

# Introducing Tool Support for Retrospective Analysis of Release Planning Decisions

Lena Karlsson and Björn Regnell

Department of Communication Systems, Lund University,  
Box 118, 221 00 Lund, Sweden  
{lena.karlsson, bjorn.regnell}@telecom.lth.se

**Abstract.** The release planning activity in market-driven requirements engineering is crucial but difficult. The quality of the decisions on product content and release timing determines the market success, but as predictions of market value and development cost are uncertain, the decisions are not always optimal. This paper presents a prototype tool for retrospective analysis of release planning decisions based on tool requirements gathered in two previous empirical studies where retrospective analysis was done manually. The supported method enables representation of different views in the decision-making process. The results from an initial validation indicate that the retrospective analysis benefits from the supporting tool.

## 1 Introduction

The purpose of this paper is to describe a tool<sup>1</sup> that supports a method for retrospective analysis of release planning decisions. The method is called PARSEQ (Post-release Analysis of Requirements SElection Quality) and aims at finding process improvement proposals for the release planning activity. The retrospective analysis is acknowledged as an important means for software process improvement [9]. Release planning is regarded as one of the most critical activities in market-driven software development [2] as it can determine the success of the product. Generation of feasible assignments of requirements to increments in a changing environment is a very important but complex task [3]. The PARSEQ method has been successfully investigated in two prior case studies performed in industry. The experiences from the case studies provided the basis for developing the tool support and were used to elicit the PARSEQ tool requirements.

In [7], the method was applied at a company developing a software product for an open market. The company had regular releases and used a requirements management (RM) tool [13] when planning their releases. The PARSEQ analysis yielded a number of process improvements such as enhancing the overall picture of related requirements, increased attention to the elicitation of usability requirements and improved estimates of implementation costs. In the second case study [8], the method was applied using a different approach. The investigated project had an agile development procedure, inspired by the Extreme Programming (XP) [1]. The project

---

<sup>1</sup> Download at <http://serg.telecom.lth.se/research/packages/ParseqTool/index.html>

was conducted in-house, i.e. both users and developers of the project were from within the company. Since the project used an agile approach to development, they had frequent iterations and regular releases when the system was put into operation. In each iteration the project used the Planning game [1] to prioritise the requirements and plan the next release. The requirements were elicited in the beginning of the project from internal stakeholders and documented in an excel sheet.

The experiences from the case studies have given input to the needed functionality of the PARSEQ tool support. We believe that tool support may improve the retrospective analysis in the following ways:

- Increase efficiency as more requirements may be analysed compared to the manual case
- Increase visualisation potential of release planning problems through charts and diagrams
- Decrease preparation and manual handling of requirements through import and export possibilities

The tool was validated in a retrospective analysis of the development of the PARSEQ tool itself. The analysis indicated that the tool is well functioning and we believe it will be helpful in future case studies.

The paper is structured as follows. Section 2 describes the PARSEQ method and Section 3 describes the PARSEQ tool that has been developed. Section 4 discusses some threats to validity and concludes the paper with suggestions for further work.

## 2 The PARSEQ Method

Retrospective evaluation is used for different purposes within software engineering. Some retrospective methods use metrics to evaluate a product from a certain perspective, such as maintainability or program structure, in order to improve software evolution [10]. Other methods aim at evaluating a conducted project in order to improve future projects [9]. Unlike these retrospective analysis methods, PARSEQ focuses on finding release planning process improvements through an analysis of earlier release planning decisions.

Release planning is used when developing products in an incremental manner, and provides the opportunity of releasing the most important functionality first instead of delivering a monolithic system after a long development time [3]. Release planning is used in incremental development and agile methods. Assigning requirements to increments is a complex task as many factors influence, such as different stakeholder needs, available resources, and technical precedence constraints. There are several techniques for release planning e.g. EVOLVE [3] and Planning game [1]. There are also a number of tools to support release planning e.g. ReleasePlanner [14] and VersionOne [15]. In contrast to these tools, the PARSEQ tool is not intended as a release planning tool, but as a tool for evaluating releases in retrospect. Thus, the aim is not to come up with a perfect release plan but to find process issues that need improvement in order to reach better release plans in the future.

The PARSEQ method is based on a systematic analysis of candidate requirements from previous releases. By identifying and analysing a set of root causes to suspected