Chapter I
Managing the Quality of UML Models in Practice

Ariadi Nugroho
Leiden University, The Netherlands

Michael R. V. Chaudron
Leiden University, The Netherlands

ABSTRACT

Many studies have been carried out to investigate what makes up good quality software. Some of the early models that define the quality of software come from Boehm (1976) and McCall (1977). Works in this field of quality models have traditionally focused on quality of the final software product. Since the 1970’s models of software have been used and this has recently attracted much attention through the popularity of model-driven software development (MDSD). However, quality of software models has rarely been considered (Lange & Chaudron, 2005). In the software development life cycle, the ability to assure software quality long before the testing phase may save a lot of money since less defects found in the testing phase will mean less effort to be allocated for rework. Currently, the importance of model quality is starting to gain attention from computer scientists. Work in this area has since focused on developing tools, metrics, and frameworks to improve the quality of models that guide implementation, particularly in the context of UML modeling which has become the de facto standard for building object oriented software. Quality of models can be considered from many different perspectives. In this chapter, we will consider the following perspectives: Firstly, is the model complete in the sense that it describes the information that developers need to know about a system? Secondly, we look at the degree in which a model of a system and an implementation correspond. This degree of correspondence indicates to what extent analyses of—or predictions based on the model are valid for the implementation. We present the main findings from case studies into quality of modeling in the software industry as well as findings from a survey amongst professional software developers. We also provide a discussion on the contemporary methods for design quality assessments.
INTRODUCTION

Despite the fact that the notions of good quality software have been around since four decades ago, many software companies are still struggling to get their software product into production without numerous defects. Defects can be interpreted as deviation from specification or expectation (Fenton & Neil, 1999).

Since defects will eventually affect the operation of software as the final product, the discussion on defects cannot neglect the notion of software quality. In general terms, the notion of quality is the absence of defects. Thus, if defect means deviations from specification or expectation, we can perceive quality as a conformance to specification and requirements/expectations.

In their search of qualifying aspects in software quality, computer scientists have come up with quality models that are generally constructed by quantitative approaches. Two of the most renowned quality models came from the work of Boehm, Brown, and Lipow (1976) and McCall, Richards, and Walters (1977). Boehm’s quality model is shown in Figure 1.

While quality models are generally more focused on the quality characteristics of the final software product, many efforts have been devoted to prescribe standard procedures and processes so that eventually software will have the quality attributes as have been defined in many quality models. In this regard, SEI (Software Engineering Institute) has come up with the Capability Maturity Model (CMM) that is currently becoming the

![Figure 1. Boehm’s quality model (©2007 Ariadi Nugroho. Used with permission)](image-url)