Agile Process Tailoring and probLem analYsis (APTLY)

Frank Keenan
University of Ulster at Coleraine
frank.keenan@dkit.ie

1. Research Area

Software process, process tailoring, problem analysis.

2. Research Problem

Developing software using a well-defined, well-understood process improves the likelihood of delivering a product with the required quality. Enhancing that process to meet recognised process standards, such as CMMI and ISO 9000, can further facilitate the development of complex systems in a repeatable and predictable way. There are tradeoffs involved, however. In particular, because projects differ in their scale, scope, and technical challenge, the same process will not suit all circumstances. *Agile* approaches to development, such as Extreme Programming (XP), SCRUM and Crystal Methodologies, recognise this dilemma and suggest that processes be tailored to each situation. The research problem for postgraduate investigation is to determine in detail how this can be achieved successfully. This will include a consideration of how best to define, maintain and give access to a knowledge base recording details of process concepts, techniques and experience.

3. Approach

There are three main strategies for process tailoring:

(a) Provide a comprehensive process framework, such as RUP, that attempts to cover all possibilities, and from which the most appropriate elements are selected at the beginning of each project.

(b) Define a set of process templates corresponding to different types of project, as with the Crystal family of methodologies [1], and select the best match for the situation presented, possibly followed by some fine-tuning of the selected process.

(c) Define a process by blending ideas and techniques from best practice and local experience.

Strategies (a) and (b) imply *static tailoring* at the beginning of a project. However, during development some adjustment may be necessary to attain optimal performance [2]. The "Living Software Development Process" [3] provides a pattern-based framework for such dynamic tailoring. Lycett describes a similar approach [4] in which each situation is assessed on four criteria: *project*, *product*, *team* and *organization*. Patterns are selected from RUP and each process evolves as a learning network.

Figure 1 is one way of describing a general framework for process tailoring based on approach (c).

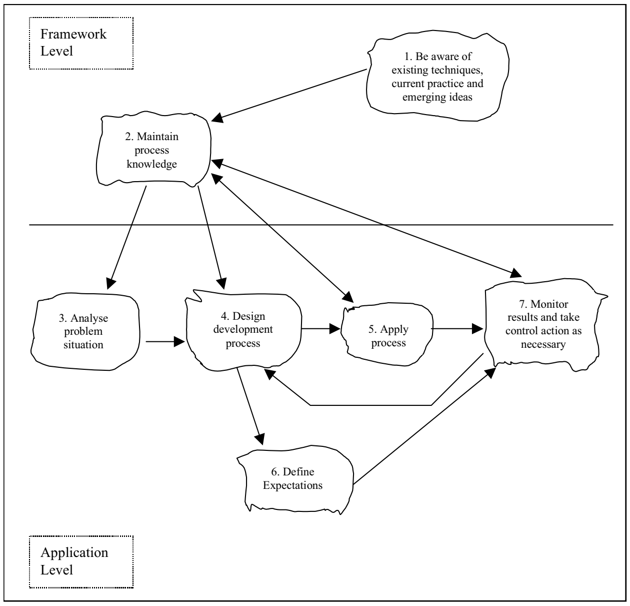
Framework Level
1. Be aware of existing techniques, current practice and emerging ideas
2. Maintain process knowledge
3. Analyse problem situation
4. Design development process
5. Apply process
7. Monitor results and take control action as necessary
6. Define Expectations
Application Level

Figure 1: Process Tailoring Framework

The *framework level* is concerned with gathering and providing information useful in the creation of development processes (2). This is a combination of what can be gathered from external sources (1), such as the ISO 9000 quality series and SPICE, together with examples of well-established practice, such as the RUP or CMMI, and knowledge from local experience of development (7). At a more detailed level, it might also record information about individual techniques for aspects of development such as requirements analysis, architectural design and testing.

There are five activities at the *application level*. The first is the assessment of the problem situation (3) to set the context for the design of the development process (4) and its subsequent application (5). A challenge for successful tailoring is to be able to capture the problem situation in a suitably informative way to enable its characteristics to be matched to process types and instances recorded in the knowledge base. To facilitate "learning" it is necessary to define process expectations in advance (6).

This model raises several related research questions:

- How can processes, process elements and past projects be recorded to capture the essence of what each offers to facilitate selection in a given problem situation?
- What is the minimal effort needed to analyse a problem situation to extract sufficient information to guide the design of the development process?
- How practical is dynamic process adjustment?

4. Justification

The concept of process tailoring is not new. For example, CMMI level 3 requires that a well-understood set of standard processes is defined and that each project is able to tailor these within certain guidelines. Offering such facilities does not, however, mean that they are used as anticipated. For example, RUP users sometimes ignore tailoring and try to include the whole process [1].

Conversely, development organisations claiming to use a particular process will often tailor it. For example, an XP process may use a subset of the defined 12 practices, rather than adopt them all.

Some approaches are *methodologies* offering a framework from which process instances can be defined [5]. For example, each of the Crystal methodologies requires certain roles, policy standards, products and tools to be adopted. Developers are free to decide how to produce the work products and may select practices from other approaches such as XP, SCRUM or DSDM. Some guidance on selection is desirable, however, otherwise it will be unclear whether the resulting process meets the basic needs of the methodology.

UML is another example of an umbrella approach but does dictate how it is used in a development process. In general, there is a need to have clearer guidelines on process frameworks and the selection of elements to complete processes derived from those frameworks.

As might be expected, successful project tailoring is not a trivial task. Jones [6], for example, claims that more than "*100% differences in work effort have been observed for projects of exactly the same size due to variations in the activities performed*".

Software tool support is needed to facilitate process tailoring. To be effective, the tool is likely to use soft case-based reasoning techniques to match the characteristics of a situation to stored knowledge about process structures and experiences.

5. Research Hypothesis

The research hypothesis is that *software engineering can be improved by tailoring each development process to individual circumstances using captured, context-specific process knowledge*.

The expected output of the research will be:

- A survey of existing process tailoring practices, as described in the literature and followed in industry.
- The design of a process-tailoring framework, covering all aspects of software production and maintenance (evolution), from which specific processes can be developed.
- The design of an agile form of problem analysis to facilitate the gathering of information needed for process design.
- Support tools for the framework that maintain a process knowledge base and assist with process construction.
- An evaluation of the resulting process tailoring approach and associated tools in an industrial setting.

6. Details of Proposed Solution

A process knowledge base will have some general information and some that is specific to the organisation in which it is used. The general part will hold techniques for individual aspects of a process and patterns for complete processes. These will be taken from the literature. The specific part is organisational knowledge recorded as a set of process patterns for completed projects. It is expected that case-based reasoning techniques will be used to store, retrieve, and update such patterns.

The desire to support an agile development philosophy is one of the main motivators for investigating process tailoring. It is therefore important that the problem analysis leading to process design be as lean as possible. System development is assumed to be a socio-technical activity, so reasonable attention has to be paid to the organisational context in which the computing system under development will be used.

Soft Systems Methodology (SSM) [7] has been selected as the basic problem analysis technique. This follows an assumption that computing systems are part of a wider human-activity system. Each problem being analysed falls in a spectrum from *hard* (well defined) to *soft* (difficult-to-define, possibly with a social and political aspect). The development of most computing systems has a soft side that should be explored to ensure that system requirements are specified accurately.

SSM, either as a methodology by itself or in combination with other approaches, is a good technique for soft problems [8]. However, it is a relatively slow process [9], designed for effectiveness rather than efficiency. Ways of making it more agile are being considered in this research. One approach, for example, is to make use of domain patterns. In particular, it may be practical to define standard patterns for certain types of organisation, such as banks, hospitals, or universities, from which specific organisational descriptions can be developed.

7. Progress so Far

Work so far has concentrated on clarifying the basic tailoring framework and identifying relevant research and practice in the literature. This includes material on the use of process patterns and the characterisation of projects. Boehm and Turner [10], for example, describe a project characterization approach that uses a 5-point radial diagram covering *size*, *culture*, *dynamism*, *personnel* and *criticality*.

As well as looking at the literature, software development companies in Ireland are being visited to assess their use of agile techniques and the extent to which tailoring occurs in practice. Recent interviews, for example, have revealed a wide range of approaches to problem analysis. In one case problems were initially characterised and classified within a few minutes, as part of a tendering process. The time for subsequent analysis ranged from 20 weeks to a matter of minutes depending on circumstances.

Good relationships have been established with several software development organisations so there is ample support for action research once the tailoring framework has been adequately defined. Action research implies a feedback and refinement cycle consistent with the agile philosophy.

It is also expected that the work will link in with existing development practices such as techniques associated with UML. RUP supported by UML and its business extensions, facilitates business domain modelling, although some weaknesses have been reported [11].

Particular contributions of the research are expected to include:

- A consideration of the implications of the agile philosophy for socio-technical systems development.
- Adaptation of SSM for more rapid deployment.
- Characterisation of software process standards and recommended practices in a form suitable for storage and retrieval.
- Use of case-based reasoning in process description.

This research is at an early stage and it is expected that the scope will be narrowed soon. I would like to acknowledge the contribution of Professor David Bustard and the Centre for Software Process Technologies (CSPT) at the University of Ulster to this work.

8. References

[1] A. Cockburn, *Agile Software Development*, Pearson Education, USA, 2002.

[2] L. Williams, A. Cockburn, "Agile Software Development: It's about Feedback and Change", *IEEE Computer*, IEEE Computer Society, June 2003 (Vol. 36, No. 6) pp 39-43.

[3]. M. Gnatz, F. Marschall, G. Popp, A. Rausch, and W. Schwerin, "The Living Software Development Process", *Software Quality Professional,* American Society For Quality, June 2003, Volume 5 Issue 3.

[4] M. Lycett, R. D. Macredie, C. Patel, R. J. Paul "Migrating Agile Methods to Standardized Development Practice", *IEEE Computer*, IEEE Computer Society, June 2003 (Vol. 36, No. 6).

[5] B. Fitzgerald, "The Use of Systems Development Methodologies in Practice: A Field Study", *The Information Systems Journal*, Blackwell Publishing, 1997 Vol. 7, No. 3.

[6] C. Jones., "Software Cost Estimation in 2002", *CrossTalk The Journal of Defense Software Engineering*, U.S. Department of Defense June 2002.

[7] P. Checkland, *Systems Thinking, Systems Practice. Includes a 30 year retrospective*, Wiley, England, 1999.

[8] I. Munro, J. Mingers, "The use of multimethodology in practice – results of a survey of practitioners", *Journal of the Operational Research Society*, Palgrave MacMillan, Mar 18 2002, pp. 369 – 378.

[9] J. Mingers, S. Taylor, "The Use of Soft Systems Methodology in Practice", *Journal of the Operational Research Society*, Palgrave MacMillan, Apr 01 1992, Vol. 43, No.4.

[10] B. Boehm, R. Turner, "Using Risk to Balance Agile and Plan-Driven Methods", *IEEE Computer*, IEEE Computer Society, June 2003 (Vol. 36, No. 6), pp. 57-66.

[11] A. Dietzsch, "Adapting the UML to Business Modelling's Needs – Experiences in Situational Method Engineering", *LNCS 2460*, Springer-Verlag, Berlin, 2002, pp. 73–83.