

# You can't tailor what you haven't modeled

Marco Kuhrmann  
Technische Universität München, Faculty of Informatics  
Munich, Germany  
[kuhrmann@in.tum.de](mailto:kuhrmann@in.tum.de)

## ABSTRACT

It is widely accepted that the one size fits all process does not exist. Software processes need to be tailored according to the respective context of companies and projects. However, tailoring a software process often remains a mystery. What is the actual context? What are the parameters to adjust a process? What are the implications of tailoring criteria? A systematic process tailoring requires the ability to anticipate needed flexibility early in the process design process, and to express this in a process modeling language. In this paper, we discuss the design of process tailoring models, which we consider crucial for the design and, eventually, the application of flexible software processes. We advocate for a constructive metamodel-based approach to improve process tailoring.

## Categories and Subject Descriptors

D.2.9 [Software Engineering Management]: Software process models

## General Terms

Management, Experimentation

## Keywords

software process, software process tailoring, flexible software processes, position paper

## 1. INTRODUCTION

Tailoring is defined as “the act of adjusting the definition and/or particularizing the terms of a general description to an alternate environment” [2]. Some research addresses the identification of criteria considered relevant to trim a process to fit it into the respective company- or project context, e.g., [10, 3, 6]. However, we still have little knowledge about the selection of process assets required to address certain project settings. Although rich software process frameworks, such

as SPEM or the V-Modell XT, provide various instruments to design tailoring models in order to support flexible software processes, tailoring criteria as reported in literature are often not implemented in such standard frameworks. In [3], we analyzed tailoring criteria and their impact on project operation—we could not find indication regarding a correlation. Taking into account further research [5, 6, 7], we concluded that we lack the ability to capture required flexibility of software processes due to a missing ability to express flexibility using today's process modeling languages.

In this paper, we stress the point of (formally) modeling software process tailoring. We contribute a discussion on requirements that modern software process frameworks should fulfill to better support flexible software processes. For this, in this position paper, we discuss the context in which a software process is customized, and the elements needed for designing flexible software processes.

## 2. PROCESS TAILORING AND CONTEXT

When it comes to selecting tailoring criteria, the one thing that matters the most is *context*. For example, given a process to be used in a project, and the process shall be tailored to the size of the project team. A provided tailoring criterion ‘team size’ has the values *large*, *medium*, and *small*. Which value must be selected? The answer would be “it depends.” In fact, selecting a value for this criterion depends on many other context variables, e.g., size and complexity of the software to be developed, time constraints, personnel availability, and team distribution pattern (co-located or distributed). Moreover, several context variables are interdependent themselves, e.g., project size and team size, or compliance requirements and structuredness of the process. Knowledge regarding the context is crucial to understand process tailoring and to define appropriate tailoring criteria. Without proper context definition, we fail in predicting the impact of tailoring criteria on projects [3].

In order to support a meaningful tailoring, some work needs to be done upfront. Process engineers have to anticipate the required flexibility, and they have to design the process properly. For this, process modules have to be designed, process configurations need to be defined, and, finally, adequate tailoring constructors [7] must be provided. Figure 1 illustrates the relationship of these basic requirements (this figure is *not* meant to be a metamodel proposal). The key element is the *process asset*, which abstracts from particular process elements, e.g., activities, work products, and roles. Process assets are composed in tailoring modules (the simplest module contains one asset), which are subject to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICSSP '14, May 26–28, 2014, Nanjing, China

Copyright 2014 ACM 978-1-4503-2754-1/14/05 ...\$15.00.

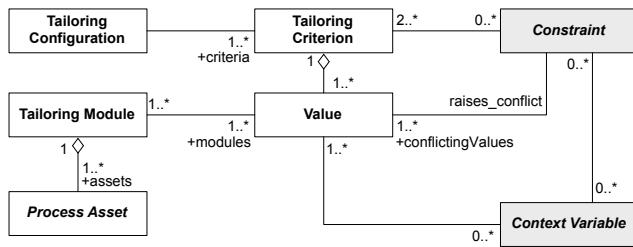


Figure 1: Key elements to build tailoring models.

tailoring. Tailoring as such relies on tailoring criteria, more specific, on the *values* of tailoring criteria. Each value refers to process modules and, thus, a particular tailoring configuration is characterized by a set of values. To fully understand tailoring, further components need to be considered: tailoring criteria may depend on each other. Therefore, certain values of tailoring criteria may raise conflicts, and, thus, imply *constraints*, e.g., cost optimization contradicts Follow-the-Sun strategy. Furthermore, selecting a particular criterion depends on the actual context. Thus, a particular *context variable* can limit the range of values. Moreover, the definition of values as such is demanding. For example, a critical question is for the scale, e.g., ordinal/discrete or continuous.

For example, in the Method Engineering domain, several contributions deal with modeling context. Wistrand and Karlsson [9] use a goal modeling technique to relate so-called method components, Karlsson and Ågerfalk [4] link project characteristics to an explicitly defined purpose, Engels and Sauer [1] introduce constraints as selector for methods, and Rolland [8] proposes a definition of the term context, which is based on pairs (*situation, decision*). The problem with these proposals is that we barely find empirical evidence on their use and feasibility. Furthermore, no existing and disseminated (standard) process engineering framework directly implements these concepts. What we find instead are simple dependencies between process modules (if chosen *A*, you also need *B*), and rather coarse-grained project characteristics, e.g., in the V-Modell XT.

### 3. CONCLUSION

Process tailoring is a critical activity. In this position paper, we briefly discussed context and its inclusion in tailoring models.

We conclude that process tailoring is, in some aspects, well understood. However, although having approaches available that allow for designing flexible processes in general, we still lack understanding regarding context and the criteria that influence software processes. Critical is our ability to express the required flexibility in today's process frameworks. For process engineers, it is crucial to understand context, criteria, and the impact on the software process in order to anticipate project situations and thus getting a notion of the required flexibility.

For this, we vote for putting more emphasis on explicitly modeling software process tailoring. Otherwise, tailoring is left to the expertise of project managers undermining the repeatability of software processes across development projects [10]. For instance, as agile methods are usually not formally modeled, further research is necessary to investi-

gate, e.g., when and why a particular practice is selected, and how selected practices can be assembled in a meaningful manner.

Software process metamodels should be enriched by (improved) tailoring models, e.g., by making tailoring criteria variable process assets to support improvement and refinement based on project experiences. Furthermore, process metamodels should be enriched by language constructs to support context modeling.

In order to clarify the notion of context, and to allow for designing meaningful tailoring criteria, the process community should put emphasis on investigating the triple (*context, criterion, impact*), and on the dependencies between tailoring criteria and the appropriateness of tailored processes.

As we believe that we cannot tailor a software process without a meaningful tailoring model, we formulate a hypothesis to be subject to further investigation: *A precisely defined tailoring model (incl. context and parameters) allows for the definition of better (e.g., appropriateness, precision, and validity) project-specific process models compared to an ad-hoc tailoring, and supports a more efficient project operation (e.g., by enhancing decision-making processes).*

### 4. REFERENCES

- [1] G. Engels and S. Sauer. A meta-method for defining software engineering methods. In *Nagl Festschrift*, number 5765 in Lecture Notes in Computer Science, pages 411–440. Springer Berlin Heidelberg, 2010.
- [2] M. Ginsberg and L. Quinn. Process tailoring and the software capability maturity model. Technical Report CMU/SEI-94-TR-024, Software Engineering Institute, Carnegie Mellon University, 1995.
- [3] G. Kalus and M. Kuhrmann. Criteria for software process tailoring: A systematic review. In *International Conference on Software and System Process (ICSSP)*, pages 171–180. ACM Press, 2013.
- [4] F. Karlsson and P. J. Ågerfalk. Method configuration: adapting to situational characteristics while creating reusable assets. *Information and Software Technology*, 46(9):619–633, July 2004.
- [5] M. Kuhrmann, D. M. Fernández, and R. Steenweg. Systematic software process development: Where do we stand today? In *International Conference on Software and System Process (ICSSP)*, pages 166–170. ACM Press, 2013.
- [6] T. Martínez-Ruiz, F. García, M. Piattini, and J. Münch. Modelling software process variability: an empirical study. *IET Software*, 5(2):172–187, 2011.
- [7] T. Martínez-Ruiz, J. Münch, and M. Piattini. Requirements and constructors for tailoring software processes: A systematic literature rewwiew. *Software Quality Journal*, 20(1):229–260, March 2010.
- [8] C. Rolland. A primer for method engineering. CREWS Report Series 97-06, Université de Paris, 1997.
- [9] K. Wistrand and F. Karlsson. Method components – rationale revealed. In *International Conference on Advanced Information Systems Engineering*, number 3084 in Lecture Notes in Computer Science, pages 189–201. Springer Berlin Heidelberg, 2004.
- [10] P. Xu and B. Ramesh. Using Process Tailoring to Manage Software Development Challenges. *IT Professional*, 10(4):39–45, July 2008.