

# Tailoring RUP to a Defined Project Type: A Case Study

Geir K. Hanssen<sup>1</sup>, Hans Westerheim<sup>1</sup>, and Finn Olav Bjørnson<sup>2</sup>

<sup>1</sup> SINTEF ICT, N-7465 Trondheim, Norway  
{geir.k.hanssen, hans.westerheim}@sintef.no

<sup>2</sup> NTNU, N-7491 Trondheim, Norway  
finn.olav.bjornson@idi.ntnu.no

**Abstract.** The Unified Process is a widely used process framework for software development. The framework is covering many of the roles, activities and artifacts needed in a software development project. However, a tailoring of the framework is necessary to fit specific needs. This tailoring may be accomplished in various ways. In this paper we describe a concrete attempt to tailor the Rational Unified Process to a defined project type; a Mainstream Software Development Project Type. The paper has focus on the process of creating the tailored Rational Unified Process as well as the resulting Rational Unified Process. The paper makes some conclusions and has a proposition for further research.

## 1 Introduction

The Unified Process [1] and the commercial variant, the Rational Unified Process, RUP [2] are comprehensive process frameworks for software development projects. RUP defines a software development project as a set of disciplines, e.g. requirements handling, implementation etc., running from start to end through a set of project phases. A project is performed by a group of actors, each having one or more well defined roles. Each role participates in one or more activities producing one or more artifacts. A discipline can run in iterations, that is, repetitions within a phase. Activities, roles and artifacts are the basic process elements of RUP.

However, RUP is a comprehensive framework, meaning that it is a more or less complete set of process elements that has to be tailored to each case as no project needs the complete set of elements.

Jacobson, Booch and Rumbaugh says in [1] p.416:

*"It [RUP] is a framework. It has to be tailored to a number of variables: the size of the system in work, the domain in which that system is to function, the complexity of the system and the experience, skill or process level of the project organization and its people."* Further on they say: *"Actually, to apply it, you need considerable further information."*

So, it is clear that RUP needs to be tailored, downscaled and specialized to the context of use. Looking at literature there are not many guidelines on doing this [3], [4], [5] although the need for good practical guidelines and advice definitively is present.

While discussing adaptation of RUP, it is important to have in mind that RUP is a methodology suited for some software development projects, not all. Before you consider using RUP as a basis for your processes you should think of what you really need and what you really do not need. RUP is designed to support four basic properties of software projects: use-case based customer dialogue and documentation, an architecture focus, iterative processes and incremental product development. The idea of adapting RUP is to make it fit each specific project not loosing these properties. It is important to keep the integrity of RUP as a framework. So, an adapted or downscaled variant still defines a project in terms of phases and still describes the work as a complimentary set of disciplines. However, some disciplines may be omitted or even added.

The goal of this paper is to provide others considering remodeling and adapting a process framework in general, and RUP particularly, an insight in how this has been done in a small software company. Some aspects of the specialization process seems to have been working well, others not. This paper presents the adaptation process and also gives an analysis of this process and its result.

The work detailed in this article was carried out as part of a national research project in process improvement and software quality called SPIKE. SPIKE is short for Software Process Improvement through Knowledge and Experience. The participants are SINTEF, NTNU, the University of Oslo and several partners (companies) in the Norwegian ICT-industry. The industrial partners are interested in improving their development process, and are seeking concrete processes and methods to help them deliver high quality software with shorter time to market.

The paper starts with a **Theoretical context**, giving a brief introduction to methodologies and frameworks and various strategies of making these fit specific project needs of process support. It then describes the action research as the **Research method** of choice. The rest of the paper is arranged according to the research method phases; **Diagnosing**, **Action planning**, **Action taking**, **Evaluating** and **Learning**. Finally a **Conclusion** is given and **Further research** suggested.

## 2 Theoretical Context

### 2.1 Software Development Methodology and Frameworks

The term methodology is defined as "A body of methods, rules, and postulates employed by a discipline: a particular procedure or set of procedures" by the Merriam-Webster dictionary [6]. Basically, a methodology describes how someone, e.g. an organization performs a task, e.g. software development. In a broad sense, a software development methodology describes aspects such as how to communicate with customers, sales strategy, how to describe requirements, use of tools, test practices, documentation, planning, reporting and so on. In our context we talk about methodologies for running projects with a defined customer having more or less defined goals initially. Besides describing techniques, roles etc. most methodologies are based on a set of basic values. Examples are *User centric*, *Architecture centric*, *Agile*, *Risk driven* and many more. RUP has four basic values: *Use-Case Driven*,