

## MAKING PROCESS IMPROVEMENT WORK

### *A Concise Action Guide for Software Managers and Practitioners*

After many (thousands of) hours, we have completed our book. It is being published by Addison-Wesley and will be available at the end of March. (See our home page, [www.processgroup.com](http://www.processgroup.com), for book availability.)

The book describes the critical steps needed to plan and implement improvement within a software development organization. The book will stimulate your thinking about:

- How software development organizations improve
- What they improve
- How they deploy and track improvements

In this newsletter we share with you an excerpt of the book describing some lessons learned from an improvement program.

## WHAT LESSONS HAVE WE LEARNED SO FAR?

If you want to learn how your improvement program is going, talk to the people who are being asked to change their behaviors and adopt new practices. This might include managers, developers, SQA personnel, and testers.

Lessons learned data comes from interviewing individuals or using discussion groups. You can conduct a lessons learned session at any time; however, three specific times are particularly useful: when a goal has been reached, when an intermediate goal has been reached, and when the improvement effort hits an obstacle.

You can use the agenda in Figure 1 to determine lessons learned and related corrective actions. As a rule-of-thumb, break the session into segments of two hours or less to avoid team fatigue. When using group interviews, construct the groups to encourage uninhibited discussion. Invite people who are willing to be frank and candid. Select a good objective facilitator, someone not in charge of the improvement effort.

*(Continued on page 2)*

## LESSONS LEARNED AGENDA

1. Clarify the scope of the session.
2. Determine strengths (what went well).
3. Determine areas for improvement.
4. Set priorities.
5. Determine corrective actions.

Figure 1

To stimulate the discussion, consider the following questions:

- Were the improvement activities tied to the business goals and problems experienced by the organization?
- Was there enough effort invested to adopt the new or improved techniques?
- Were new practices tailored appropriately to the needs of each project? Were pilot projects used to ensure their appropriateness?
- Is there evidence that the results of the organization are improving? (This can include anecdotal stories as well as metrics.)

Here is an example of this process for a complete execution of the improvement cycle. We have explained the lessons in more detail here than would be typical during a brainstorming session involving people familiar with the issues. The data in this example comes from one division of a large multinational company that produces software for digital image processing equipment.

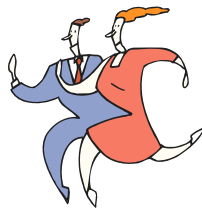
### 1. Clarify the scope of the session

Lessons learned from one complete execution of the improvement cycle.

### 2. Determine strengths (what went well)

*Lesson 1: Decentralizing the improvement action plan gives each project team ownership over its plan.*

The centralized improvement action plan for the division was split into three separate plans. When each plan addressed a single product line's specific goals and problems, the project teams became excited about the improvement program.



*Lesson 2: Break the action plan into small chunks and just start.*

Starting with an overwhelming list of project problems, we decided to focus on the immediate issues that were impacting projects A and B. This included auditing the release process, inspecting critical code, and consolidating our two defect tracking

***When each plan addressed a single product line's specific goals and problems, the project teams became excited about the improvement program.***

systems used by the developers and the help desk. Starting on a small scale gave our improvement program momentum, something it had never had before.

*Lesson 3: Don't preach when an example can say everything for you.*

The improvement team had been pushing risk management for six months with

***Do not struggle for too long.***

little impact. When the project manager of our smallest project described at a division meeting how risk management techniques were used to turn his project around from near disaster, other project teams started using the process.

*Lesson 4: Do not struggle for too long; get some help.*

We knew we wanted to work on project planning, but the teams would either not make the time for planning, or they would plan to the point of overkill. If they planned too much, they threw up their hands and said that planning was not practical. We brought in an outside coach to help us plan two important releases and get back on the right track. The teams now understand what level of planning is appropriate.



*Lesson 5: Guide people in applying each new technique to their work.*

People have so much going on they do not know where to start. When we deployed the inspection technique, people resisted because they said they had too much code to inspect. We developed guidelines to help them select appropriate sections of code and documentation for inspection. Each team picked 20 percent of its new work that matched at least one of the following criteria:

- The most critical to the program's operation
- The most used section in the product
- The most costly if defects were to exist
- The most error-prone section
- The least well-known section
- The most frequently changed section

(Continued on page 3)

## WHAT LESSONS HAVE WE LEARNED SO FAR? *(Continued from page 2)*

Similarly, when we deployed the estimation technique, people resisted because they said they had too much work to estimate. We developed

### *Focus all new techniques on the needs of the project.*

guidelines to help them select appropriate sections of their project for the estimation process. Each team selected project components that matched at least one of the following criteria:

- The component is high risk (either difficult technically, critical to the success of the project, or full of unknowns).
- There are no historical estimate data on which to base our estimates.
- There are many varying opinions about how much work the component will take and the development approach that should be used.

### *Lesson 6: Focus all new techniques on the needs of the project.*

The techniques we deployed initially did not stick. The project team members did not see how the techniques would help them with their current issues. To address this problem, we started every improvement deployment session with a quick review of the project's top three problems and top three goals. During the deployment session, we allowed the team members to question where and when the new technique would help them on their "top six list." The project members could always see how the new technique applied to their current work.

### *Lesson 7: Obtain management buy-in for the plan before execution.*

The improvement team developed an action plan with some participation from the developers. Nevertheless, execution of the plan during the implementation phase was hopeless. No one had time to

implement the plan, and the management team treated the improvement effort as a task to avoid at all costs. After six months, the improvement team was ready to give up. Finally, one improvement team member asked the management team members what they would like to see changed in the improvement program. The clear message was that they felt that the improvements were being forced on them. In hindsight, the improvement team was in a rush to have the processes adopted and did not take the time to allow the managers to critique the plan or the proposed solutions before deployment.

### *Lesson 8: Keep measuring defect density and end user customer satisfaction.*

We measured product defect density (the number of defects found by the users for each release) and end user customer satisfaction. These numbers are now shared with our customers. Communicating this information has built an excellent trust level with our customer base. We have enduring metrics that guide our improvement program.

## 3. Determine areas for improvement

### *Lesson 9: The process-centric approach to improvement was very difficult to sell.*

The improvement plans were initially organized around the key process areas of the CMM, and the process improvement team spent most of its time generating excessive documentation. The engineers and managers avoided the improvement effort at all costs.

### *Lesson 10: Using the same communication technique as everyone else allows the message to be lost.*

Each developer was receiving 200 to 300 e-mail messages each week related to his regular project work. Getting our improvement message out using e-mail to the project teams was impossible.

### *Lesson 11: Allowing private data to become public sets perilous expectations.*

We started to measure how many projects satisfied the CMM Level 2 activities using the mini-assessment process. One time we released project-specific data to some senior managers because we were trying to be helpful to the management

### *Using the same communication technique as everyone else allows the message to be lost.*

team. That set the precedent for future mini-assessments. Projects became focused on the "score." We are now truly in a score-focused mess.

### *Lesson 12: Be careful of what information you ask for!*

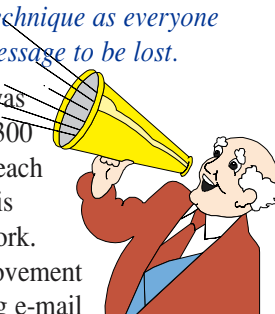
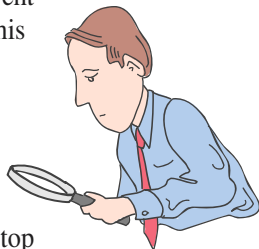
We wanted to encourage people to put their sample process documents into the process assets library (PAL), so we measured the percentage of projects that submitted documents to the PAL. This measurement caused projects to submit everything they had to the PAL to earn extra credit. Now the measurement is meaningless and the PAL is completely full with who knows what.

### *Lesson 13: Using a scoring system for process adoption can encourage inappropriate behavior.*

We measured how many inspections (peer reviews) each team performed annually. To maintain our ISO9001 registration, we established a minimum requirement for each team to conduct two inspections annually. Now many teams refuse to do more than two each year. The benefit of finding defects has been replaced by the requirement to perform the minimum number of inspections.

**...This is all the space we had in this newsletter! Please see the book for the remainder of the article.**

Potter, N., Sakry, M. *Making Process Improvement Work – A Concise Action Guide for Software Managers and Practitioners*, Addison-Wesley, 2002. ISBN: 0-201-77577-8.



# Practical Solutions for your Software Development Challenges

## ❑ Understand customer needs. Clarify product requirements early.

In this workshop, IN SEARCH OF EXCELLENT REQUIREMENTS, software engineers, managers, requirements analysts and user representatives learn how to gather, document, analyze and manage customer requirements for software applications.

## ❑ Decrease product development time-to-market.

In this workshop, ACCELERATING PRODUCT DEVELOPMENT FOR SMALL SOFTWARE PROJECTS THROUGH CYCLE TIME REDUCTION, project managers and their teams learn how to accelerate delivery through specialized schedule optimization techniques.

## ❑ Manage 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-v1.1 (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](http://www.processgroup.com).

Contact us at 972-418-9541 or [help@processgroup.com](mailto:help@processgroup.com) to discuss your needs.

## What is in the book?

Here is the book's Table of Contents:

Foreword by Karl Wieggers.

Preface.

Acknowledgements.

Chapter 1. Developing a Plan.

- Scope the Improvement.
- Develop an Action Plan.
- Determine Risks and Plan to Mitigate.
- Chapter Summary.

Chapter 2. Implementing the Plan.

- Sell Solutions Based on Need.
- Work with the Willing and Needy First.
- Keep Focused on the Goals and Problems.
- Align the Behaviors of Managers and Practitioners.
- Chapter Summary.

Chapter 3. Checking Progress.

- Are We Making Progress on the Goals?
- Are We Making Progress on our Improvement Plan?
- Are We Making Progress on the Improvement Framework?
- What Lessons Have We Learned So Far?
- Chapter Summary.

Conclusion.

Appendices.

References.

## 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](mailto:help@processgroup.com)

Web: [www.processgroup.com](http://www.processgroup.com)

POST back issues are on line