THE PROCESS GROUP

POST

Vol. 7, No. 1

HELPING YOU IMPROVE YOUR SOFTWARE ENGINEERING PROCESS

March 2000

Editor's corner

Does e-commerce software need engineering?

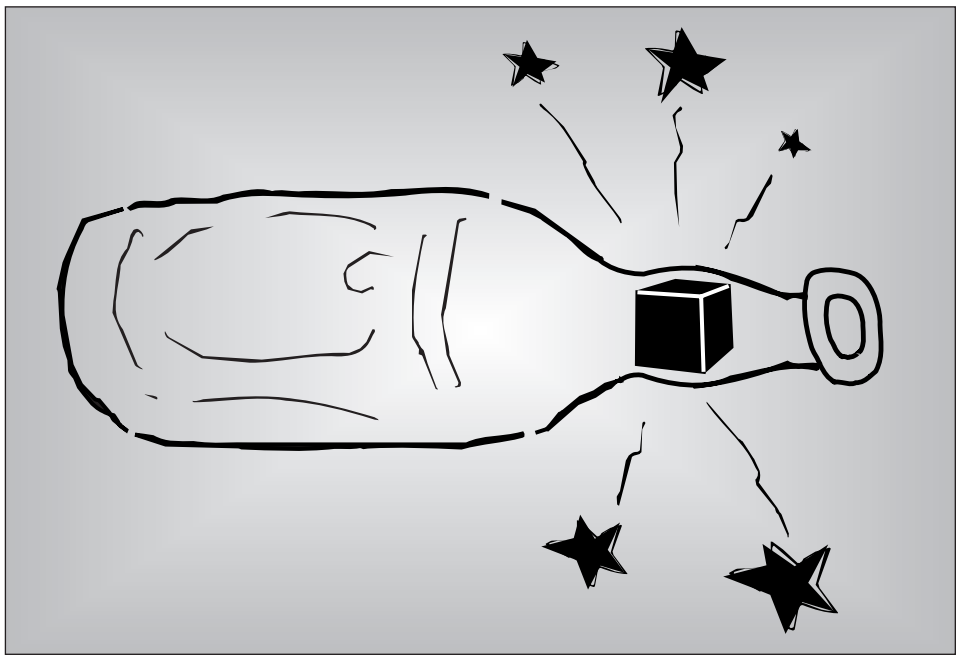
I have noticed a temptation to treat e-commerce software as radically different from other software products. This treatment can encourage development teams to discard techniques that have worked in the past or waste too much time creating new techniques.

E-commerce software certainly has some unique requirements, such as performance with high traffic, reliability during lengthy transactions, accuracy of links to sister components and the necessity for data security. E-commerce software also has many similarities. It needs to be built and shipped quickly, work as advertised, not cost a fortune to develop, and be usable on different machines.

E-software can be created painfully using a code-and-ship life cycle. It can also be achieved with less pain, cost and risk by employing mini-management and mini-engineering techniques. These consist of well-tested concepts used on a scale in proportion to the project's size. For example, using small-scale versions of goal clarification, design, test, risk identification and early bug detection.

One large company we helped needed to provide a web browser and e-mail service to 3,000-plus PCs across its various divisions. The part-time team was given only a few weeks to complete the project. The project involved a complex design of servers, encrypters and web applications. The team could have taken a code-and-ship approach, run fast and hoped for the best. Instead, we coached them through approximately six hours

(Continued on page 3)



Bottlenecks are a common problem.

Meeting tight schedules through cycle time reduction

by Dr. Dennis J. Frailey

Senior Fellow, Raytheon; Adjunct Professor, SMU

How do you meet a tight schedule? Skimp on quality? Add more staff? Have people work overtime? These are the classic, short-term solutions to schedule problems. They sometimes work -- but they do nothing to head off the next schedule crisis, and often make it worse. None of these addresses the real causes of schedule problems. Why does it take longer than it should to develop software? Experts in the field of cycle time say it all boils down to three fundamental problems: variability, overly-complex processes, and bottlenecks and constraints, often imposed by the organization on itself.

Short cycle time gives you a competitive edge. Studies by McKinsey and Co. show that being on time but 50% over budget loses about 3% of total potential profit over the product's lifetime. But meeting budget while being six months late loses 32% of total profit potential. Why? Because during that six months your competitor has gained an edge that you can seldom recover from. In environments like software development, where product lifetimes are short, your product may never make it out the door.

Cycle time problems are easy to spot. Just look for long queues and rework in your process. Is there a wait for access to test equipment? Do the software developers often misunderstand the system engineers or customers? Does management approval take too long? Does it take a lot

(Continued on page 4)

Software delivery problems? We have answers now.

- ❏ **Decrease product development cycle time.**
  In this three-part workshop, CYCLE TIME REDUCTION FOR SMALL SOFTWARE PROJECTS, project managers and their teams learn how to accelerate delivery through specialized schedule optimization techniques.

- ❏ **Launch projects effectively. Meet project deadlines and reduce risks.**
  In this three-day SOFTWARE PROJECT PLANNING AND MANAGEMENT workshop, project managers and their teams learn how to meet deadlines through better estimation, reduce surprises using risk management, schedule work for better optimization, understand and negotiate project trade-offs, and track progress.

- ❏ **Meet project deadlines. Scope and estimate the project work.**
  This one-day SOFTWARE ESTIMATION workshop (a subset of Software Project Planning and Management) helps teams develop more accurate estimates.

- ❏ **Avoid schedule delays caused by needless product rework. Find defects rapidly.**
  This two-day INSPECTION (PEER REVIEWS) workshop teaches teams to efficiently find defects in code and documentation. (Includes moderator skills.)

- ❏ **Hands-on SEI CMM/CMMI. Perform a mini-CMM gap-analysis.**

  The following workshops are available:

  - ❏ SEI LEVEL 2 (one day), SEI LEVEL 3 (two days), SEI LEVEL 4 (one day).
  - ❏ SEI CMMI—Overview of CMMI-v0.2 (One half-day presentation).

- ❏ **Identify critical changes to improve organizational results. Benchmark against the CMM.**
  A SOFTWARE PROCESS ASSESSMENT examines your organization's software practices and generates a focused list of the critical areas for improvement. Our SEI authorized Lead Assessors conduct customized CMM-based appraisals.

- ❏ **Goal/problem-based improvement.**
  This two-day SOFTWARE ENGINEERING PROCESS IMPROVEMENT workshop provides a systematic approach for organizations to improve their development capability. It includes: getting management support, focusing the organization on the critical issues, planning the improvement and effecting change.

- ❏ **Tailored assistance. Dedicated phone-based assistance.**
  This service consists of customized education and coaching on your specific problems (e.g., meeting deadlines, quality and cultural change.)

- ❏ **Audio cassettes:**
  * The Role and Focus of a Software Engineering Process Group (SEPG)
  * Making Change Happen—a 10-Piece Tool Box

*Detailed information on our services is available at **www.processgroup.com.***

All services are tailorable in content and delivery. They are available on-site and include unlimited telephone/email support.

*Contact us at **972-418-9541** or **help@processgroup.com** to discuss your needs.*

Please send us your e-mail address to receive future issues.

Editor's corner

(Continued from page 1)

(3 percent of the project duration) of planning, risk mitigation and schedule optimization to determine how their limited resources could meet the deadline. As a result, the team achieved an on-time and on-budget deployment.

Whether you are building e-commerce or not, you have delivery goals to meet.

I recently attended a workshop for SEI CMM Level 4 and 5 software organizations. They presented their techniques for optimizing schedule, quality and cost. At lunch, I asked one of the e-commerce representatives from India how his company could compete with code-and-ship companies in other parts of the world. He looked confused and then replied that all of his competitors in India were also SEI CMM Level 5. He added that with their capability they could guarantee worldwide customers a completely reliable solution at lower cost. Many of his competitors in the USA could not guarantee anything, hence their competitive advantage.

Whether you are building e-commerce products or not, you have delivery goals to meet. Small-scale, well-tested concepts such as regression testing, architecture design, project estimation, risk analysis, requirements management and inspection still provide very effective techniques to ensure your organization's success.

Neil Potter

The Process Group

Mailing address: The Process Group
P.O. Box 700012
Dallas, TX 75370

Telephone number: 972-418-9541

Fax number: 972-618-6283

E-mail: help@processgroup.com
Web: www.processgroup.com

POST back issues are on line

Checking improvement

By Neil Potter and Mary Sakry

When an improvement program is in full swing, it is necessary to monitor progress. A mini-assessment obtains a quick snapshot of the improvement program. The results show which practices are being adopted and which are not. A mini-assessment is not an audit or a full process assessment, but a friendly check to determine progress.

Before a mini-assessment is conducted, it is important to decide what practices will be checked for adoption. This could include the activities described by the organization's development life cycle, SEI CMM, ISO9001 or Bellcore model. A list of questions is based on these criteria.

Sample mini-assessment questions

Does your team:

- Perform inspections or walk-throughs for critical work products (e.g. code, design, test cases, and plans)?
- Perform black-box testing?
- Perform configuration management (CM) of all critical work products (from plans to code)?
- Have adequate cross-site development version control?
- Have adequate computer network stability?
- Use a process for estimation?
- Use a process for risk management?

Mini-assessment process

There are five steps to the mini-assessment process.

1. Meet with the developers and managers to explain what will be checked and how.
2. Perform the mini-assessment (verbal interview with questionnaire).
3. Publish the results (organization summary).
4. Debrief with the participants.
5. Improve the questionnaire (add examples and remove ambiguity).

- **Meet with the developers and managers to explain what will be checked and how**

Start with a one-hour meeting of all the participants and describe the process. Involving the developers and managers throughout the mini-assessment process will help ensure its effectiveness and create buy-in.

- **Perform the mini-assessment (verbal interview with questionnaire)**

The mini-assessment team interviews each development team using the defined questions. A verbal interview is used to ensure the quality of the responses. For example, if the project team answered yes to, "Do you perform configuration management of all critical work products?," then you can ask clarifying questions about CM practices.

The interview typically takes one or two hours, depending on the number of questions asked.

- **Publish the results (organization summary)**

The results of a mini-assessment can be published graphically using a bar chart (see Figure 1) to indicate the percent of "yes" responses. This chart shows the trend of the group. If the organization is very large, you may decide to publish the results for each major division.

Keep the data confidential

The graph indicates that progress was made between January '99 and May '99. Ground was lost in September. The cause of this decline was the attrition of personnel.

Keeping the data confidential

We suggest that the assessment team keep all individual project data in confidence. Each team can obtain data about its own score for the purpose of its own improvement. The focus of management should be on the published, organization-wide trends.

- **Debrief with the participants**

The mini-assessment can cause resentment among the interviewees if it is not managed carefully. The developers may perceive it to be an audit, regardless of what happens. To make the process effective, the mini-assessment team needs to obtain involvement from the organization before and after the mini-assessment. A

(Continued on page 4)

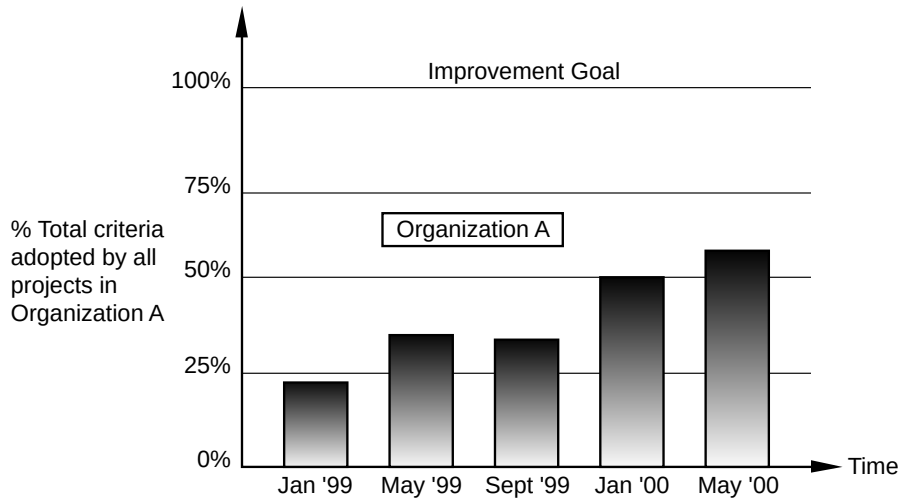


Figure 1

Meeting tight schedules

(Continued from page 1)

of paperwork to purchase needed development tools? Must data be converted to a different format? Work waiting to be done is called WIP or Work In Process. The more you have, the longer your cycle time. What causes excess WIP? The three culprits mentioned in the first paragraph: variability, complexity, and barriers/bottlenecks.

How do we improve cycle time? By attacking the three fundamental problems. Sometimes the proper actions are "counterintuitive". For example, the cycles of learning principle says it's faster to do the job three times, in small increments, than to do it all at once. The small batch principle says that "economy of scale" doesn't always work. Smooth flow, another cycle time principle, says the optimal process is one where each step flows at the same speed, as the cars on a train, rather than having each step go as fast as it can, as the cars on a highway. What does this mean in practice? Instead of encouraging everyone to go as fast as they can, you need to look at the overall system, find its bottlenecks, and focus resources and ingenuity to optimize performance at the bottlenecks. For example, it may be more sensible to have a programmer help out in the testing process than to have him/her write programs faster. Overly-complex processes and bottlenecks often result from precautionary measures that protect against unlikely or inexpensive problems. One company saved a lot of time and money by eliminating the requirement for travel authorizations. The reduction in bureaucracy saved millions each year—much more than the slightly increased cost of inappropriate travel. After all, how many employees will risk their jobs just to take an unjustified trip?

You can shorten schedules permanently by attacking the root causes of cycle time problems. And when you deal with causes instead of symptoms, you save money and improve product quality. This is the wonderful secret of cycle time reduction —you win on all counts! The techniques are not hard—you just have to apply basic principles in a methodical fashion and be open to new ways of doing your work. And if you aren't sure whether to try it, remember your competitors -- they're out there working to be faster than you.

The Process Group has collaborated with Dennis J. Frailey on a new program called *Cycle Time Reduction for Small Software Projects*. Please call or visit our web site for details.

Checking improvements

(Continued from page 3)

debrief with the participants, after the results have been published, identifies what aspects of the mini-assessment process were effective and what aspects need to be improved. This debrief typically takes one hour, utilizing the following agenda:

1. Brainstorm what the organization liked about the mini-assessment process.
2. Brainstorm areas for improvement of the mini-assessment process.
3. Remove invalid or irrelevant items from the brainstorm list.
4. Set priorities for areas to improve.
5. Assign responsibilities for actions.

• **Improve the questionnaire (add examples and remove ambiguity)**

In addition to a debrief, it is also beneficial to invite between three and ten interviewees to help review the criteria used in the mini-assessment. This review can help remove ambiguities in the questionnaire and elicit examples to improve clarity. The questionnaire can be improved by adding a column to help describe the use or intent of the practices being advocated. An example is shown in figure 2.

Mini-assessment questions	Use or intent of the practices
Perform inspections or walk-throughs for critical work products (e.g, code, design, test cases, and plans)?	To find defects systematically. Gilb, Fagan or Weinberg processes are acceptable.
Perform black-box testing?	The software is tested against the requirements specification and user guide.
Perform configuration management of all critical work products (from plans to code)?	Version control is conducted for all critical outputs created when using the standard development life cycle (e.g., requirements, design, code, test cases, user guide). Teams can use a tool or manual version numbering.

Figure 2.

Summary

This mini-assessment process is an effective way to understand which practices have been adopted and which have not. The information is used to understand current gaps, obtain insight on problems with deployment, and provide a basis for replanning the improvement effort.