

Chapter 2

Modeling Software Development Processes

Gerhard Chroust

Kepler University Linz, Austria

Marco Kuhrmann

Technische University München, Germany

Erwin Schoitsch

Austrian Research Centers – ARC, Austria

ABSTRACT

In this chapter the authors discuss the WHY and WHAT of modeling software development processes: defining the components of a software process and proposing 5-dimensional grid of attributes of existing models: strategy and path, levels, main subprocesses, components and aura. Specific process models, currently used or historically important, are described. This is followed by an extensive discussion of methods for and problems of modeling a software process, followed by a shorter discussion on the enactment of process models via software engineering environments. The chapter closes with a discussion of the human aspects concerning introduction and enactment of a process model.

INTRODUCTION

The Need for a Defined Software Process

Information and Communication Technology (ICT) has become the no. 1 business driver of economy and the key to practically all human activities. The ongoing reduction of the size and the cost of computers together with increased speed and performance allows the implementation of functionalities

completely unimaginable even a decade ago. This creates a demand for more software with higher sophistication and with higher quality. Additional requirements are shorter time-to-market together with flexibility and adaptability fo software with respect to answering user needs.

Since approximately 1970 (Royce, 1970) it is understood that the development of a software system is a multi-phase process. Each phase is concerned with contributing a different view and focus of the software artifact to be produced. In order to guarantee an orderly, systematic process this stepwise process should be defined and described

DOI: 10.4018/978-1-60566-856-7.ch002

by a so-called process model. Based on strong statistical evidence this has been supported by Watts Humphrey from the Software Engineering Institute (Humphrey, 1989) introducing the Capability Maturity Model (CMM). It claimed that a defined process model and its appropriate enactment was one of the key steps in order to reach a development capability beyond a certain basic level. Initially only big companies produced 'industrial' software. Due to the multitude of procedures and rules existing in these companies (they often also produced hardware using an appropriate 'development process'), the introduction of a 'software development process' was a somewhat natural innovation (Wang & King, 2000). The trend was intensified by the US Department of Defense's requirement for software contractors to show (and prove) the appropriate level of capability in order to be awarded a contract.

The reduction of the size and the cost of computers (down to PDAs and sophisticated mobile phones) together with an equally dramatic increase in performance and storage has not only provided computers for everybody, but has also created business opportunities for numerous small and very small companies (SMEs), often consisting of one or a few persons, often start-up companies based on a single innovative idea. They profit from the low initial investment, the ability to offer and sell their product world wide via Internet without intermediary or sales organizations. Due to their small size they are able to provide high flexibility and customer-orientation.

Closer investigation of these small and medium companies shows (SPIRE Project Team, 1998) that they often have a highly chaotic, invention-driven production process and a disregard of the associated quality and management processes. As a consequence they cannot guarantee sustainable, verifiable quality, and business return on investment.

In this chapter we present the state-of-the-art of software development process models with respect to the various types, their components,

the strategies they incorporate, giving examples of existing models. We discuss aspects of creating a process model and various human aspects, including issues of introduction of process models. It must be stressed that a software process model in itself is only useful if enacted properly. Therefore we discuss enactment of process models which leads to a short discussion of Software Engineering Environments (for details see Kuhrmann et al. (2010) in this book).

Prominent Process Models are listed in "Selected Process Models - Overview" with their references, therefore in the rest of the text no references to them are given.

BACKGROUND

Software Quality and Software Processes

Quality is the key to successful, sustainable software development. Low quality increases the cost for error correction during development and for maintenance after the delivery of a product. Obviously low quality also reduces market acceptance and hampers later evolution of a product.

In the various disciplines of engineering many different approaches exist to guarantee, to assist or at least to check ex-post the quality, especially the correctness of an industrial product in relation to defined and implicit requirements. The correctness of a product can be established beyond doubt if it is constructed following a proven algorithm or if it has exhaustively been tested. Other methods can only establish correctness to a certain degree (e.g. parallel or inverse construction). Some methods carry over to software engineering, e.g. foolproofing (Nakajo, 1985), inspections (Gilb & Graham, 1993), formal development methods, prototyping, following a well-established process, documentation of work, etc.

For software development (which basically is an engineering discipline!) many of the approaches

30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/modeling-software-development-processes/37907

Related Content

Information Technology as a Service

Robin G. Qiu (2007). *Enterprise Service Computing: From Concept to Deployment* (pp. 1-24).

www.irma-international.org/chapter/information-technology-service/18476

Next-Generation IT for Knowledge Distribution in Enterprises

Ramón Brena, Gabriel Valerio and Jose-Luis Aguirre (2011). *Enterprise Information Systems: Concepts, Methodologies, Tools and Applications* (pp. 1836-1846).

www.irma-international.org/chapter/next-generation-knowledge-distribution-enterprises/48647

Assessment Strategies for Servant Leadership Practice in the Virtual Organization

Darin R. Molnar (2010). *Leadership in the Digital Enterprise: Issues and Challenges* (pp. 181-193).

www.irma-international.org/chapter/assessment-strategies-servant-leadership-practice/37095

A Maturity Model of Strategic Information Systems Planning (SISP): An Empirical Evaluation Using the Analytic Network Process

Zijad Pita, France Cheong and Brian Corbitt (2011). *International Journal of Enterprise Information Systems* (pp. 30-57).

www.irma-international.org/article/maturity-model-strategic-information-systems/58045

E-CRM and CMS Systems: Potential for More Dynamic Businesses

Silvia Brito Fernandes (2010). *Social, Managerial, and Organizational Dimensions of Enterprise Information Systems* (pp. 157-171).

www.irma-international.org/chapter/crm-cms-systems/37913