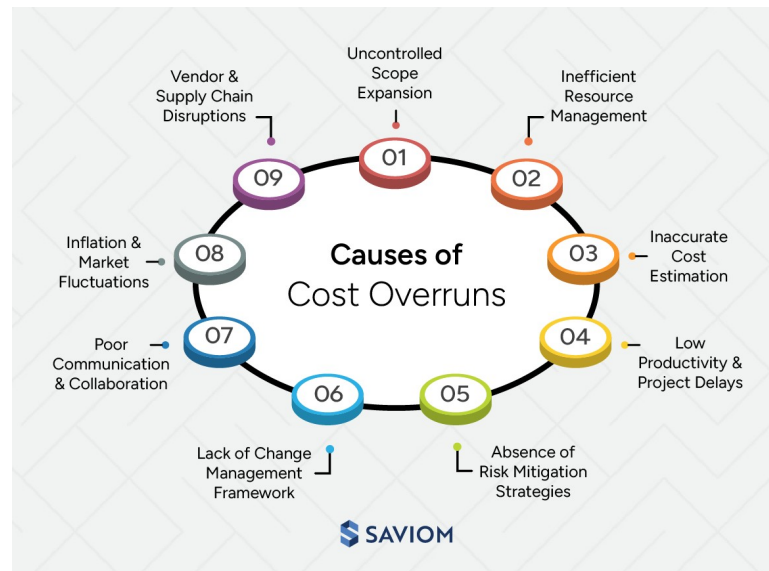


Customer satisfaction has become the motto of many organizations attempting to survive and thrive in today's increasingly competitive world. At the same time that organizations are focusing on customer satisfaction, there is a growing perception that software quality is the weak link in developing high-quality products. The software crisis that has persisted over the last two or three decades has become even more critical as software pervades our day-to-day lives.

The term "software crisis" may be unfair. Software is pervasive because of its power, and software technology is advancing at a breath-taking pace. Despite these advances, it seems that the complexity of the problems being addressed by software is growing more rapidly than our ability to develop and maintain the software. The ability to develop and deliver reliable, usable software within budget and schedule commitments continues to elude most software organizations.

The search for solutions to these challenges has continued for many years. After two decades of unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, organizations are now realizing that their fundamental problem is the inability to manage the software process. In many organizations, projects are often excessively late and over budget, and the benefits of better methods and tools cannot be realized in the maelstrom of an undisciplined, chaotic project.



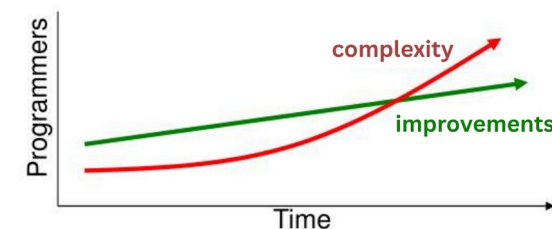
Excerpts from Software Engineering Institute. (1994). *The Capability Maturity Model*. Reading, MA. Addison Wesley.

Examples of software catastrophes abound. An unpublished review of 17 major Department of Defense (DoD) software contracts found that the average 28-month schedule was missed by 20 months. One 4-year project was not delivered for 7 years; no project was on time. Deployment of the B1 bomber was delayed by a software problem, and the \$58 billion A12 aircraft program was canceled partly for the same reason.

One recent Government Accounting Office (GAO) report [GAO-93-13] on the major software challenges states, "We have repeatedly reported on cost rising by millions of dollars, schedule delays of not months but years, and multi-billion-dollar systems that don't perform as envisioned." After summarizing more than 20 GAO case studies involving software or software-related problems in the military, this report concludes, "The understanding of software as a product and of software development as a process is not keeping pace with the growing complexity and software dependence of existing and emerging mission-critical systems."

While the military has its own unique problems, industry has its share of problems as well. Managers in industry have said, "I'd rather have it wrong than have it late. We can always fix it later." The consequences of this attitude on customer satisfaction seem clear. For many organizations, whether military or commercial, the message is that large software projects mean potential trouble.

One task force that investigated the "software crisis" [DoD87] wrote that "few fields have so large a gap between best current practice and average current practice," and went on to conclude that "today's major problems with military software development are not technical problems, but management problems."



To understand which improvements are most critical to an organization, one must first understand the differences between immature and mature software organizations. In an immature software organization, software processes are generally improvised by practitioners and their managers during the course of the project. Even if a software process has been specified, it is not rigorously followed or enforced. The immature software organization is reactionary, and its managers are usually focused on solving immediate crises (better known as fire fighting). These organizations routinely exceed schedules and budgets because they are not based on realistic estimates. When hard deadlines are imposed, they may compromise product functionality and quality to meet the schedule.

In an immature organization, there is no objective basis for judging product quality or for solving product or process problems. There is little understanding of how the steps of the software process affect quality, and product quality is difficult to predict. Moreover, activities intended to enhance quality, such as reviews and testing, are often curtailed or eliminated when projects fall behind schedule. The customer has little insight into the product until delivery.

The Immature Organization

- Software processes are generally improvised during the course of a project.
- Managers are reactionary, solving immediate crises.
- Functionality and quality are compromised when hard deadlines approach. "The death march to Egghead."
- No objective basis for judging product quality; thus quality is difficult to predict.

A mature software organization, in contrast, possesses an organization-wide ability for managing software development and maintenance processes. It accurately communicates the software process to both existing staff and new employees, and carries out work activities according to the planned process. The processes mandated are documented, usable, and consistent with the way the work actually gets done. The process definitions are updated when necessary, and improvements are developed through controlled pilot-tests and/or cost benefit analyses. There is broad-scale, active involvement across the organization in improvement activities. Roles and responsibilities within the process are clear throughout the project and across the organization.

In a mature organization, managers monitor the quality of the software products and the process that produces them. There is an objective, quantitative basis for judging product quality and analyzing problems with the product and process. Schedules and budgets are based on historical performance and are realistic; the expected results for cost, schedule, functionality, and quality of the product are usually achieved. In general, the mature organization follows a disciplined process consistently because all of the participants understand the value of doing so, and the necessary infrastructure exists to support the process.

The Mature Organization

- The software process is accurately communicated to existing and new staff.
- Work activities are actually carried out according to the planned process.
- The mandated processes are *usable* and consistent with the way work actually gets done.
- Roles and responsibilities are clear, schedules and budgets are based on historical performance.