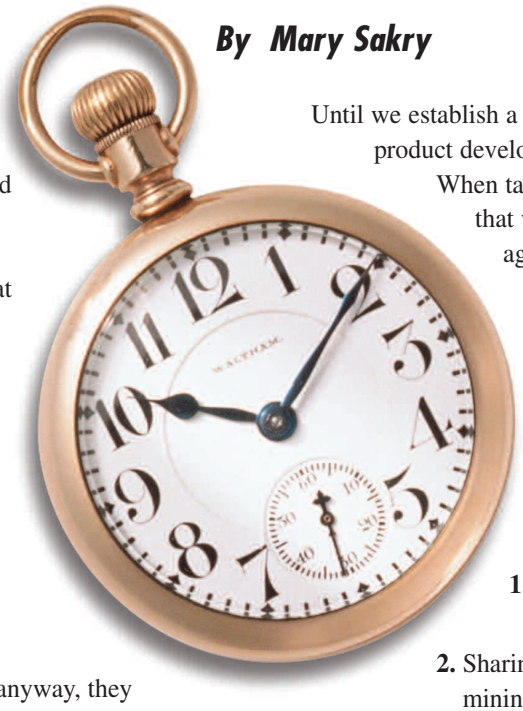
THE PROCESS GROUP

POST

Vol. 11, No. 2 HELPING YOU IMPROVE YOUR SOFTWARE ENGINEERING PROCESS www.processgroup.com/newsletter.htm October 2004

"DEALING WITH THE CUSTOMER IS TOO DIFFICULT AND TAKES TOO MUCH TIME"

By Mary Sakry



Software developers often believe that they need to produce their product in a hurry without consulting customers or end users. This means that they are deciding what to create and what features will be included. The result is often a product that will not function as the users expect and might take more releases than needed to meet the users' needs.

Working in the dark happens for many reasons – most of them tied to beliefs about what would happen if we did actually talk to customers. Other reasons relate to perceived or actual difficulties encountered in trying to understand customers. These beliefs include:

Belief 1: We can't talk to the customers; anyway, they would just demand more stuff

Belief 2: The customers won't accept less functionality in the release

Belief 1: We can't talk to the customers; anyway, they would just demand more stuff

Pleasing users by guessing what they want is a risky approach to developing products. If we are good at guessing, then we will predict exactly what should go into each product and our customers will buy all our products! Sometimes developers get lucky and do just that, but most of the time, they will produce something that misses the mark. Sometimes the functions don't do quite what the users expect; sometimes they are overwhelmed with features they don't even need or, worse, the functionality just plain doesn't work.

Developers can be fearful of contacting users because they think they will demand even more functionality and never be satisfied. Since release schedules are usually already tight, if not impossible to meet, they reason that establishing a rapport with customers will take additional time they do not have.

Until we establish a good dialogue between developers and users, product development can be little more than guesswork. When talking with customers, we might determine that we are doing too much and that we could agree on priorities and reduce the volume of work. We could plan staged releases that add functionality over time or realize that some of the features are overly complex.

You can begin the dialogue with users by holding customer focus meetings. In its simplest form this gathering can consist of discussions on how the new release or product will be used. It could include:

1. Planned questions about current functionality and how users currently do their jobs.
2. Sharing ideas of what a product could do and determining if these would be features they would use.
3. Discussions of ideas using text, pictures, use case narratives, context diagrams, prototypes, and menu-transition diagrams (dialogue maps).

Once developers start talking to real users, they learn all sorts of things about how the customers do their work and what they really need. Users can help clarify needed functionality, possible solutions, business rules, the needed sequence of user tasks, and their frustrations and constraints.

Belief 2: The customers won't accept less functionality in the release

Developers have been known to please users by stuffing as many features as possible in a release. *More* functionality is perceived to be better than *needed* functionality. This misconception can happen on both sides. All of us can make the mistake of adding features to create a "rich feature set," rather than sit down together and decide what really needs to be there this time, and what can wait for another release.

Continued on page 3

MANAGING REQUIREMENTS

By Neil Potter

The problem

Requirements management is a well-established series of steps for tracking and evaluating change requests. In 40% of the companies we have observed, however, managing requirements changes is perceived as luxurious and theoretical – something only performed by people in snail-paced industries with plenty of time on their hands!

This article describes the common intuitive thinking that prevents teams from implementing effective requirements management, and some of the variations in implementation to make requirements management effective.

Intuitive thinking

There are two common beliefs that cause people not to establish an effective requirements management system. First, the project members believe that there is no time to evaluate requirements changes; it is believed to be quicker just to make the change. Second, the impact of the change to other groups and other parts of the system is assumed to be negligible. These intuitive thoughts are risky and lead to unexpected schedule delays, poor communication among impacted areas and quality problems that can be time consuming and difficult to fix.

The lack of requirements management can cause the requirements change problem to snowball. For example, if you respond to a request by saying, "No problem – we will have that change request complete in the first release," you might be interpreted as implying that all change requests can be accomplished with no impact. The requester can choose to hear this implicit message loud and clear and dream up other changes that would be "cool to have."

Premature agreement to change requests can appear to be intuitively correct because it is "customer focused." However, it can leave your team buried in unachievable commitments and risk the customer not achieving his or her goals.

The solution

Your requirements management solution does not have to be excessively formal, documentation-centric or slow. Design the solution robustly enough to set clear expectations on managing requirements changes among the team, customers and management personnel. Involve affected groups in the creation or review of the process to work out difficulties.

To establish your requirements management process, answer these questions:

1. What is the objective of the requirements management process?

2. Who will collect requirements changes?

- Select a project member, analyst or manager.

3. Who needs to evaluate and agree to requirements changes before they are made?

  e.g., Customer champions, managers, team members and impacted system representatives.

4. How will the requirements be stored and labeled so that there is no confusion regarding the current state of the requirements?

  e.g., Place requirements document under version control or use a requirements management tool.

5. How will impact to schedules be evaluated?

6. How will approved schedule and requirement changes be communicated to affected parties?

Example process

1. Objective: Requirements changes will be collected, assessed, and agreed to jointly, based on the estimated effort to complete the change, the risk to project success, and the desired release schedule.

2. All requirements changes will be emailed to change-request@sc.com.

- The project manager will collect all requests received in the file, reqs-change-log-versionN.doc. All requirements changes that are not submitted to change-request@sc.com will be ignored.

3. The Requirements Review Board (RRB) will evaluate all major requirements changes weekly. The RRB will consist of the development team, customer champions and product management.

- The change request list will be emailed to the team prior to the session. Each requirement will have a preliminary evaluation of major or minor. The team will discuss the major items and review the minor items.
- More frequent RRB sessions can be held to process changes within one week.
- The RRB will decide on each major change using the status labels defined in step four.
- The project manager will decide on minor changes.

4. The requirements will be stored in the file, product-requirements-verN.doc. The version number (N) will be incremented to reflect a newly-published version.

- Each requirement will be labeled, "Approve now," "Approve for subsequent release," "Reject," or "Evaluate further."

5. The project manager will assess the schedule impact of all minor changes with no further required discussion. The RRB will evaluate all other changes with respect to risk, cost (effort) and schedule.

6. All new requirements changes and revised schedules will be emailed to the product team and reviewed in the bi-weekly team status meeting.

MANAGING REQUIREMENTS *(Continued from page 2)*

Fast track

Notice in the example process there is a fast track. Changes that can easily be evaluated as minor can pass through with an appropriate label. The RRB can then deal with the changes that need attention. You might decide all change requests that are either low risk, or take less than half a day to implement are "minor." However, if the impact cannot be easily determined, then it should proceed to the RRB for a more accurate evaluation.

Establishing a requirements management process is not only the key to setting stakeholder expectations; it changes how managers and customers make requests. Customers and managers know conceptually that all changes have an associated cost, but they might not have internalized this because previous requests were accepted with the response, "No problem, we can do that." When they know that the impact of requirements changes will be assessed and communicated, they are more careful in the changes they request.

Different implementations for different situations

Design the requirements management process to suit the complexity of the problem you have. Use the process you have designed and frequently make changes to match your situation. If you suggest to all players that the change control process should be dumped, and they agree, you don't have an effective process yet.

Consider the following implementations when you implement requirements management:

1. A simple change log and control board

For most projects in one location, a simple text file or spreadsheet under version control is adequate to record requirements and changes. The control board might consist of the whole project team, or representatives of development, test and requirements, along with the project manager. When there are requirements changes, the control board meets (in person or electronically) to decide on the changes.

2. Involving cross-functional teams on the control board

Cross-functional teams should be represented on the board when the requirements changes impact other teams and systems. For example, changing one field in a database might be good for the database team, but it could cause application teams to spend weeks of additional time upgrading and testing their programs. If there is a group that would be upset with you if a change were made secretly, then that group should be represented on the control board!

A text file or spreadsheet might still be adequate to record requirements and associated changes, but thought is needed regarding where the file is stored, who can change it, and how impacted areas are informed of new versions. A web page might be a good solution with automatic email notification of changes.

For very large systems, you might have several control boards that look after unique sets of requirements (for example, one board for user interface changes, one for changes to hardware, and one for changes that impact middle-ware programs). The boards could operate individually with representatives of each board being a member of the other boards to verify the impact of specific changes.

3. Using a requirements management tool

Requirements management tools (such as DOORS and Requisite Pro [1]) are effective for storing requirements and making them accessible to other groups [2]. Requirements tools assist you in storing additional information regarding each change, and enforcing the change control process. Tools should be considered once requirements are well written and managed. At this point, the needs for a tool are understood and sufficient data exist to make the tool a valuable investment.

Summary

Requirements change management is the filter that verifies that customer expectations are managed and that the project is investing its scarce resources wisely. Resistance to requirements management is addressed when the relevant stakeholders have their needs captured and addressed by the process.

Implementing requirements management is not conceptually difficult, but it does require thought regarding who should be involved, where data are kept, and how the status of changes should be communicated. It also needs someone to lead the way and set the example.

References

[1] Guidance on selecting a requirements management tool: http://66.34.135.97/lib/rmtools.html

[2] Wiegers, K. Software Requirements. P 253, Second Edition, Redmond, WA: Microsoft Press, 2003.

"DEALING WITH THE CUSTOMER IS TOO DIFFICULT AND TAKES TOO MUCH TIME"

(Continued from page 1)

For example, pick a package that you use often and think about all the features that it has. How many of the features do you actually use? In some cases, packages have been bloated with features that no one uses, but they seemed like a great idea at the time. Some of us will go a lifetime and never use half the features, which is probably a good thing, since several of them might not have been tested. The creator could have probably left out several features and spent more time making the remaining features work better.

When we engage the users in discussion about how long it takes to develop specific feature sets, we can help them see the trade-offs. The users themselves are often the best people to choose what should be included, what should be delayed for another release, or canceled.

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. Reduce costs.**
In this workshop, ACCELERATING PRODUCT DEVELOPMENT FOR SOFTWARE PROJECTS THROUGH LEAN CYCLE TIME REDUCTION, project managers and their teams learn how to accelerate delivery through specialized schedule optimization and "Lean Thinking" 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 CMM/CMMI gap-analysis.**
The following workshops are available:
- ❏ SEI CMM: LEVEL 2 (one day), SEI LEVEL 3 (two days), SEI LEVEL 4 (one day)
- ❏ SEI CMMI: Overview (half day), LEVEL 2 (one day), LEVEL 3 (two days)
- ❏ SEI INTRODUCTION TO CMMI (three days)

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

❏ **Goal/problem-based improvement.**
This two-day MAKING PROCESS IMPROVEMENT WORK 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.

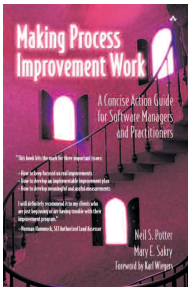
❏ **Manage your subcontractors.**
In this one and one-half-day workshop, SOFTWARE SUBCONTRACT MANAGEMENT, software engineers, managers and subcontract managers learn how to define a software product to be outsourced, write a subcontract management plan, select appropriate vendors and manage the project to completion.

❏ **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.)

*Detailed information on our services is available at **www.processgroup.com**. Contact us at **972-418-9541** or **help@processgroup.com** to discuss your needs.*

Come see our book!

www.processgroup.com/tpgbook.htm
Also available in Chinese at
www.china-pub.com



Here is the book's Table of Contents:

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