4
The Capability Maturity Model

4.1 Introduction

The Capability Maturity Model (CMM©) is a process maturity model which enables an organization to define and evolve its software processes. It is a premise in software engineering that there is a close relationship between the quality of the delivered software product and the quality and maturity of the underlying software processes. Consequently, it is important for a software organization to devote attention to the software processes as well as to the product. The CMM is a framework by which an organization may mature its software processes. It has been influenced by the ideas of some of the leading figures in the quality movement, such as Crosby, Deming, and Juran, etc.

Crosby's quality management maturity grip describes five evolutionary stages in adopting quality practices [Crs:80]. Crosby's ideas were adopted, refined, and applied to software organizations by Watt Humphrey in Managing the Software Process [Hum:89] and in early work done at the Software Engineering Institute. This lead to the development of a maturity model termed the Process Maturity Model (PMM) by the Software Engineering Institute [Hum:87]. The PMM is a questionnaire-based approach to process maturity, and subsequent work and refinement of the PMM lead to the Capability Maturity Model. The CMM is, in effect, the application of the process management concepts of total quality management (TQM) to software. The main rationale for the development of the CMM was the need of the Department of Defense (DOD) to develop a mechanism to evaluate the Capability of software contractors.

The CMM v1.0 was released in 1991, and following pilots it was revised and released in 1993 as v1.1. The Software Engineering Institute has been working on the CMMI© project, and the objective of this project is to merge the software CMM and the Systems CMM, and also to make the CMM compatible with SPICE (15504), the emerging international standard for software process assessment. The CMMI v1.0 was released in July 2000 and is expected to replace
the CMM in the future. SPICE is expected to be released as an international standard in the future, and it currently exists as an ISO technical report. It is termed a type 2 technical report, and this indicates that full agreement on its definition as a standard has not been achieved at this time but is expected to be achieved in the future.

The real benefit of the CMM is that it allows the organization to follow a logical path in improvement, and to evolve at its own pace, via the evolutionary path of satisfying the CMM maturity levels. Each maturity level provides a foundation for further improvements, and the maturity levels cannot be skipped. There is an increase in capability associated with greater maturity, and this enhanced capability is reflected in quality, timeliness of projects, reliability, etc. The result of a successful CMM improvement program is an evolution from an immature and poorly defined software process to a defined, measured, controlled, mature, and effective software process.

The maturity of an organization is a rating given to an organization following an external CMM assessment of the organization by a qualified external assessment team. The capability of the organization provides an indication of the expected results in its software projects. The expectation is that a highly mature organization will deliver the agreed functionality on time, within budget, with high quality, and reliability.

A mature organization is expected to have sound project management, configuration management, requirements management, and quality management practices in place. The process performance represents the actual results achieved by following the software process, and is contrasted with the expected results or software process capability. The immature organization has no objective mechanism to assess the quality of the product and little understanding of the importance of building quality into the product via peer reviews and testing. Immature organizations have little understanding as to how the steps in the software process may affect the quality of the product.

There are three key aspects of good software engineering, namely, people, technology and process (Fig. 4.1). Most problems are due to a defective process, and the focus is to fix the process rather than blaming people. This leads to a culture of openness in discussing problems, and identifying solutions.

![Figure 4.1: Process Triangle](image-url)