Chapter 1
Quality-Driven Software Development for Maintenance

Iwona Dubielewicz
Wroclaw University of Technology, Poland

Bogumila Hnatkowska
Wroclaw University of Technology, Poland

Zbigniew Huzar
Wroclaw University of Technology, Poland

Lech Tuzinkiewicz
Wroclaw University of Technology, Poland

ABSTRACT

Software maintenance is sometimes considered as a special kind of activity that is separated from a software development process. Meanwhile, the opposite is true; maintenance should be taken into account from the beginning of the software development process. Because a model-based software development is the prevailing software development paradigm, the maintainability should be considered within models that arise in software development process. We claim that the quality of the models arising in the software development process has a positive influence on their maintainability: the higher quality of the models, the more effective maintainability activity. The background for our consideration is MDA approach, and the scope of the consideration is limited to perfective maintenance only. The set of so called 6C quality characteristics is assumed to define a quality of MDA-models. Our selection of 6C quality characteristics is justified by the fact that they are related to activities performed on models within the maintenance. To assess MDA-models in the context of the maintainability, we define checklists for the 6C characteristics. These checklists are used for derivation of some measures which are useful in checking to what scope a given characteristics is satisfied. The main advantage of the approach is its independence of the knowledge of future changes of user requirements that trigger perfective maintenance. In the chapter, we demonstrate a simple example of how to assess the quality of PIM-models that are the realization of the CIM-model. Additionally, we discuss how to select, for further development, the best PIM-model from the set of possible solution.

DOI: 10.4018/978-1-61350-438-3.ch001

INTRODUCTION

Software maintenance is an integral part of a software life cycle. It means that software maintenance should not be considered without the context of software life cycle. The consequence is that "the main problem in doing maintenance is that we cannot do maintenance on a system which was not designed for maintenance" (Schneidewind, 1987). So, our basic assumption when dealing with maintenance is manifested by the phrase: *Don't think about software maintenance if you didn't think about it during software development process*.

Nowadays, dominating approaches to software development form the group called Model Development Engineering (MDE). It means that models are considered as basic artifacts elaborated within the software development process. One of the MDE approaches, Model Driven Architecture (MDA), (Kleppe, Warmer & Bast, 2004)) introduces three categories of models: Computer Independent Model (CIM), Platform Independent Model (PIM), and Platform Specific Model (PSM), and proposes the software development as a process of elaboration and transformation of the models according to the schema:

```
CIM → PIM → PSM → Code
```

It means that on the basis of initial CIM model, first PIM and next PSM models are elaborated, and finally PSM model is transformed into Code.

In the chapter, we follow the scheme of MDA. This pattern may be used repeatedly in the life of the software product. The first application of this scheme is associated with the initial development of the software product – with the issue of the first release of a software product. The subsequent application of the same scheme may result from the need of the product maintenance, and is concerned with the issue of a new release of the software product. The scope of subsequent applications of MDA scheme depends on the kind of maintenance. ISO vocabulary (ISO/IEC 24765, 2009) defines three kinds of maintenance: adaptive, perfective and corrective. In the chapter, we concentrate on perfective maintenance, and we have omitted adaptive and corrective maintenance. The reasons are twofold. First, perfective maintenance covers about 60% of total maintenance costs (Canfora & Cimitile, 2000; Deissenbock, 2009), and is concerned with a change in the problem domain. Second, adaptive maintenance is concerned with the solution domain and corrective maintenance concentrates mainly on modification of code. Additionally, perfective maintenance entails modification of all MDA models, while adaptive maintenance entails modification of PSM model and code, and corrective maintenance entails only modification of code.

The MDA models may be defined in different ways depending on the applied software development methodology. There are many methodologies that are based on the MDA approach (Kleppe, Warmer & Bast, 2004). In the chapter, to define the MDA models we use Quality Driven Software Development (QUAD) methodology (Dubielewicz, Hnatkowska, Huzar & Tuzinkiewicz, 2010). QUAD belongs to the group of Unified Process methodologies. The main reason for QUAD selection is its specificity, i.e. the quality and evaluation models are applied to selected artifacts on each stage of software development. Maintainability is one of the quality characteristics defined in (ISO/IEC 9126-1, 2001), and models are the main artifacts of the developing software. Therefore, maintainability may be assessed in the same way as it has been proposed in QUAD methodology for software quality evaluation. The knowledge of QUAD is not necessary because the required elements are introduced and explained in further presentation.

Maintainability, similarly to other ISO quality characteristics (ISO/IEC 9126-1, 2001), may be considered from external and internal perspective of quality specification.