Problem management is the dominating process within corrective maintenance. It not only handles software problems but also provides quantitative feedback important for assessing product quality, crucial for continuous process analysis and improvement and essential for defect prevention.

Chapter 2 gives a detailed description of the problem management process and its constituent phases, activities, and roles involved. It also provides (1) general definitions on the basic terminology utilized within corrective maintenance, (2) roadmaps visualizing the corrective maintenance from different perspectives, (3) general descriptions of the most important issues within support (upfront maintenance) process – the process tightly collaborating with the problem management process, and finally, (4) it gives a glimpse into Corrective Maintenance Maturity Model (CM³): Problem Management – a process model specialized to only problem management process within corrective maintenance.

INTRODUCTION

Corrective maintenance is a very important process for achieving process maturity. It not only handles the resolution of software problems, but also provides a basis for quantitative feedback important for assessing product quality and reliability, crucial for continuous process analysis and improvement, and essential for defect prevention and necessary for making different kinds of decisions. Yet, the domain of corrective maintenance has received very little attention in the curriculum within academia or industry. Current literature provide very coarse-grained descriptions. Extant process models dedicate at most one or two pages to describing it.
In large software organizations, a corrective maintenance process is seldom monolithic. Instead, it is a complex family of various processes collaborating with each other, where each process executes a clearly defined task. In this chapter, we present one of the main processes within corrective maintenance – problem management process. We also illuminate the most important issues within support (upfront maintenance) process – the process tightly collaborating with the problem management process.

The first section, “Software Maintenance,” gives a general introduction to the definition of maintenance and its categories. It also illuminates background and problems concerning these definitions. The next section, “Corrective Maintenance,” presents corrective maintenance and its basic terminology. The sections, “Maintenance Organizations,” “Product Perspective,” and “Life Cycle Perspective” visualize the corrective maintenance process from the organizational, product, and life cycle perspective, respectively. Then, “Problem Management at the Maintenance Execution Process Level” describes problem management process, its phases, and roles involved in its execution. Finally, the last section, “Corrective Maintenance Maturity Model: Problem Management,” gives a glimpse into Corrective Maintenance Maturity Model (CM³) – a process model specialized to only corrective maintenance. Our description of problem management in this chapter is based on the theory extracted from CM³ (Kajko-Mattsson, 2001a).

**Software Maintenance**

What is corrective maintenance and how does it relate to other types of maintenance work? To explain this, we must place corrective maintenance within total maintenance and identify its relation to other maintenance categories. The majority of the software community follows the IEEE suggestion for defining and categorizing maintenance. As depicted in Table 1, the IEEE defines software maintenance as “the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or to adapt to a changed environment” (ANS//IEEE STD-610.125 1990).

Not everybody agrees on this definition today. There prevails a controversy on the choice of maintenance scope, its constituents, time span, and on drawing a dividing line between software development and software maintenance (Chapin, 2001; Kajko-Mattsson, 2001d; Kajko-Mattsson, 2001f; Parihk, 1986; Schneidewind, 1987; Swanson, 1999). This is clearly visible in so widely varying cost estimates of software maintenance, spanning between 40%-90% of the total software life cycle cost (Arthur, 1988; Boehm, 1973; Cashman & Holt, 1980; DeRoze & Nyman, 1978; Glass & Noiseux, 1981; Jones, 1994; Mills, 1976).

According to Martin and McClure (1983), Pigoski (1997), and Schneidewind (1987), the definition suggested by the IEEE states that maintenance is entirely a
Supporting Consistency during the Development and Evolution of Quality Unified Use-Misuse Case Models
www.igi-global.com/article/supporting-consistency-during-the-development-and-evolution-of-quality-unified-use-misuse-case-models/142038?camid=4v1a