Continuous tailoring activities in software engineering

Vincent Ribaud, Philippe Saliou

Continuous tailoring activities in software engineering

P. Saliou and V.Ribaud
Thales Information Systems, Brest & Département d'Informatique, Faculté des sciences, Brest
E-mail : Philippe.Saliou@univ-brest.fr, Vincent.Ribaud@mythales-is.com

Abstract

Software activities belong to different processes. Tailoring software processes aims to relate the operational software processes of an organization to the effective project. With the information technology industry moving ever faster, established positions are undergoing constant evolutionary change. The failure of a complex tailoring process of a management information system is reported. There is a need to adopt software processes that can operate under constant change. We propose to add a new dimension called continuous tailoring which is applied at any moment, to each software activity and for each actor. This process is validated in an immersion system intended for young post-graduates before they begin their careers in the software industry. Relationships between tailoring and improvement are considered.

Keywords

Tailoring, software process improvement, quality management, information systems.

1 Introduction

Software engineering can be minimally defined as the set of activities involved in developing, operating and maintaining software. Software engineering standardization is an attempt to integrate, regulate and optimize existing best practices and theories [Wang02]. Software companies use software engineering and software quality standards as the foundation of their quality assurance process or of their quality management system. On the other hand, innovation and change occur faster in the software industry than in any other industry. "Faster, better, cheaper" : people are expected to deliver products faster, of better quality, and at lower cost.

The experience presented in this paper attempts to reduce the gap between operating under the control of stable, well-defined and controlled processes and surviving in a rapidly evolving information technology industry. Faced with this challenge, each project should answer with a kind of continuous adaptation. Hence, the larger question of software process improvement is also addressed.

Quality management baseline is the set of process requirements, standards, corporate commitments and best practices controlling the software processes. The quality management baseline needs to be tailored to a project baseline. The tailoring usually begins at the tendering phase and is refined at the start of the project. Section 2 describes a project where strong tailoring was envisaged and defined at the beginning, but where it did not work as expected.

In section 3, we propose to consider tailoring software activities as a continuous task, managed by the project manager and supported by the whole software team. This adds a new dimension to the software processes we have called harrow processes, using the metaphor of a tine harrow from ecological and sustained agriculture. Tine harrows are used to control weeds; tines can be raised or lowered individually to cultivate specific areas.

Section 4 presents the framework and the projects where we are experimenting and validating the

continuous tailoring. Students in the French technological education system graduate after 4 years of studies which include at least three work placements. Thanks to a strong partnership with Thales Information Systems, students obtain a Master diploma within a software engineering apprenticeship by immersion system [RS03]. Young graduates are immersed in a team led by an experienced software manager and have to deliver real software products under the control of Tempo, the quality management system of Thales-IS. Continuous tailoring is applied at each phase of the project and some observations are presented in this section.

Continuous performance improvement is one of the main issues of ISO 9000:2000 as well as ISO/IEC 15504 (SPICE). In section 5, we try to place continuous tailoring in the light of continuous improvement.

2 Tailoring in practice

The authors both worked for nearly ten years at Thales Information System (formerly Syseca Inc), a software services company. We led projects and developed several management information systems under the control of TEMPO, Thales Information System corporate baseline. « TEMPO is the foundation of the company corporate culture and is the basis for sustaining good working practices. It is a set of procedures, guides and instructions defining how the company operates, and how it is organised, providing a framework for project management, software development and system integration activities [TIS02]». The quality system model adopted is the ISO 9001 standard. From a software development point of view, TEMPO defines a general lifecycle (requirements analysis, preliminary design, detailed design, coding and unit testing, integration and integration testing, software testing).

TEMPO needs to be tailored to each project. The methods and documents provided have a broad range. Tailoring is a vital component of the response to solicitation phase and early project phases. This tailoring process defines the activities to be performed and products to be developed and delivered. With the SPICE terminology, TEMPO is the company's set of standard processes, and tailoring TEMPO to a project results in its instanciated process. TEMPO is somewhat too general, and depending on the domain/size/methods/tools/... of the project, there is a lack of more precise and dedicated standard processes.

Some lessons learned

Ariane was the last project led by the primary author before leaving the Thales Information Systems company. Ariane processes were peer-reviewed by the second author. Ariane (which is operational and still growing) is a Management Information System for procurement by the textile distribution branch of several regional purchasing centres (each supplying around 40 stores). The solution implemented is based on an n-tier architecture, with a central database, local servers in the stores and tablet pen-based computers (without keyboards) radio-linked to the local servers [Sal98].

Oracle Designer was chosen in the perspective of a highly-generated implementation from models. Models are produced according to CADM, the companion method of Oracle CASE tools suite. Moreover, the information system should be commissioned with several stages through successive work packages. In terms of technique, development tools and organization, highly innovative and strong choices were made. As well-disciplined project managers, we started the project with the tailoring of the TEMPO baseline. We spent about a month writing the Project Plan, especially to define the managerial and the technical processes. The main difficulty was to tailor the software development process with a partial knowledge of the Designer tool and its associated CADM method, but without any real practice. The project team received several weeks of technical training. So, the project started with an optimistic software development process, in accordance with TEMPO requirements and intended to address two different issues :

- the definition of milestones and reviews in order to ensure contractual commitments (cost, lead-times and performance) and deliveries of work packages;
- the use of a model-driven method built onto an uppercase tool with each project engineer working on a different functional area.

The optimistic software development process is depicted in figure 1. Unfortunately, the envisaged

tailoring did not work. We returned from our incremental and iterative process to a traditional "V" model of lifecycle. We came back to a traditional client-server (and centralized) architecture. At the same time, we abandoned the generation of code modules (we kept the highly-generated process for the data) to go back to a traditional analysis and design phase followed by a traditional implementation phase using Oracle lowercase tools suite. The main reasons for this reversal were :

- we did not pay enough attention to technical architecture;
- we underestimated the constraints and the complexity of Oracle underlying framework;
- tailoring at the beginning requires real practice and at least one other similar experience;
- the model-driven process is more powerful but more difficult to master than a programming approach;
- the proposed process relies on the idea that it would be possible to change mindsets in one big step instead of building on a progressive and continuous sequence of steps in order to achieve progress.

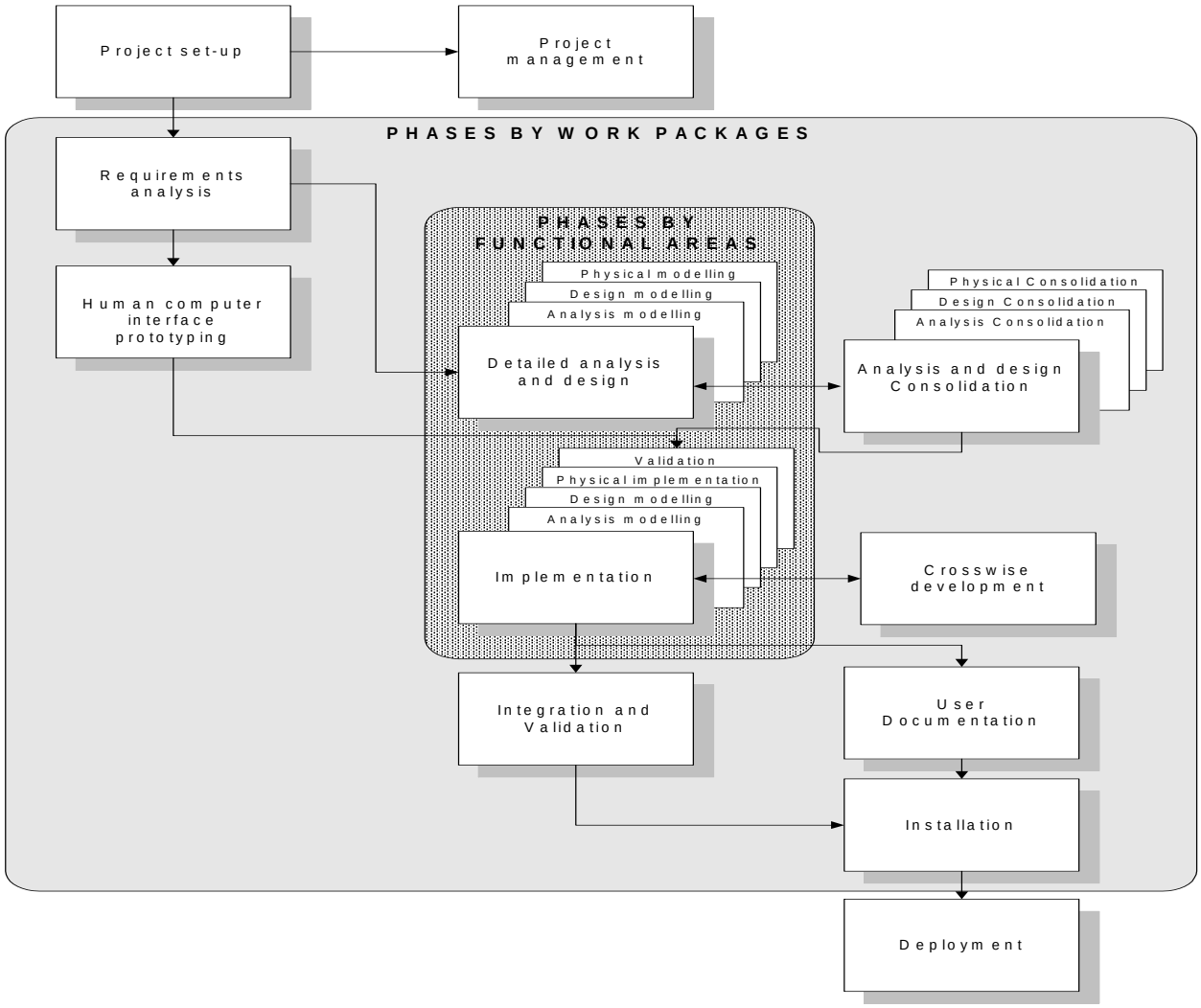

Figure 1 : Ariane envisaged development process

3 Continuous tailoring

In 2000, the primary author left his industrial post to join a professional education institute. The second author also works in this institute, and is a part-time consultant at Thales. We have kept a strong rela-

tionship with Thales Information Systems by providing it with consulting services on projects as well as methodological research on TEMPO baseline. We wanted to solve the identified problems within Ariane, following changes in technology (UML, n-tier architecture, complex frameworks and associated tools). It took us 6 months to build dedicated standard processes for current management information systems, including "on demand" tailoring activities. Then after experimenting with these processes during one year, we chose to structure these activities in a separate dimension called continuous tailoring or "harrow process". One year on, we now believe that this new dimension (adding adjustable tines to suppress weeds over the desirable area) is essential to achieve modern management information systems requirements.

Standard processes for current management information systems

The starting point is TEMPO, Thales Information Systems baseline. In fact, we have built a specialization of TEMPO with the following constraints : a medium-sized team (from 3 to 8 people), a "Management Information System (MIS) orientation", an n-tier architecture, an ISO 9001-like quality management system, a software development process belonging to the Unified Process family, a model-driven approach, uppercase tools suite and complex frameworks.

Our software development process relies on the truly specific features of the unified process that is use case driven, architecture-centered, iterative and incremental [JBR99]. Nevertheless, UML is used when it makes sense to do so. For example, analysis is carried out through a hierarchical functional decomposition rather than an object-oriented analysis.

We kept the main idea of the 2TUP (2 Track Unified Process) that is, to separate functionalities and technical architecture. « 2 Track literally means that the process follows two paths... At the close of evolutions concerning both functional model and technical architecture, system implementation consists in merging results issued from the two paths. This merging leads to a Y-development process [RV02]».

Continuous tailoring : a new dimension in project management

Innovation and change occur ever faster in the software industry. In a rapidly changing environment, pressure for change acts on at least four complementary lines :

- the team;
- software development process, methods and tool suites based on a model-driven approach;
- complex technical frameworks and n-tiered architectures;
- last but not least, the market place requires the delivery of products faster, at lower cost and with better quality.

Dealing with these orthogonal constraints generally falls to the project manager. This requires time and continuous effort. Unfortunately, these problems are rarely considered together, or if they are, only at the beginning of the project when most of the problems are unknown. Hence, the idea is to perform a "just-in-time" tailoring when the need occurs but as a planned activity. This adds a new dimension which plays a crucial role in dealing with technological and methodological evolutions as well as enhancing team performances.

The project manager will have to : 1) identify activities for which a previous tailoring is needed; 2) define as well as possible these activities dealing with cost and delay, anticipation (sow at the right time in order to harvest later), evaluate team skills and performances; 3) control and validate the activities results either alone or with help from peers.

4 Experimenting continuous tailoring

Experimentation framework

Students in the French technological education system graduate in 4 years, having completed at least three training periods in a firm. Thanks to a strong partnership with Thales Information Systems, we

had the opportunity to immerse young graduates in an imitation of the real world. The plan of action is built on a 6-month project. Young graduates make up two teams of 5; each team is led by one of us acting as project manager. A contract defines the customer-supplier relationship. The TEMPO specialization depicted in section 3 is the project baseline, including the continous tailoring dimension. Each project team has its own office with individual working post and common installations. Each project team uses a different and complete software engineering tools suite (Oracle and Rational/IBM Websphere). At the end of the academic year, the team members obtain a Master's degree in Software Engineering.

The first year, we considered tailoring practices (activities) as part of the related software processes along with the other practices (e.g. "*Ajust the design*" with "Preliminary design"). The second year, we placed these practices in the harrow-dimension.

Definition of continuous tailoring

Tailoring practices (activities) as other practices are roughly-defined, estimated and scheduled in the project plan. Continuous tailoring applies to four process categories (in the SPICE sense) : Customer-Supplier (CUS), Engineering (ENG), Project (PRO) and Support (SUP). Each process category is a set of processes addressing the same general area of activity. Each process is a set of practices that address the same purpose [Spi95, part 9]. Tailoring practices refer to a process; they are intended to improve the process but mainly to contribute to the process realization, i.e. to achieve the process goals and to deliver required outputs.

Let us present a list of the tailoring practices, defined by the first author for his project, applied to the engineering process. The table 1 illustrates the new dimension added to ENG process category. The first column relates to the ENG main processes. The second column relates to the base practices of each process. The third column relates to the tailoring practices to be performed for each process. The inclination and variable size of the third column symbolizes the tines of the harrow : tines need to be adjusted in order to kill weeds over the desired areas and to the required depth.

Software development engineering		
Requirement capture	Functional requirements capture Technical requirements capture Requirements consolidation	Requirement capture definition and set-up
Technical architecture	Generic design Technical prototype	Technical architecture validation Development framework exploration
Analysis	Requirement analysis Human-computer interface mock-up	Analysis definition and set-up
Design	General design Detailed design Relational database consolidation	Design adjustement Database modelling and implementation guide
Coding - Unit testing	Coding - Unit testing	
Integration-Qualification	Software test plan elaboration Software test description elaboration Integration Internal validation Qualification	Corporate test procedures learning

Table 1 : Harrow-dimension of the Engineering process category

Performing continuous tailoring

The tailoring practices, as well as the base practices, are described in the project plan at an abstract level. Some well-known practices are detailed while others remain general. Before the related process starts, the software manager should write a work card describing in detail how to carry out the required tailoring.

Let us illustrate continuous tailoring practice on the design process. “The purpose of the Develop software design process is to establish a software design that effectively accommodates the software requirements; at the top-level this identifies the major software components and refines these into lower level software units which can be coded, compiled, and tested [Spi95, part 2]”. The baseline offers a set of base practices intended to accomplish this goal. These base practices are described at an abstract level identifying “what” should be done without specifying “how”. The main difficulty for a project is to be provided with a response to “how ?” suitable for the project specificities : technique, methodology, ...

When the development environments are stable and mature, the way to use it can be found “off-the-shelf” or at least progressively built from successive projects and capitalized in the corporate baseline. When methods and technologies are continuously evolving, the “software component” definition differs according to frameworks, models and tools. Then we need to adapt constantly the “how to design” issue to innovations and changes. That is precisely what the tailoring activities should provide answers to. For example, adjusting the design is aimed at understanding, preparing and defining the design model that should be used during the design phase. Thus, the instructions given in the “Design adjustment” card are intended to answer to these questions :

- which role does the design play in our software process ?
- what are the determining elements and relevant models useful in elaborating the design models ? why are they used ? how are they elaborated ?
- how do project design constraints (modelling language e.g. UML, process e.g. 2TUP, technical framework e.g. J2EE, ...) correlate with corporate baseline requirements ?

The approach envisaged to answer to these questions is to carry out the retro-design of the technical requirement prototype (a previous tailoring activity) used to validate the technical architecture. The output of this activity is a design document which follows TEMPO guidelines.

Benefits of continuous tailoring

Inexperienced graduates greatly benefit from these tailoring activities : very often, they found it difficult to start the related work and carrying out the tailoring product gave them the required boost.

The project manager has difficulties in conciliating production constraints (mainly cost and lead-time) with the need to survive in the market place. This requires constant adaptation to technological and methodogical innovations and team competence enhancement.

The continous tailoring dimension is a valuable tool of the project manager. It helps to materialize cost and lead-time of tailoring activities; to involve everyone in the team to the improvement process; to promote team members’ creativity and self-training; to show improvement initiatives and results to the top-level managers.

5 Continuous tailoring and/or continuous improvement

ISO 9000 advocates the continuous improvement principle illustrated by the Deming wheel (Plan-Do-Check-Act) [Mit04]. ISO/IEC TR 15504 (SPICE) develops a 2-D process capability assessment model with both a process and a capability dimension including a level 5 : continuously-improving level. Both approaches rely on a quantitative understanding of process capability and management.

Continuous process improvement and continuous tailoring are two different but complementary approaches with the same goal. The main difference is that tailoring is undertaken while producing the software in order to deal with unanticipated project constraints while in process improvement, change is based on a quantitative understanding of the effectiveness of process changes.

The ISO 9000 continuous improvement principle supposes that there are no stable states for a quality management system while the harrow-dimension supposes that there are no stable states for an MIS development process. There are always ways of improvement intended to emphasize productivity as well as enhancing the quality of the products. Thus, there is a strong need to adjust the process while performing it.

In the software process improvement approach, discipline and creativity are often in opposition [CF02]. It is important to balance discipline and creativity, that is one of the goals of the continuous tailoring dimension : adopting software processes that can operate under constant change.

Continuous process improvement requires commitment from all the staff. Unfortunately, the effective processes allow many actors (developers, system administrators, ...) to stay outside of the management quality system. In our proposal, continuous tailoring can be triggered at each moment on each activity that leads to a natural commitment of the concerned actors.

6 Conclusion

Unfortunately, tailoring usually begins at the tendering phase and is refined at the start of the project. Moreover tailoring is usually carried out by the project manager and the software team is not involved. We argue that tailoring software activities is a continuous task, managed by the software manager and supported by the whole team. We believe that it is a good way to manage the complexity of change and innovation and to motivate a team in order to improve their processes and performance.

7 Literature

[CF02] Reidar Conradi, Alfonso Fugetta, Improving Software process Improvement, IEEE Software – july/August 2002 , IEEE, 2002

[JBR99] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison-Wesley Longman, 1999

[Mit04] Henri Mitonneau, ISO 9000 Version 2000, Dunod, 2004

[RV02] Pascal Roques, Franck Vallée, UML en action, Eyrolles, 2002

[RS03] Vincent Ribaud, Philippe Saliou, Software Engineering Apprenticeship by Immersion, International Workshop on Patterns in Teaching Software Development, ECOOP'03, 2003

[Sal98] Philippe Saliou, Ariane – gestion de l'approvisionnement textile du groupe Leclerc, Le Bulletin Technique – Vol. 6, Num. 3 , Thales Information System, 1998

[Spi95] Software Process Improvement and Capability dEtermination (SPICE), Software Process Assessment - Version 1.00, http://www.sqi.gu.edu.au/spice/docs/baseline, 1995

[TIS02] TEMPO, information systems development under control, Thales Information System, 2002

[Wang02] Yingxu Wang, Software engineering standards : review and perspectives, Handbook of Software Engineering and Knowledge Engineering, World Scientific, 2002

8 Author CVs

Philippe Saliou

Philippe Saliou, 40 years old, B.S., M.S. and Ph.D. in Computer Science, has been an Associate Professor in the Computer Department of Brest University, France, since 2000.

Before entering the academic world, he spent thirty years in the software industry, essentially as an information system project manager. His last industry job was in UNICOPA, one of the leading agri-food co-operatives in France (4200 employees, 30 industrial sites, 18000 farmers), where he was the Computer Project Director. He spent more than eight years in Thales Group, a world leader in electronic systems serving aerospace, defense and information technology, where he led important projects such as the French Hydrographic and Oceanographic Institute information system, the display system of the Charles de Gaulle aircraft carrier, the textile procurement information system for the supermarket chain Leclerc (15 regional purchasing centers, 600 stores), etc.

He teaches software engineering, information systems through model-driven approach, software configuration management, requirements capture, software analysis and design and other software engineering activities.

His research interests include information system and software engineering. He is currently working on ways to improve processes by adopting software processes that can operate under constant innovation and change in a rapidly evolving technology industry. He is also interested in developing new pedagogical approaches for improving teaching and learning in software engineering education. Over the last two years (2002-2004) he designed and set up a new Master Degree in software engineering relying on an apprenticeship by immersion paradigm, in cooperation with Vincent Ribaud.

Vincent Ribaud

Vincent Ribaud, 45 years old, B.S in Mathematics, M.S and Ph. D. in Computer Science has been an Associate Professor in the Computer Department of Brest University, France, since 1994, and partly senior consultant at Thales Information System, a software services company .

He began as a lecturer at Rennes and Brest Universities before spending six years in the software industry, essentially as an MIS project manager and technical architect. Since 1996, he has led the software engineering research group at Brest University.

His research interests include software engineering, information system and generative design and programming. He is currently working on software process improvement and on semantic integration of software components.

He teaches software engineering, distributed systems and networking. Eighteen years of software engineering teaching has led him to co-invent (with Philippe Saliou) a new pedagogical paradigm called apprenticeship by immersion.