applicable to a broad range of companies.

Acknowledgements

This work has been partially supported by "Xunta de Galicia" grants PGIDIT05SIN10502PR and 2006/4, and "Ministerio de Educación y Ciencia" (PGE y FEDER) grant TIN2006-15071-C03-03 (for Oscar Pedreira, Miguel Luaces and Nieves Brisaboa); and the project COMPETISOFT (CYTED- 506PI0287) and the ESFINGE project supported by the "Ministerio de Educación y Ciencia (Spain) and FEDER" (TIN2006-15175-C05-05) (for Mario Piattini).

References

[1] Ginsberg, M., Quinn, L. (1995): Process tailoring and the software Capability Maturity Model. Technical report, Software Engineering Institute (SEI).

[2] Biolchini, J., Mian, P.G., Natali, A.C.C., Travassos, G.H. (2005): Systematic review in software engineering. Technical report, Systems Engineering and Computer Science Department, Rio de Janeiro.

[3] Kitchenham, B.(2004): Procedures for performing systematic reviews. Technical report Software Engineering Group, Department of Computer Science, Keele University.

[4] Nanda, V. (2001): On tailoring an organizational standard software development process for specific projects. In: Proceedings of the 11th International Conference on Software Quality, pp. 1-13.

[5] Fitzgerald, B., Russo, N., OKane, T. (2000): An empirical study of system development method tailoring in practice. In: Proceedings of the Eighth European Conference on Information Systems, pp. 187-194.

[6] Fitzgerald, B., Russo, N., O'Kane, T. (2003): Software development method tailoring at Motorola. Communications of the ACM 46(4), pp. 65-70.

[7] Budlong, F., Szulewski, P., Ganska, R. (1996): Process tailoring for software project plans. Technical report, Software Technology Support Center of the U.S. Air Force.

[8] Hanssen, G.K., Westerheim, H., Bjrnson, F.O. (2005): Tailoring RUP to a defined project type: A case study. In: Product Focused Software Process Improvement: 6th International Conference, PROFES 2005, pp. 314-327. Lecture Notes in Computer Science (3547/2005), Springer.

[9] Basili, V., Rombach, D. (1987): Tailoring the software process to project goals and environments. Communications of the ACM, pp. 345-357.

[10] von Wangenheim, C.G., Weber, S., Hauck, J.C.R. (2000): Experiences on establishing software processes in small companies. Information and Software Technology 48, pp. 890-900.

[11] Bowers, J., May, J., Melander, E., Baarman, M., Ayoob, A. (2002): Tailoring XP for large system mission critical software development. In: XP/Agile Universe 2002: Second XP Universe and First Agile Universe Conference, pp. 100-111. Lecture Notes in Computer Science (2418/2002), Springer.

[12] Cao, L., Mohan, K., Xu, P., Ramesh, B. (2004): How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. In: Proceedings of the 37th Hawaii International Conference on System Sciences – 2004, pp. 1-10.

[13] Westerheim, H., Hanssen, G.K. (2005): The introduction and use of a tailored unified process a case study. In: Proceedings of the 2005 31st EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA05), pp. 196-203.

[14] Oshana, R. (1998): An industrial application of cleanroom software engineering – benefits through tailoring. In: Thirty-First Annual Hawaii International Conference on System Sciences-Volume 6, pp. 122-131. IEEE Press.

[15] Oshana, R.S. (1998): Tailoring cleanroom for industrial use. IEEE Software (1998) pp. 46-55.

[16] Yong, I.C., Min, S.Y., Bae, D.H. (2001): Tailoring and verifying software process. In: Eighth Asia-Pacific Software Engineering Conference (APSEC'01), pp. 202- 209, IEEE Press.

[17] Xu, P., Ramesh, B. (2003): A tool for the capture and use of process knowledge in process tailoring. In: Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS03). IEEE Press.

[18] Xu, P. (2005): Knowledge support in software process tailoring. In: Proceedings of the 38th Hawaii International Conference on System Sciences – 2005, pp.1-9.

[19] Park, S., Naa, H., Parka, S., Sugumaranb, V. (2006): A semi-automated filtering technique for software process tailoring using neural network. Expert Systems with Applications 30, pp. 179-189.

[20] Kitchenham, B.A., Peeger, S.L., Hoaglin, D.C., Emam, K.E., Rosenberg, J. (2002): Preliminary guidelines for empirical research in software engineering. IEEE Transactions on Software Engineering 28(2), pp. 721-734.

[21] Hikichi, K., Fushida, K., Iida, H., ichi Matsumoto, K. (2006): A software process tailoring system focusing to quantitative management plans. In: Product-Focused Software Process Improvement, 7th International Conference, PROFES 2006, pp. 441-446, Lecture Notes in Computer Science (4034/2006), Springer.

[22] Brodman, J.G., Johnson, D.L. (1997): A software process improvement approach tailored for small organizations and small projects. In: Proceedings of the International Conference on Software Engineering 1997 (ICSE 97).

[23] Keenan, F. (2004): Agile process tailoring and problem analysis (aptly). In: Proceedings of the 26th International Conference on Software Engineering (ICSE04).

[24] Kim, S.Y., Choi, H.J. (2005): An evaluation of process performance for a small-team project -a case study. In: Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS05).

[25] Bustard, D.W., Keenan, F. (2005): Strategies for systems analysis: Groundwork for process tailoring. In: Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS05), pp. 357- 362, IEEE Press.

[26]. Hollenbach, C., Frakes, W. (1996): Software process reuse in an industrial setting. In: Fourth International Conference on Software Reuse (ICSR'96), pp. 22-30, IEEE Press.

[27] Lobsitz, R.M. (1996): A method for assembling a project-specific software process definition. In: 29th Hawaii International Conference on System Sciences (HICSS'96) Volume 1: Software Technology and Architecture, pp. 722-730, IEEE Press.

[28]. Welzel, D., Hausen, H.L., Schmidt, W. (1995): Tailoring asrid conformance testing of software processes: The ProcePT approach. In: 2nd IEEE Software Engineering Standards Symposium, IEEE Press.

[29] Henninger, S., Baumgarten, K. (2001): A case-based approach to tailoring software processes. In: Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR 2001, pp. 249, Lecture Notes in Computer Science (2080/2001), Springer.

[30] Mnkandla, E., Dwolatzky, B., Mlotshwa, S. (2005): Tailoring agile methodologies to the southern african environment. In: Extreme Programming and Agile Processes in Software Engineering, pp. 259-262, Lecture Notes in Computer Science (3556/2005), Springer.

[31] Pikkarainen, M., Salo, O. (2006): A practical approach for deploying agile methods. In: Extreme Programming and Agile Processes in Software Engineering., pp. 213-214, Lecture Note in Computer Science (4044/2006), Springer.