Chapter VI

Maturing Requirements Engineering Process Maturity Models

Pete Sawyer, Lancaster University, UK

Abstract

The interest in Software Process Improvement (SPI) in the early 1990s stimulated tentative work on parallel models for Requirements Engineering (RE) process improvement in the late 1990s. This chapter examines the role of SPI and the implications of the exclusion of explicit support for RE in the most widely used SPI models. The chapter describes the principal characteristics of three RE-specific improvement models that are in the public domain: the Requirements Engineering Good Practice Guide (REGPG), the Requirements Engineering Process Maturity Model (REPM), and the University of Hertfordshire model. The chapter examines the utility of these models and concludes by considering the lessons learned from industrial pilot studies.

Introduction

The risks posed to software development projects by weak requirements engineering (RE) practice have become widely recognized during the last decade. This has spawned a great deal of investment in RE methods, tools, and training by practitioner organizations and in RE research by the wider software and systems engineering communities.
The focusing of attention on RE during the early 1990s coincided with the deployment of software process improvement (SPI) that was stimulated by Humphrey’s pioneering work in the 1980s (Humphrey, 1989). However, a European survey of organizations engaged in SPI programs during this period (Ibanez & Rempp, 1996) confirmed that the SPI models then available offered no panacea for RE problems. Indeed the organizations consulted identified requirements specification and the management of customer requirements as the principal problem areas in software development that they faced. Even enthusiastic adopters of SPI programs found that while SPI brought them significant benefits, their problems with handling requirements remained stubbornly hard to solve.

The Software Engineering Institute’s Capability Maturity Model for Software (SW-CMM) (Paulk, Curtis, Chrissis, & Weber, 1993), which was becoming widely deployed at this time, touched on RE practices but provided little specific guidance. To redress this, the Requirements Engineering Adaptation and IMprovement for Safety and dependability (REAIMS) project conducted the first systematic application of the principles of SPI specifically for RE. This resulted in the publication of the Requirements Engineering Good Practice Guide (REGPG) (Sawyer, Sommerville, & Viller, 1999; Sommerville & Sawyer, 1997) in 1997.

This chapter reviews the state-of-the-art of process improvement for RE. It starts by reviewing the background to process improvement in the software and systems engineering industry. It then considers the nature of RE processes and the pressures and trends that have merged in recent years. It argues that for sociotechnical systems, RE practice needs to be particularly strong. It then reviews three RE process improvement methods and examines the extent to which they have been validated. The chapter concludes by summarizing the options open to organizations seeking to systematically improve their RE processes.

### SPI Models and Standards

Humphrey’s pioneering work on Software Process Improvement (SPI) in the 1980s (Humphrey, 1989) led to the development of the *Capability Maturity Model for Software* (SW-CMM), developed at the Software Engineering Institute under the sponsorship of the United States Department of Defense. Humphrey’s work reflected a realization that the piecemeal adoption of better methods and tools would not deliver the improvements in software quality increasingly demanded by customers. Rather the whole development lifecycle needed to be addressed in order to identify weak areas and focus improvement efforts. From the customer’s perspective, SPI allows software contractors to be assessed against a common model and provides a stimulus for contractors to increase product quality and meet cost and delivery targets. From the contractor’s perspective, SPI represents a strategic organizational tool for containing costs and increasing market-share.

The SW-CMM does this by defining a generic model of the software development process structured around five *maturity levels*. Process maturity represents the degree to which a process is defined, managed, measured, controlled, and effective (Paulk,
Related Content

Innovating Healthcare through Remote Monitoring: Effects and Business Model
www.irma-international.org/article/innovating-healthcare-through-remote-monitoring/144814/

Integrated Software Testing Learning Environment for Training Senior-Level Computer Science Students
www.irma-international.org/chapter/integrated-software-testing-learning-environment/29601/

Model-Driven Requirements Specification for Software Product Lines
www.irma-international.org/chapter/model-driven-requirements-specification-software/49167/

Test Case Reduction Using Data Mining Technique
www.irma-international.org/article/test-case-reduction-using-data-mining-technique/166543/

Experiences with Modelling Early Requirements
www.irma-international.org/chapter/experiences-modelling-early-requirements/23416/