Chapter IX

A Survey of Object-Oriented Design Quality Improvement

Juan José Olmedilla, Almira Lab, Spain

Abstract

The use of object-oriented (OO) architecture knowledge such as patterns, heuristics, principles, refactoring and bad smells improve the quality of designs, as Garzás and Piattini (2005) state in their study; according to it, the application of those elements impact on the quality of an OO design and can serve as basis to establish some kind of software design improvement (SDI) method. But how can we measure the level of improvement? Is there a set of accepted internal attributes to measure the quality of a design? Furthermore, if such a set exists will it be possible to use a measurement model to guide the SDI in the same way software process improvement models (Humphrey, 1989; Paulk, Curtis, Chrissis, & Weber, 1993) are...
guided by process metrics (Fenton & Pfleeger, 1998)? Since (Chidamber & Kemerer, 1991) several OO metrics suites have been proposed to measure OO properties, such as encapsulation, cohesion, coupling and abstraction, both in designs and in code, in this chapter we review the literature to find out to which high level quality properties are mapped and if an OO design evaluation model has been formally proposed or even is possible.

Introduction

In the last two decades there has been a growing interest and effort put after the idea of improving the quality of the software processes (Humphrey, 1989). This increasing trend had it origin in the application of statistical process control techniques (Oakland, 1990) from the manufacturing industry to our sector, thus creating a new discipline that has been called software process improvement (SPI) (Humphrey, Snyder & Willis, 1991). This discipline aids organisations to improve their software producing processes by, firstly, identifying all the broad areas of the process, their goals and the activities and sub-activities needed to achieve them and secondly by establishing a path through which the process can be incrementally improved, this path is a set of quality levels, each of them defined by the areas and their associated goals to be accomplished. Fundamental to the SPI are the associated metrics (Fenton & Pfleeger, 1998) that are the tool by which the organisation can tell at each moment where it is in the path, each of the aforementioned goals has an associated set of metrics that help to tell if it has been achieved and to what extent. Although there are alternative SPI models and methods, like CMMI (Paulk et al., 1993) or SPICE (ISO/IEC, 1999), an organisation can always adhere to a concrete definition of process quality and a way to measure it and improve it.

However the product arena does not seem to be so established in terms of quality improvement models. A fundamental question that managers and developers often face is when it is worth to improve a software product by reengineering it or on the contrary start it all over from scratch. Fowler (2000) states that

*There are times when the existing code is such a mess that although you could refactor it, it would be easier to start from the beginning ... I admit that I don’t really have good guidelines for it.* (p. 66)

In this case Fowler was talking about the refactoring technique but something similar can be said about other OO design knowledge elements. There is plenty of knowledge, more or less formalised, about identifying situations in which to