

Applying a Process Improvement Model to SoftBench 5.0

Software organizations are under market pressure to reduce their cycle time and improve their development processes. The conventional approach is to work on one, usually at the expense of the other. For SoftBench 5.0 we decided to jump right in and attack both using a 12-month release cycle and CMM (Capability Maturity Model) level-2 processes. Using CMM-prescribed project management processes, we reduced SoftBench 5.0's cycle time by 35%, improved product usability, and improved our ability to predict release dates. We also greatly improved the organization's ability to select, plan, estimate, and track software projects.

Reference 1 describes the software improvement project at our division that put in place the CMM process. Here we briefly summarize CMM and our approach to using it for SoftBench 5.0.

Business Environment

SoftBench is an integrated application development environment for C, C++, and COBOL running on UNIX systems. It was first released in 1988. Since then the cycle time (that is, the time between one major release and the next) has varied from 18 to 24 months. In previous releases of SoftBench, the first part of the project was very unstructured. It typically involved market research, customer visits, prototyping, and design, but these activities were not well-integrated. At some point we would decide what functionality should make the release and what functionality would be rescheduled for the next release. A cross-functional team would be put into place to manage and focus the release. This model provided little control over requirements or schedule.

By the time we started SoftBench 5.0, we had taken important steps to improve our product development process. First, we had a life cycle in place based on user-centered design. We had piloted elements of the user-centered design process with SoftBench 4.0, but the life cycle had not been tested on a large-scale project. Second, we had organized into cross-functional business teams, which helped speed alignment between marketing and R&D by putting a single manager in charge of both functions. And finally, we had just completed the SoftBench 4.0 test phase on schedule, proving that we had the ability to plan and schedule the latter phases of a project.

To make matters more interesting, our new division manager, who had experience reducing cycle time, improving quality, and improving predictability using the Software Engineering Institute's Capability Maturity Model (SEI CMM), challenged us to get to CMM level 3 in 36 months, a process that normally takes two to three years just to go from level-1 to level-2 CMM compliance.

Capability Maturity Model

In 1987 the Software Engineering Institute (SEI), based at Carnegie-Mellon University, published the first version of the Capability Maturity Model (CMM). The initial intent of the CMM was to provide a process maturity framework that would help developers improve their software processes.

CMM describes five levels of software process maturity (Fig.1). At the *initial process level* (level 1) an organization operates without consistent application of formal procedures or project plans. When things get tight, the level-1 organization always reverts to coding and testing. At level 2, the *repeatable level*, controls are established over the way an organization establishes its plans and commitments. Requirements, plans, and procedures are documented, at least at the project level, which means the process could be repeated in the future as long as the type of software being developed doesn't change too much. At the *defined level* (level 3), the organization has documented both its management and engineering processes. This allows the organization to begin to improve the processes over time. Level 4, the *managed level*, is where an organization can quantitatively measure its development and management processes. Finally, at level 5, the *optimizing level*, the development process operates smoothly, and continuous improvement occurs on the defined processes established in the previous levels.

For each level of process maturity, CMM describes the related key practices that characterize that level of process maturity.

Each key process area is defined by a set of one or more goals, as well as the specific practices which, if followed, help achieve the goals. The key process areas and practices are intended to describe what needs to be done to efficiently and predictably develop and maintain software. The CMM does not attempt to specify how software should be developed and managed, leaving that interpretation to each organization, based on its culture and experience.

Project Infrastructure

We chose to move to level 3 by adopting CMM level-2 processes immediately on all new projects. SoftBench 5.0 was the first and largest project to use the new processes and our project infrastructure was designed to support this approach. The key components of our project infrastructure were: a life cycle based on user-centered design, a Web server connected to our configuration management system, and a process consultant and a project lead.

The life cycle had been under development for about a year and we had already used it successfully on some parts of the previous SoftBench release. The life cycle uses a simple waterfall model, augmented with CMM level-2 practices and user-centered design.

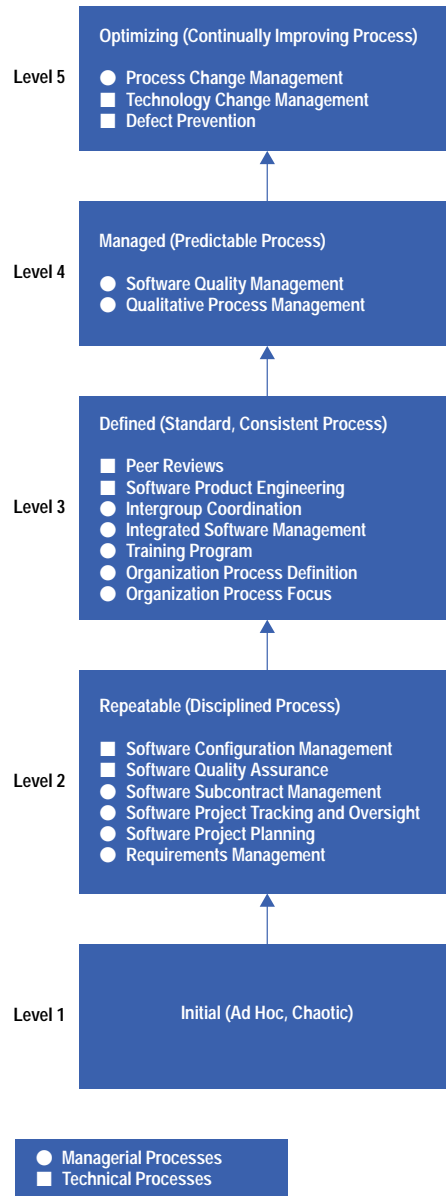


Fig. 1. The five layers of the software Capability Maturity Model. As an organization adopts the practices specified in the model, its software processes should see greater productivity and quality.

CMM level-2 practices ensure that requirements, plans, and schedules are documented, reviewed, and approved by management. Moreover, level-2 practices ensure that as requirements or designs change, the associated plans and schedules are revisited to make sure they are still valid.

User-centered design is based on the premise that a product's success depends on how well the product addresses the needs of the people who use it. User-centered design does this by involving potential users in key development activities, such as profiling user characteristics, characterizing goals and tasks, and validating potential product features and design alternatives.

All of our project documents were checked into SoftBench CM, SoftBench's configuration management system. A Web home page was created for the SoftBench project, allowing us to retrieve documents from SoftBench CM and display them with a Web browser, such as Mosaic or Netscape. The home page included a section for each of the SoftBench teams (to point to customer survey data, requirements, and designs), and sections for product documents, project planning documents, project schedules, and life cycle guidance. We've always checked project documents into our configuration management system, but the addition of the Web browser really improved the visibility and access to these documents. Fig. 2 shows our Web intranet structure.

The third key component of our project infrastructure was the process consultant and project lead. We had a full-time project lead and a full-time process consultant focused on the CMM practices, both as part of the formal management team. We also had a half-time user-centered design consultant from our human factors organization to help us apply the user-centered design techniques. Having these two individuals share accountability for both process and project management proved to be a major success factor.

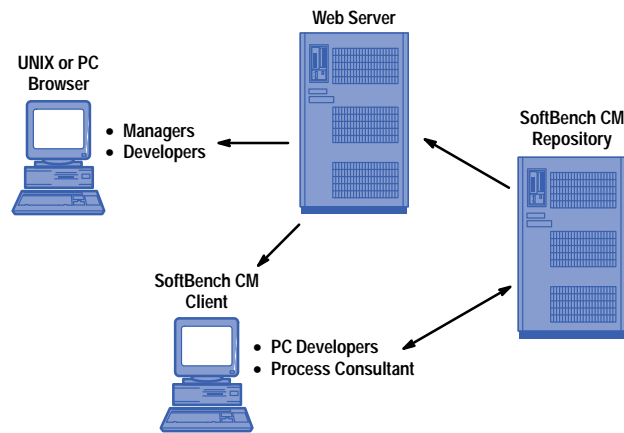


Fig. 2. The network configuration that supported the project infrastructure for the development of SoftBench 5.0.

Reference

1. D. Lowe and G. Cox, "Implementing the Capability Maturity Model for Software Development," *Hewlett-Packard Journal*, Vol. 47, no. 4, August 1996.

Bibliography

1. W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Company, 1989.

Acknowledgments

Many individuals contributed to the success of SoftBench 5.0 and our SEI initiative. We'd like to specifically acknowledge the following for their hard work and perseverance: Jack Cooley, Guy Cox, Doug Lowe, Alan Meyer, and Jan Ryles.

Deborah A. Lienhart
 Project Manager
 Software Engineering Systems Division
 Scott Jordan
 Process Consultant
 Software Engineering Systems Division

- ▶ [Return to Article 1](#)
- ▶ [Go to Next Article](#)
- ▶ [Go to Journal Home Page](#)