# A Framework for Method Tailoring: A Case Study

Chaitali Patel[1], Sergio de Cesare[1], Nicola Iacovelli[2], Antonio Merico[2]

[1] Brunel University,
Uxbridge, Middlesex,
UB8 3PH, United Kingdom
{chaitali.patel | sergio.decesare}@brunel.ac.uk
[2] Svimservice S.p.A.,
Via Massaua 18,
Bari, Italy
{nicola_iacovelli | antonio_merico}@svimservice.it

**Abstract.** Software development methodologies (SDM) have been traditionally defined in a prescriptive manner with an underlying assumption of universal applicability. However, as industrial practice suggests, this assumption is fundamentally flawed. Software development projects very rarely adopt a methodology in such a rigid fashion. Conversely methodologies are normally adapted to meet specific contextual characteristics. This adaptation, known as Method Tailoring (MT), generally occurs implicitly. Implicit adaptation has several drawbacks. Firstly, responsibility and consequences are not attributable to the decisions made during MT. Secondly, MT experience is not captured, thus not being shared and reused within the organization. As a consequence, implicit MT leads to reactive rather than proactive adaptation with negative effects on both productivity and efficient use of resources. In order to alleviate the problems described above, this paper proposes a framework aimed at assisting software development teams and organizations in the elicitation of their Method Tailoring processes. As a result the know-how and experience accumulated during the practice of Method Tailoring is made explicit and organized for the benefit of future projects. The framework has been applied a posteriori to a project carried out by a medium-sized software development company for the Italian national public health service.

## 1 Introduction

Software development methodologies (SDM) have been traditionally defined in a prescriptive manner with an underlying assumption of universal applicability. However, as industrial practice suggests [1-5], this assumption is fundamentally flawed. Software development projects very rarely adopt a methodology in such a rigid fashion. Initially introduced as a means to alleviate the problems of the software crisis, SDM manifested serious limitations due to the lack of any predisposition to flexibly adapt to the needs of specific projects, domains and organizational settings. As a consequence, significant undesired effects were generated. Such effects include lack of a political and organizational dimension, goal displacement, developers' resistance to change, and inhibition of creative thinking [6, 7]. In recent years the limitations of

prescriptive methodologies have been recognized by both academics and practitioners. The need to tailor methods to specific situations and context is now seen as imperative [1, 8-11]. The key issue in methodological support to software development today is flexibility and adaptation of methodologies. In response to these problems and issues, at least four different solution streams can be identified.

Firstly, a new methodology can be defined from scratch each time it is considered necessary. This represents an extreme solution and probably the less timely and most expensive [12, 13]. In addition there is a high risk of reinventing the wheel with each methodology without harvesting the benefits of reusable knowledge and experience.

Secondly, a methodology can be selected from those available on the market or in the published literature. This is known as the Contingency Approach. This solution preserves the rigidity and inefficiencies of the previous option; moreover, detailed knowledge of a wide range of methodologies would be needed in order to adequately make a sound judgment with significant effects on the cost of resources and training. Teams with specialist skills on individual methodologies would be required. Given the specific characteristics of each methodology (e.g., paradigm, notations and terminology), the knowledge transfer would be difficult as well as the allocation of resources. In addition, many authors also argue that existing rigid methods cannot adequately cover all contingencies [4, 12, 14].

Thirdly, adaptation and flexibility can be achieved through mixing and matching parts (i.e., method fragments) of different methodologies [12, 13, 15]. This is known as Method Engineering. This solution has the benefit of not constraining the development team with a predefined solution; although effort needs to be placed on harmonizing method fragments, e.g., different notations, techniques and terminology [13]. When fragments are derived from methodologies with different underlying paradigms, harmonization becomes particularly problematic and may include, for example, paradigm transformations and mappings in order to conserve desired levels of integration and traceability [9, 10, 16]. As a result, Method Engineering has not been popular in practice.

The fourth solution stream is represented by Method Tailoring (MT). MT refers to the adaptation of one methodological framework to specific software development projects [1, 7, 17, 18]. Past studies show that in practice, organizations develop or adopt one SDM organization-wide and then tailor the method to specific projects. In particular Guimaraes' [19] study found that 77% of the firms that uses SDM employ a single, formal methodology. An essential condition for MT to be effective is the adoption of a non-prescriptive and flexible methodology. The evolution of SDM has proceeded in this direction [11, 20]. Currently methodologies like the Rational Unified Process (RUP) and Object-Oriented Process, Environment and Notation (OPEN) Process Framework (OPF) do not prescribe the use of a particular process, but define a set of process components which can be selected and chosen from to suit certain project/organizational characteristics [20]. The difference with the previous solution is the availability of process components (or method fragments) developed with the same underlying paradigm. Problems of consistent and coherent mixing and matching of method fragments (e.g., process components) are significantly reduced. Consistency in notation and terminology increases the level of knowledge sharing and reuse across teams leading to easier resource allocation. Method tailoring represents a balanced solution. It maintains the benefits of 'standardization' while it allows for controlled flexibility and adaptation to specific contexts.

MT, although in a primitive form, has been conducted implicitly by practitioners even earlier than the research literature has recognized the significance of the problem area [1]. Software developers and project managers have instinctively and sometimes subconsciously carried out tailoring in one form or another [7]. However, various problems arise when adaptation of methods is conducted implicitly [21]. Firstly, responsibility and consequences are not attributable to the decisions made during MT. Secondly, MT experience and rationale for selecting/adapting method fragments is not captured, thus not being shared and reused within the organization. As a consequence, implicit MT leads to reactive rather than proactive adaptation with negative effects on both productivity and efficient (re)use of resources.

Modern methodologies currently provide the necessary building blocks to assemble a tailored process hence suggesting some form of contingent adaptation, besides flexible methodologies, software developers require practical guidance in MT [7, 15]. The limited literature on the topic provides examples of theoretical frameworks that can be applied to this problem; although the frameworks themselves have not been empirically validated. The aim of this paper is to bring together the MT related literature in order to identify underlying commonalities and define a framework for assisting software development teams and organizations in the elicitation of their Method Tailoring processes. As a result the know-how and experience accumulated during the practice of Method Tailoring is made explicit and organized for the benefit of future projects. The proposed framework is applied to a real-world industrial project as a means of capturing the tailoring process used.

The paper is structured as follows. Section 2 briefly reviews the MT related literature from which common elements are derived and used to construct the framework presented in Section 3. Section 4 presents the case study and applies the framework to it. Section 5 presents final conclusions.

| Name of Framework | Framework based on following Observations | Different terminology used for describing each component of the framework | | | | |
|---|---|---|---|---|---|---|
| | | Context | Method Fragments | Experience Capture | MT Process | Tailored process |
| The process of configuration of situational methods from Harmsen, F., S. Brinkkemper, and H. Oei [13] | Methods are never followed literally; they are tuned to the situation at hand.<br>Knowledge and experience of the project team determine the structure of the development process and the resulting products in order to deliver the desired IS.<br>All kinds of project factors related to the technology, the development expertise, the external factors and application domain characteristics influence an approach suitable for the project.<br>Large parts of the new methods are taken from old methods | Project environment, Characterization of project | Method fragments, Method administration, Method Base | Project performance, Experience accumulation | Selection of fragments, Assembly of fragments, Validation, Request for adaptation | Situational methods |
| Components of a social process for Method fragment adaptation from Baskerville, R. and J. Stage [23] | Methodologies are not a true representation of how systems are developed in practice.<br>Systems development process are emergent; they are only transient regularities in work place that are constantly shifting form.<br>We must improve our understanding of and means to support, the way in which development is conducted in practice. | Work setting, set of determinants of fragment selection | Method Fragments, Innovated method fragments, | Capture practice | On going accommodation (Select, invent and combine) | Work practices |
| Methodology for tailoring SDMs to projects and refining the SDM from Henninger, S., et al.[18] | There is a need to create flexible software processes so organizational methodologies can be tailored to individual needs of projects and capture experiences that are used to refine and modify the standards.<br>There is also a need to transform the methodologies into resources for managers and developers, to truly support the development process as it is actually practiced.<br>The framework is based on organizational learning principles to capture and provide relevant development knowledge throughout the development lifecycle. | Elicit project characteristics | Software development resources (process framework, Tailoring criteria, Experienced bases repository, Guidelines and standards) | Experienced packages, Lesson learned | Tailored project activities | Software creation (Tailored process used, Team level review) |
| A framework for ISD Method Use from - Fitzgerald, B., N.L. Russo, and E. Stolterman [7] | Methodologies have several pressures that support or are against the use of methodologies.<br>Formal methodologies are not used in practice as prescribed deliberately as they do not fit the specific project situation.<br>Methodologies play various roles in the organization which influences the actual way methodologies are used in practice. | Development context: Problem situation and Business Opportunity, | Original formalized methodology | Developers' experiences and expertise, repertoire of strategies. | Profile of development environment, developer embodied factors and roles of methodology in practice all influences method in action | Method in Action |

**Table 1: Comparative Analysis of Method Tailoring Frameworks.**

## 2 Literature Review on Method Tailoring

Method Tailoring frameworks serve a dual purpose. Firstly, they define the fundamental components required for tailoring SDM. Secondly, frameworks represent a means of eliciting the tailoring process applied by software development teams and organizations in industrial projects. This section carries out a brief comparative analysis of MT frameworks derived from the literature. The analysis is aimed at synthesizing previous research work in order to highlight common issues, components and relationships that will form the basis of the framework presented in section 3.

Among the few MT frameworks that have been developed the following are considered: Harmsen, Brinkkemper et al. [9], Baskerville and Stage [18], Henninger, Ivaturi et al. [19] and Fitzgerald, Russo et al. [7]. As shown in Table 1, the analysis reveals a high degree of overlap between the frameworks, both in terms of observations and components.

The common MT components, which will be detailed in the section 3, are:

- Context - Contextualization of the project environment,
- Method Fragments - Maintenance of method fragments
- Method Tailoring Process - Process of selecting method fragments and adapting the method
- Tailored Method - Presentation of the tailored process
- Experience Capture - Means of reflection and capturing experience for reuse.

Although similar components are identified, two main differences characterize the MT frameworks. Firstly, the terminology used to describe the components differs. Table 1 highlights the main components (or activities) and the different terminology used to describe each concept within the component. Secondly, each author elaborates and focuses on different aspects of the MT framework. For example, [13] places emphasis on the maintenance (creation, deletion and modification) of method fragments, while [13, 15, 18] concentrate their attention on modeling tools, and [17, 22] focus on the capturing of experience. Therefore previous research tends to highlight specific areas of MT, rather than treating the process holistically.

Most of these frameworks (such as such as [12, 13, 23]) have been developed theoretically based primarily on researchers' deductive observations (as summarized in the Table 1), with limited exposure of the frameworks to live software development projects [17, 23, 24]. As a consequence, these frameworks remain high-level with the limitation of not providing examples of how the individual components are implemented in industrial settings.

This paper builds upon previous research work by detailing further the individual MT components in terms of questions which can help practitioners reflect upon how tailoring has been carried out in individual projects as well as across projects at a wider organizational level.

## 3 Framework for Eliciting the Tailoring Process

Figure 1 depicts the Method Tailoring framework applied to the case study in Section 4. The subsections that follow will describe each component along with relevant questions that can be applied to guide the elicitation of the tailoring process used in software development projects. Table 2 provides a summary for this section.

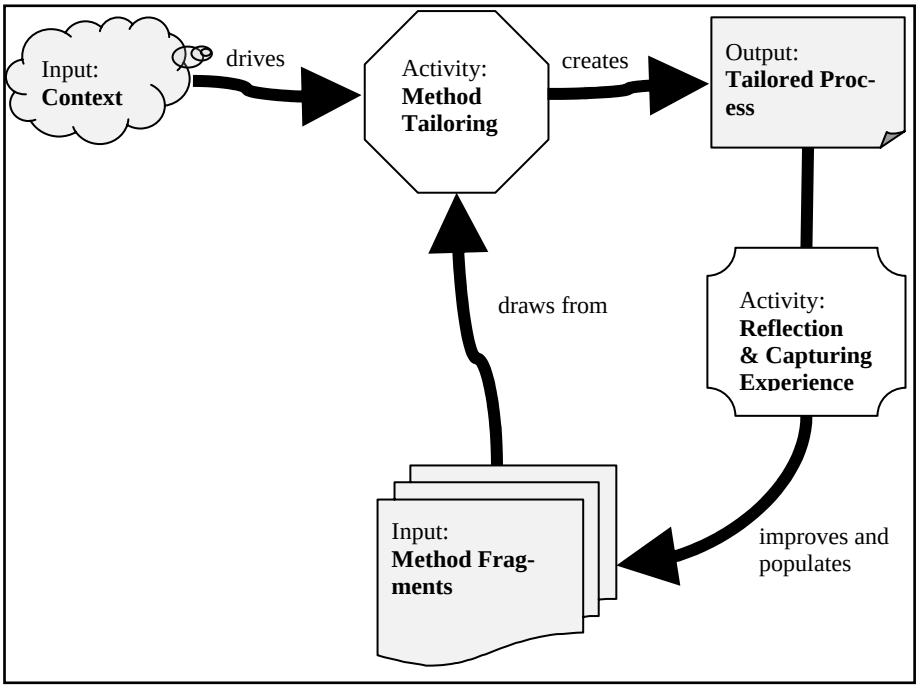


**Figure 1: Method Tailoring Framework.**

### Context

In the process of method tailoring, the starting point is a given, dynamic and evolving environment which is part of a larger organizational setting. This component is known as context. Fitzgerald et al. [7] consider context as a given that cannot be changed and is an input to the software development process. Fitzgerald et al. and Harmsen et al. [7, 13] consider the project environment to include both the supplier organization as well as the customer organization. A distinction is also made between the project environment and characterization of the project. They state that the project environment includes existing information infrastructure, the users, and the organizational culture while project contingency factors, such as application characteristics, external factors, technical factors, and the available development expertise are in

some way determined forming project characterization. Context in this framework includes characteristics that affect the selection and adaptation of method fragments. Glass [1] considers size, application domain, criticality and innovativeness factors that differentiate projects and captures context. Similarly, Cockburn [9] also considers size, criticality and project priority as differentiating project factors. Based on the comparative analysis of the literature, the context component in this framework has four categories of factors that influence software development. These broad categories are: Organizational characteristics, team dynamics/structure, project characteristics and product characteristics. Methods are tailored to a particular context; hence it is important to investigate how this context is captured and defined in practice.

**Method Fragments**

This component is often described by using the building blocks analogy. It is widely accepted that in order to tailor methods, methods need to be composed of method fragments. These method fragments provide the advantages of standardization as they are described with the same notation and terminology. The degree of flexibility of a SDM has great impact how easy or difficult tailoring is. One way to gain or achieve tailorability within SDM is to modularize SDM fragments for flexibility [4]. Keller [25] also discusses the modularization of fragments as a prerequisite to tailor methods. Method Fragments need to also have other characteristics that allow them to be assembled to form tailored methods. Modularized method fragments need to be loosely coupled to each other and highly cohesive [25, 26].

**Method Tailoring Process**

This component refers to the actual process of selection and assembly of method fragments to fit the particular context. In the elicitation of the tailored method it is important to identify how tailoring occurs in relation to the context (both project and organization). This implies understanding how method fragments are selected including what factors affect the addition, removal or modification of fragments. If in the organization method tailoring is formalized in any way, then key questions concern the roles and responsibilities of those involved in the tailoring process and the way decisions are documented and maintained for future memory.

**Tailored Method**

The tailored method is the result of MT. This includes all elements that would be typically found in a SDM. Given its application in a specific project, as well as future projects that share similar characteristics, it is essential to present, document and store the tailored method so that all interested parties can access and retrieve it. The format with which the tailored method is presented is essential to its future reusability and therefore to act as a template in other projects.

**Reflection and Capturing Experience**

All the frameworks analyzed from the literature were based on one same observation; People in practice tailor methods based on past experiences. Hence it is important to capture this experience for others to reuse. This enables an organization to learn from past successes and mistakes. This component deals with the issue of capturing method tailoring experience and presenting it in a reusable way so others can use the information. For example, capturing rationale for selecting certain fragments is also important, as this helps gather knowledge so team members can use and learn from others experiences.

| MT Components | Topics/Scope | Related Questions |
|---|---|---|
| Context (contextualization of the project environment) | Scope of context, capturing context. | Organizational: size, types of applications developed, team dynamics/structure, etc.<br>Project: size and complexity of project, type of application, etc.<br>What contextual factors have significantly affected the tailoring process? |
| Method fragments and maintenance | Addition, deletion and modification of method fragments.<br>Modularity of fragments.<br>Case tool support: Repository or method base for storing method fragments, retrieving and assembling method fragments. | What methodology does your organization use? Is it prescriptive? Does it have modularized method fragments that you can select? Are they loosely coupled and highly cohesive? Do they use the same notation and terminology? |
| Method Tailoring Process | Incremental ME, selection and adaptation of method fragments to project characteristics, case tool support. | Is MT done? Is MT seen as an important activity or a negative activity as in not following the process rigorously? How do you tailor your process based on the context you have and the Method fragments? How are these method fragments selected in practice and how are they assembled? Who does the tailoring and when is it done? Have you formalized the method tailoring process? Does your organization have tailoring guidelines that you can follow? |
| Tailored Method | Ability to reuse the tailored process, case tool support. | How is it applied and updated? How are templates produced, used and adapted? |
| Capturing experience | Reflection, capturing method rationale, review process, reuse, case tool support. | What were the lesson learned in terms of benefits and limitations derived from the tailoring experience of the project? How is experience captured and preserved? |

**Table 2: Components of MT framework related to the questions.**

## 4 Case Study and Application of Framework

The framework described in the previous section was applied to a project carried out at S-Service, a medium-sized software development company located in southern Italy and founded 30 years ago. S-Service is specialized in the development of information systems for the healthcare sector and local public administrations (PA). Over the years it has extended its presence throughout the national territory becoming one of the leading software/information systems firm for the healthcare and PA sectors. The philosophy underlying the overall organization, as well as its approach to providing integrated business solutions to its customers, is grounded in the quality of its processes and products. S-Service's Quality System was externally certified in 1996 (in accordance with UNI ISO EN 9001) and received a special mention from Premio Qualità Italia in 2000.

This case study documents a large development project carried out for the Italian National Health Service (INHS). The project is aimed at the realization of a series of 'software services' strategically intended to improve the quality of service of the INHS allowing medics, healthcare staff and citizens to directly interact with local and regional health structures through the Internet. The services were called Network Application Services (NAS). Ten NAS were defined and allocated to distinct subprojects. The case study therefore refers to a coordinated project divided into ten subprojects. The NAS were from a technological perspective based on web services.

The SDM adopted for this project was the Rational Unified Process (RUP). RUP was tailored to suit specific contextual organizational and project characteristics. The tailoring process was elicited a posteriori by applying the framework proposed in this paper. The following subsections are structured in line with the framework components and describe S-Service's experience in tailoring this specific project. Due to limitations of space, a representative extract of the case study documentation is presented. As the case study will demonstrate the framework contributes in the following areas: (1) provide the project team and the organization with a structured means of eliciting the tailoring process; (2) allow the development team to document the experience of the project for the benefit of future development and maintenance efforts and (3) facilitate reflection and constructive self-criticism.

### 4.1 Context

RUP was originally adopted by S-Service in 2002. The decision to adopt RUP was strongly influenced by the emergence of object-oriented methodologies on the market. Given RUP's characteristic of being a methodological framework rather than a rigid and binding methodology, decisions are taken at the beginning of each project as to the process, roles and artifacts that will be applied and managed. With the experience previously acquired, the project team adopted a tailoring process that had evolved from previous projects.

Contextual factors that most affected method tailoring for this project were as follows:

- Presence of staff/team members highly competent in the domain area with detailed knowledge of the business processes and rules of the public

healthcare sector. This led to the decision of not adopting RUP's Business Modeling workflow given that the business domain had been previously documented and staff was competent in the area.

- Domain and type of client: In the Italian public healthcare sector, as with all public bodies, contracts are acquired through a request for proposals and a public bidding process; hence very formal procedures which require precise and detailed documentation of the system to be delivered. This affected the selection of artifacts and their level of formality.
- Requirements of Enterprise Application Integration (EAI) with existing legacy systems. This factor influenced the whole of lifecycle which needed to take legacy and integration requirements into consideration, affecting requirements documentation as well as analysis, design and implementation artifacts.

From past tailoring experiences at S-Service, tailoring at the organizational level involves adjusting the method to (1) the types of applications normally developed (e.g., business applications), (2) the organizational models or (3) the technological architectures used. Considering that the objective is to obtain a method that is applicable to most of the company's projects, tailoring mainly tends to eliminate those components that are likely not to be ever applied (e.g., components specialized for the development of real-time software).

### 4.2 Method Tailoring Process

The tailoring process is driven by the Quality Assurance (QA) team in collaboration with the development team. A series of meetings take place in which tailoring decisions are made. These decisions are then formally documented in the Quality Plan. The Quality Plan defines the overall method adopted by the project. In general, the tailoring process initiates at the beginning of a project. Previous experiences suggest that the tailored method can either be totally decided at the start of the project or the method can be refined in various iterations as development progresses. In any case all workflow details and activities must be defined before the corresponding workflow takes place. At S-Service RUP is tailored in order to provide the optimal development process (and workflows) defined by the best cost/benefit ratio.

Tailoring occurs most frequently at the project level. Project level tailoring is mainly aimed at adapting the method to specific project characteristics. Often a method tailored for a project is used as the basis for developing further projects with similar characteristics, such as case tools, the target technological architecture and the development environment. In this situation a method tailored at a project level has the potential of being used at a departmental level and therefore covering a class of common projects.

Method tailoring is a part of the overall development planning. As far as the tailoring is concerned planning involves:

a) The eventual elimination of some of the original method activities/tasks
b) Addition of new activities/tasks
c) Adjustment (tweaking/redefinition) of activities/tasks to suit the project characteristics

d) Definition of checks (e.g., reviews/inspections) on the deliverables aimed at assuring correctness, consistency and traceability of requirements all the way down to implementation.

The above tailoring activites must be motivated and are documented in the Quality Plan with refernce to other appropriate documents (e.g., naming standards document, planning document, etc.)

When carrying out MT a reference method is normally adopted. The reference method usually is the one that best integrates with the case tool chosen to support the lifecycle.

### 4.3 Tailored Process/Method

As previously mentioned, method/process tailoring is mainly documented in the Quality Plan and other documents. The Configuration Management Plan is also important to this end since it documents, among other things, how software and documentation should be configured, versioned and so on.

For this project the Quality Plan documented the following elements:

- Selected RUP workflows: Support and Management workflows (e.g., configuration management, project management, testing, etc.) were not selected since the corresponding processes were already catered for in the company's standard procedures.
- A table was prepared in which the following information was reported for each workflow:
  - Workflow: Name of RUP workflow
  - Activity: Name of workflow detail activity
  - Objective: Goal of the activity or workflow as it is defined by the method
  - Inclusion: Specifies whether an activity has been included in the tailored process. If an activity is not included reasons are given in the Notes column.
  - Output: Outputs of activities and workflows are defined (i.e., artifacts)
  - Notes: Annotations, suggestions, reasons are annotated
- Checks to carry out on design and implementation artefacts aimed at verification (consistency between inputs and outputs of each phase, traceability checks, respect of standards, etc.) and validation (software testing with appropriate procedures). The checks are tailored depending on the deliverables produced and the tools used (e.g., Rational Rose).

The RUP deliverable templates were examined and adapted to the needs of the project, therefore producing tailored templates that were adopted by each subproject team.

Scheduling of the tailored project activities was documented in the Project Plan defining:

- Roles and skills required to carry out the activities
- Resources required by the project (human, tools, infrastructure)
- Scheduling (e.g., Gantt chart)

- Monitoring times and metrics
- Predicted and actual project costs
- Evaluation (both technical and managerial) of project activities aimed at highlighting critical aspects.

The plans are living documents that are updated during projects in order to correct inadequate situations or to prevent critical situations caused by erroneous tailoring decisions.

### 4.4 Reflection and Capturing Experience

According to the organization's standard procedures, the Project Manager is required to write a 'Project Closure Report' documenting the overall project experience. The report is submitted to Senior Management and it contains an overall analysis of the project including decisions taken, reasons for deviating from predefined standards and objectives. An evaluation of tailoring decisions made is included in the report. The report is accompanied by a set of metrics that serve as indicators of project success.

At present these metrics are recorded on paper documents. The next objective will be to populate a repository containing information on all projects. This repository would support decisional processes (which methods to adopt, tailoring, etc.) and evaluations (benchmarking) for the development and maintenance of future projects.

Various benefits of MT have been identified and documented for this project. These benefits can be summarized as follows:

- Consistency of the development process within the same project
- Consistency of the techniques used by the team members (same training, reduction in communication overheads, standard communication protocols)
- Consistency of artifacts, models and standards
- Possibility to reuse the same process in similar projects (a sort of process polymorphism) by simply tweaking where necessary
- Benchmarking and measurements applied in one project can be used as the basis for predictions in other projects.

### 4.5 Method Fragments and Maintenance

Reuse at a project level is applied when a new project is started and the decision is taken to adopt a method previously used in the company. The project would benefit from past experience and previously tailored and documented methods. The latter (along with the Project Manager's final evaluation) represent a point of reference for the project's processes and deliverables as well as support the initial definition of the project's activities.

Generally speaking, for every tailored development project the following elements are "instantiated": (1) a configuration library containing software artifacts and documentation and (2) a "Development Standards" document defining for the specific project standards and techniques that analysts, designers and programmers must adopt. The "Development Standards" document generally includes a description of

the repository's logical structure. The repository contains workflow artifacts and a description of a "generalized method" that that can be specialized if the case. For example, for a project using an object-oriented approach (e.g., RUP with Rose) the repository would describe, for example: packages and artifacts they contain, naming conventions of packages and artifacts, package contents and relationships between packages.

The Development Standards document also defines development patterns that must be adopted for the realization of different architectural components. For example, the MVC pattern used for the web tier of applications and a formal description of how such pattern should be applied.

There normally is one Development Standard document for each production line and referenced in each project. A production line is characterized by the same methodology, the same development technology and the same software architecture.

## 5 Conclusion

This paper has presented a case study of a large-scale software development project in which the Rational Unified Process was tailored to suit the specific contextual characteristics of the organization, project team, client and application type. A Method Tailoring framework derived from the literature was adopted to conduct the research.

The main findings of the case study were as follows. The organization had previous tailoring experiences; as a consequence the MT process was formalized in some aspects (e.g., defined tailoring roles and responsibilities along with formal documents in which decisions were annotated) and not fully mature in other aspects (e.g., repository management and benchmarking required more growth). From a procedural perspective tailoring was carried out iteratively (for most part) and hierarchically conducted. Three levels were identified organizational, departmental and project. In the case of the project described in Section 4, the project itself was divided into ten subprojects. Even in this case successive refinements were applied. The initial subprojects served as pilots for the following ones.

At a project level methods at S-Service are selected from a repository of previously tailored RUP methods. The selection is based primarily on application type, system architecture adopted and case tools. The repository contains generalized methods that can be specialized (through a form of process 'polymorphism') to similar projects.

In terms of the MT framework presented in this paper, its validity is twofold. Firstly, it is grounded in previous research work documented in the literature. Secondly, it has been applied to document a large-scale industrial case study. Although the work presented in this paper provides a contribution to the area of Method Tailoring, it does nonetheless have its limitations.

Firstly, at this stage of the research, investigations have been conducted only on one case study. It is desirable to explore further development organizations in order to identify commonalities and differences with the organization and project described in this paper. Work is currently being carried out on additional case studies of MT in two large development organizations. The first operating at a national level in the United Kingdom, while the second operating as a multinational worldwide. These

further case studies will highlight, among other things, how the dimensions (medium, large and very large) affect MT decisions and processes.

Secondly, the framework can be further refined in light of the findings related to the S-Service case study. These relate to expanding each framework component (e.g., detailed questions and identification of further relationships between the components). This expansion would positively affect the work being carried out on the aforementioned case studies.

## References


1. Glass, R., *Through a Glass, Darkly. Methodologies: Bend to Fit?* The Software Practitioner, Data Base Advances, 1996. **27**(1): p. 14 - 16.
2. Hardy, C.J., J.B. Thompson, and H.M. Edwards, *The use, limitations and customization of structured systems developement methods in the United Kingdom.* Information and Software Technology, 1995. **37**(3): p. 467-477.
3. Brooks, F.P., *No Silver Bullet - Essence and Accidents of Software Engineering.* Computer, 1987. **20**(4): p. 10-19.
4. Cameron, J., *Configurable development processes.* Communications of the ACM, 2002. **45**(3): p. 72 - 77.
5. Fitzgerald, B., *Formalized systems development methodologies: A critical perspective.* Information Systems Journal, 1996. **6**(1): p. 3-23.
6. Avison, D.E. and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*. Third ed. 2003: McGraw HIll.
7. Fitzgerald, B., N.L. Russo, and E. Stolterman, *Information Systems Development: Methods in Action.* 2003: McGraw Hill.
8. Iivari, J., R. hirschheim, and H.K. Klein, *A Dynamic Framwork for Classifying Information Systems Development Methodologies and Approaches.* Journal of Management Information Systems, 2001. **17**(3): p. 179 - 218.
9. Cockburn, A., *Agile Software Development*. The Agile Software Development Series, ed. A. Cockburn and J. Highsmith. 2002: Addison Wesley.
10. Firesmith, D.G. and B. Henderson-Sellers, *The OPEN Process Framework*. 2002, London: Addison Wesley.
11. Hidding, G.J., *Reinventing Methodology: Who Reads It and Why?* Communications of the Acm, 1997. **40**(11): p. 102 - 109.
12. Kumar, K. and R.J. Welke, *Methodology engineering: a proposal for situation-specific methodology construction*, in *Challenges and Strategies for Reseach in Systems Development*, W.W. Cotterman and S.J. A., Editors. 1992, John Wiley.
13. Harmsen, F., S. Brinkkemper, and H. Oei, *Situational Method Engineering for Information-System Project Approaches*, in *Methods and Associated Tools for the Information Systems Life Cycle*. 1994. p. 169-194.
14. Fitzgerald, B., N.L. Russo, and T. O'Kane, *Software development method tailoring at motorola.* Communications of the Acm, 2003. **46**(4): p. 64-70.
15. Tolvanen, J.-P., *Incremental Method Engineering with Modeling Tools*, in *Department of Computer Science and Information Systems*. 1998, University of Jyvaskyla: Finland. p. 301.
16. Ter Hofstede, A.H.M. and T.F. Verhoef, *On the feasibility of situational method engineering.* Information Systems, 1997. **22**(6-7): p. 401-422.
17. Henninger, S., *Turning Development Standards Into Repositories of Experiences.* Software Process Improvement and Practice, 2001. **6**(3): p. 141-155.

18. Henninger, S., et al. *Supporting Adaptable Methodologies to Meet Evolving Project Needs*. in *XP/Agile Universe*. 2002.
19. Guimaraes, T., *A study of application program development techniques.* Communications of the ACM, 1985. **28**(5): p. 494 - 499.
20. Henderson-Sellers, B., *Method engineering for OO systems development.* Communications of the Acm, 2003. **46**(10): p. 73-78.
21. Lycett, M., et al., *Migrating Agile Methods to Standardized Development Practice.* Computer, 2003. **36**(6): p. 79 - 85.
22. Rossi, M., et al. *Method Rationale in Method Engineering*. in *Proceedings of the 33rd Hawaii International Conference on Systems Science*. 2000. Hawaii.
23. Baskerville, R. and J. Stage. *Accommodating Emergent Work Pratices: Ethnographic Choice of Method Fragments*. in *IFIP TC8/WG8.2 Working Conference on Realigning Research and Practice in Information Systems Development*. 2001. Boise Idaho: IFIP Kluwer Academic Publishers.
24. Kellner, M.I., L. Briand, and J.W. Over. *A method for designing, defining, and evolving software processes.* in *Proceedings., Fourth International Conference on the Software Process.* 1996.
25. Kellner, M.I. *Connecting reusable software process elements and components*. in *Proceedings of the 10th International Process Support of Software Product Lines.* 1996.
26. Yoon, I.-C., S.-Y. Min, and D.-H. Bac. *Tailoring and verifying software process*. in *Eighth Asia-Pacific Software Engineering Conference (APSEC'01).* 2001. Macao, China.