5010
Energizing Your Software Development Process with Web Technology

Douglas Lowe and Scott Jordan

Hewlett-Packard
3404 E. Harmony Rd., Ft. Collins CO 80525
Phone: 970-229-6510

The Promise of Software Process Improvement

The practice of software process improvement (SPI) is not new to software. But until recently, process improvement in the software sector has not been able to demonstrate successes that rivaled the proven results in manufacturing. The increasing popularity of applied models for quality management and software process improvement is a testament to the need for improved software effectiveness. Models such as the Software Engineering Institute's Capability Maturity Model (SEI CMM) and the ISO-9001 quality standards are gaining rapid acceptance within the software industry.[1]

Hewlett-Packard is no exception to this trend toward more rigorous application of process standards. One HP division, the Software Engineering Systems Division (SESD), made the commitment to achieve CMM Level 2 maturity in twelve months. The primary driver for this commitment was the need to reduce cycle times. As a result of moving from CMM Level 1 to Level 2 in under 12 months, the division was able to achieve a 50% cycle time reduction across all projects, while at the same time reducing overall costs and improving product quality.[2]

There are many factors that influence the success of process improvement (SPI) efforts, including:

- management commitment
- clear assignment of responsibility for process improvement
- well-respected process improvement staff
- involvement of technical staff
- amount of time and resources dedicated to process improvement
- well understood process improvement goals

These factors and their significance to process improvement success are well documented[3]. All of these factors played a role in our journey to improve software process maturity. This paper, however, will describe an additional success factor for HP's Level 2 CMM SPI project -

The use of World Wide Web (or Intranet) technology coupled with configuration management (CM) technology to facilitate rapid process adoption. [Ref. Figure 1]

This paper will describe how Web and CM technology can be combined to facilitate *and even accelerate* software process adoption, project communication, and software process improvement.

Overcoming the Obstacles to Software Process Improvement

In its study of the factors that influence process improvement success, the Software Engineering Institute reports that there are significant barriers to successful process improvement, including:

- Excessive organizational politics
- Turf guarding
- Discouragement and cynicism from previous experience
- Perception that process improvement gets in the way of the "real" work.

The SEI notes that most of these barriers are under management control. HP's experience would support this claim. HP's experience would also support the assertion that these barriers can be overcome in the following ways:

1. Demonstrating real commitment to process improvement and providing real results can help reduce the cynicism generated by prior process improvement efforts that fell flat.
2. Breaking down the barriers between teams, functional groups, and levels of management can minimize turf guarding. Communication and project visibility play a significant role here.
3. Making sure that the ultimate goal is project success (and not the process in and of itself) can help ensure that the process does not become bureaucratic.

The Web as Process Facilitator

Intranet or Web technology can be an important tool for overcoming some of these barriers. The Web can facilitate process adoption and improvement by:

- Establishing and communicating software process standards
- Making it easy to instantiate software process standards on every project
- Managing and communicating project deliverables and documentation
- Keeping project status and process effectiveness visible.

The Web is essentially a communication tool. Combining the efficiency of Web communication with organizational standards for managing key software processes can minimize the barriers to successful process improvement (and even successful software projects). In short, the Web can be used to make visible those elements of software development that have a tendency to be invisible.

Moreover, when Web technology is linked to configuration management tools, the combination provides an easy to use process and project communication cockpit that ensures that:

- changes to the project and the process are tracked, controlled, and communicated
- current versions are in use at all times, a key requirement of CMM and ISO-9000.

In SESD, our configuration management system is used extensively in conjunction with the web linkage to provide transparent access to the latest document version. Refer to Figure 2. These documents can be either process specifications or project related specifications or plans.

Communicating Software Process Standards

Many organizations have tried to develop complete, robust software process standards only to run into the insurmountable hurdle of getting project teams to actually use the process standards. One reason this hurdle is so difficult to overcome is that the documented process standards become unwieldy and out of date, especially when they are documented in large printed notebooks that are difficult to locate and manage.

One of the proven benefits of the Worldwide Web and Intranets is the ability to put policy and procedure documentation on-line, making the documentation easy to find and easy to keep up-to-date. Software process documentation is no exception. With the combination of Web technology and configuration management technology, organizations can:

- put software process standards on-line,
- version the software process standards, and
- take advantage of hypertext markup language (HTML) capabilities to make process standards easier to understand and apply.

Putting software process standards on-line usually begins with process architecture. A basic software process architecture should include the following elements, shown in Table 1.

Table 1

Life Cycle Element	Addresses:
Policies	**Responsibilities**
Procedures	**Activities**
Templates	**Work Products**
Examples	
Training	**Capabilities**
Checklists	**Verification**

This process architecture can easily be integrated into the Software Life Cycle in order to simplify communication of what is expected during project execution. Once this architecture is defined, the software life cycle can be documented, versioned, and presented on-line via the Web to the entire development staff. The life cycle can even be documented in stages, for example in Table 2, starting with the highest level view. Over time more architectural components, like procedures and training, can be added using hypertext links.

Table 2

Software Life Cycle - Requirements Phase				
DELIVERABLE	PROCEDURE	OUTLINE	EXAMPLES	TRAINING
Project Requirements Summary	X	X	X	X
Software Requirements Specification	X	X	X	X
Software Development Plan	X	X	X	X
Software Configuration Management Plan	X	X	X	X
Software Quality Assurance Plan	X	X	X	X
Requirements Complete Report	X	X	X	X
Approval Checklist	X	X	X	X

Presenting versioned software process standards via the Web provides several benefits:

- The hypertext links make it easier for managers and developers to locate and use the software process standards. As a result, *software process standards are applied more consistently across an organization*.

- Because managers and developers are accessing the standards from an on-line repository instead of an out-of-date binder, *they are more likely to use the current standards*.
- Because the life cycle is actually being accessed from a CM archive, *changes to the software process are controlled*.

Applying Process Standards to a Project

Once the software process architecture is in place and on the Web, it's a simple task to translate the organization's software process into a project-specific process description, as shown in Table 3. This involves:

- creating an on-line project notebook (also called a project home page)
- populating the project notebook with project plans and work products, such as requirements and design documents.

The intent is to make the project home page the single point of navigation to all information relevant to the project. HTML links are used to link elements on the home page to the actual project artifacts, such as the project plan. The project artifacts are actually stored in the CM archive. By connecting the CM archive to a Web server, the HTML links are able to directly access artifacts in the CM system, including any version of the artifact. This ensures that documents are versioned with audit trails and change control capabilities.

As the single point of navigation the project home page should point to all project-specific and organization-wide policies and procedures. The Web home page allows for quick and easy customization of procedures, because the project team can simply copy the organizational procedure, customize it to suit the project objectives, and then link the customized version into the project home page. This makes the process immediately visible to the project team and it leaves a record of the process to facilitate repeatability in the future.

Managing Project Deliverables and Documentation

The project home page also serves as the single point of navigation for all projects work products. The project home page simplifies distribution of preliminary and final work products - a simple e-mail can alert reviewers and consumers of the availability of a work product on the Web page. From the developers' perspective, the Web provides a simple mechanism for communicating to reviewers, managers, and even customers. Often the developers will actually record and document more information than they have in the past, because the Web provides a means to advertise their contribution. From a manager's perspective, the visibility of the work products on the home page makes it easier to determine what's done and what still needs work. And quality assurance now has a non-intrusive mechanism for reviewing work products for completeness and consistency with process standards.

Because the work products are stored in a CM archive, they are not only versioned, but changes to the documents can be controlled. Ideally, the link between the Web server and the CM archive will also allow developers and managers to access any version of a document (including tagged versions) and to view the change log.

When all these elements are put together - project deliverables, procedures, configuration management - the result can look like this:

Table 3

Project Notebook - Test Phase		
Dashboard	**Deliverables - Test Phase**	**Process Guidance**
Build Status 3 Failures	Integration Test Cases System Test Cases Defect Reports Test Results Defect Reports Product Package - Object Code - Tested Software - Installation Procedures & Doc - User Documentation - User Training - Transition or Migration Doc - Packaged Software Release Readiness Report	Software Development Plan SCM Plan Integration Test Plan System Test Plan Build System Guidelines Test System Guidelines Defect Tracking Guidelines
Test Status 98% Passed		
Defect Status New Fixed Open 3 5 8		
Other Project Info		
Schedule Status Pert Chart Schedule **Project Archives** Project Status Reports Project Issues List SCM Activity Reports SQA Activity Reports		

The upper left corner of the project notebook demonstrates another useful feature of the home page - the Project Dashboard. The dashboard centralizes dynamic status information for the project, the kind of the information that managers and developers can use on a daily basis. In the example above, the dashboard includes nightly build results, test results, and defect metrics. These dynamic links can be set up to connect to Web pages with more detailed information, like the detailed build errors and links to the defect tracking system. The dashboard puts dynamic - even real-time - project status information at the managers' fingertips, improving their ability to track quality and schedule.

Tracking Projects - The View from the Top

The Web even provides value for senior management. With a few simple Web pages, senior management can keep informed about:

- which projects are active,
- what the projects are trying to accomplish, and
- where they are in the life cycle.

Table 4

Project Index			
PROJECTS	**DATA SHEET**	**STATUS**	**PROJECT PLAN**
SoftBench 5.1	Data Sheet	August	Test
SoftBenchCM 1.9	Data Sheet	August	Design
CodeAdvisor 2.0	Data Sheet	August	Requirements

The project index shown above, in Table 4, provides links to key project information for all active projects, including:

- on-line notebooks

- one page data sheets
- latest status reports
- Current project plans and schedules.

Using simple Web index pages like these helps increase the visibility of projects, and allows the management team to quickly access important information about all the organization's projects, without having to find the project manager and ask for paper copies of project deliverables.

Another key indicator of project progress is project milestones. The Web can be a simple way to ensure planned project milestones are visible, so that everyone in the organization knows when project phases are expected to complete. An example of this is shown in Table 5.

Table 5

Project Milestones		
	Milestones	**Date**
SoftBench 5.1	Requirements Complete	Done - February 21
	Design Complete	Done - May 10
	Code Complete	Done - August 10
	Testing Complete	November 28
	Shipment Release	February 19
SoftBenchCM 1.9	Requirements Complete	Done - October 4
	Design Complete	January 22
	Code Complete	April 30
	Release to Mfg	July 1
CodeAdvisor 2.0	Requirements Complete	December 8
	Design Complete	February 4
	Code Complete	April 12
	Release to Mfg	June 21

Accelerating the Projects' Results

Since a key focus for our improvement efforts was cycle-time reduction, we benefited from using WebNotebook technology by taking advantage of increased efficiencies in communication, organization, reuse and training. Refer to Figure 3.

Using the combination of the Web browser, configuration management and process standards provided an initial jump-start for project teams that were beginning a new project. The project team could easily identify what deliverables would be needed, and the sequencing of these. They could find examples of these deliverables from previous projects, and they had a good starting point for training the team on what processes would be used for a few of the key process areas, such as project planning, requirements management, configuration management and quality assurance.

For projects that are in their second time through, the benefits are even greater. Once a base of project specific documents and processes are established and safely stored in the CM repository, the efficiencies come from reuse of portions of prior projects' schedules, plans, deliverables and processes, and from the reduced time needed for training new members of the project team. Prior experience and

documentation on the project are excellent sources of informal training for new team members.

Improvement in results of the project teams also came from increased visibility into the progress on the project, and taking earlier corrective actions to keep the project on track. WebNotebook technology provided much more detailed insight into the project's progress between milestones, than we had using a paper based or face-to-face meeting based tracking system.

Conclusion

Back at SESD, WebNotebook incorporated the Web and CM techniques described above, and helped the organization reach CMM Level 2 ahead of plan. Ten months after the start of the improvement project all of the division's product teams had been internally assessed to be operating at Level 2, based on quarterly reviews and in-depth audits of each team.

Even more important, the division met the business goals that were linked to the process improvement project. Table 6 below shows a dramatic improvement in the product teams' performance compared to the average for the previous year[4].

Table 6

Objectives	1994 Average	1995 Average	1996 Average
Reduce Product Cycle Time	22.5 months	10.0 months	6.5 months
Reduce Project Slip (Delay)	9.5 months	0.7 months	0.0 months

Figure 4 indicates cycle-time trend and project slip trend information for SESD over a longer time period. The reduced cycle time and project delay translated into direct cost savings for the division. The CMM and Web improvements in 1995 reduced overall operating costs by an estimated two million dollars, and required an investment of $250K.

References

[1] Paulk, Mark C.; Curtis Bill; Chrissis, Mary Beth; & Weber Charles V. "Capability Maturity Model for Software, Version 1.1", (CMU/SEI-93-TR-24, ADA263403) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

[2] Lowe, Douglas E.; & Cox, Guy "The Fast Track to Implementing the Software Engineering Institute's Capability Model Level 2", *Hewlett-Packard Journal*, Vol. 47, no. 4., August 1996, pp. 1-14.

[3] Goldenson, Dennis R.: & Herbsleb, James D. "After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success", (CMU/SEI-95-TR-009) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995.

[4] Lowe, Douglas E. "A Rapid Design and Deployment Method for CMM Level 2", *Proceedings of the SEPG'96 Conference*, SEI Software Engineering Process Group Conference, Atlantic City NJ, May 1996.

Capability Maturity Model(sm.) and CMM(sm.) are service marks of Carnegie Mellon University.
© Copyright 1996 Hewlett-Packard Company. All rights reserved.

Figures 1 - 4 are available in paper copy only, or at the presentation session.

If you wish to have a copy of the presentation materials, you can contact the author, Doug Lowe, by e-mail : lowe@fc.hp.com for a paper copy or electronic Powerpoint version.