

The Software Quality Advisor Online

Increasing the Value of Every Member of Your Team Part 2

Randall W. Rice, CSQA, CSTE



Increasing the Value of Every Member of Your Team, Part 2

In Part 1, we discussed how people are typically underutilized on a team and looked at the nature of "rustout."

In this article, we will show seven specific ways to add value to your team in measurable ways, such as:

- Grow each member cross-functionally
- Find each person's passion and focus on it
- Make each person an expert on something
- Make each person a mentor to someone

Measuring Your Improvement

For each of the seven areas of improvement, I asked myself, "What value was added to teams that I have led to improve in these areas?" My numbers may not be the same as yours. It's OK to determine the percentage gains for your own team or organization.

Seven Ways to Increase the Value of Your Team

With all of that out of the way, let's look at seven ways you can increase the value of your team.

#1 – Grow Each Team Member Cross-functionally

Most people can do many things well. When people can perform multiple tasks:

- They add depth to the

team. If something happens to one person on the team, you have a reserve.

- They relate to what others do, so they have an understanding of what others go through to do their jobs.
- They have a greater sense of belonging because they can do more than just one thing.
- They can multiply time and effort by picking up others' work when needed.

A major question is "How do we grow each person to perform multiple functions?" Part of the answer is in the next point on finding what people are passionate about

doing. Look for talents and skills. Most people have many talents – tap into them! Another part of the answer is to build skills by training and mentoring. However, I mention that after passion and talents because training builds on what is already in a person.

Add 10% overall increase in value for each function on your team where you have cross-functional support.

#2 – Find Each Person's Passion and Focus on it

(Continued on page 2)

Inside this issue:

Links, Quotes, and Questions from the Mailbag 4

Change...A Process To Be Managed 7

Calendar of Events 8

Book Review—*Net Privacy* by Michael Erbschloe and John Vacca

Overview

The subtitle of *Net Privacy* is "A guide to implementing an ironclad ebusiness privacy plan." The subtitle is an excellent short synopsis of the book. I ran across this book while doing research on the privacy topics of information security for my new security testing course.¹

What I Liked About the Book

I found the following very good about *Net Privacy*:

- ***An organizational emphasis on information privacy***

The authors go to great lengths to reinforce the point that information privacy is not a technology issue—it's an organizational concern. That means that

assuring information privacy must not only be rooted in a sound and workable information privacy policy, but that policy must 1) be understood by everyone in the organization, 2) must be followed by everyone in the organization, 3) must be followed by business partners and suppliers, and 4) must be tested.

- ***The message that privacy measures can be tested***

In fact, this book even shows how to test for privacy leaks. Most books just mention something like, "You need to test," but they fail to give any kind of guidance or examples. This book gives several good ways to test privacy in your organization.



(Continued on page 2)

Book Review—*Net Privacy*

(Continued from page 1)

- **Adequate detail**

I never left any topic asking “why?” or “how?”. The level of detail is a good balance between readability and having enough detail to explain the topics.

• **Examples**

There are many real-world examples explained in the book about how information privacy on the web has been compromised and how it impacted the victims and the web sites that allowed it to happen. These examples are good to build a case for strong privacy policies.

• **Proven and practical techniques**

The techniques shown in the book are solid and workable. They may not be easy to apply because of organizational issues, but

they are what is needed to protect data privacy on the web.

• **Templates and checklists**

There are great examples of how to apply the concepts and things to keep checking. Data privacy concerns, like security and many other concerns are ongoing. These tools help you keep up with the concerns.

Audience

- Senior business management
- Senior IT management
- Webmasters
- QA analysts
- Testers
- IT Auditors

Topics and Outline

1. The Threat to Privacy and Corporate Vulnerability
2. The Nature of Privacy Problems
3. The Regulatory and Legislative Environment
4. Organizing to Protect Privacy
5. Conducting a Privacy-Needs Audit
6. Evaluating Technology Needs for Privacy Protection
7. Developing the Enterprise Privacy Plan
8. Implementing the Enterprise Privacy Plan
9. Managing Privacy on the Enterprise Web Site
10. Managing Privacy on Internet Supply Chains
11. Managing Privacy Efforts over the Long-Term
12. Protecting the Privacy of Enterprise Storage and Processing
13. Protecting the Privacy of Corporate Communications

(Continued on page 5)

Increasing the Value of Every Member of Your Team Part 2

(Continued from page 1)

We all have different motivations for coming to work every day. For some, it's the money. For others, it may be security. However, if people on your team are working at tasks they could not care less about, then you've got a passionless bunch of people on your hands.

I often get comments on my training evaluations that read something like, “You can tell you have passion about what you are teaching.” I have thought before, “Thanks for the compliment, but pity me if I didn't like what I do!” Really, I mean I would go nuts if I didn't care about communicating about perhaps one of the most boring topics in the world – software testing. To me it's a challenge to find new and unique ways to get the points across in ways that stick in people's minds. It's so rewarding to see a new test team leader leave a class with a dozen ideas to try – and they are excited about it.

Your challenge is to act like a coach and find what motivates or de-motivates every single person on your team. This may not be quick or easy. You may be challenged with finding a way to match a person's passion with the task. Here are a couple of examples:

Sue wanted to be a software developer in college and she graduated with a computer science degree. After she graduated, she had a hard time finding a job, so she took a job with your company as an entry-level tester just to

get her foot in the door. For six months, she has been trying to break software, but she really enjoys building software instead of breaking it. By this point, Sue is becoming bored and disillusioned – a prime candidate for rustout. One day her test team leader asks if she would like to start working with the new automated test tool the company purchased. No one else on the team can code and everyone else has given up trying to use the tool. Sue jumps in and learns the tool in two days and within two weeks has created automated test scripts for over 30% of the existing test script library. The scripts work great and Sue actually likes going to work. She hasn't given up her goal of being a software developer, but she is happy again and very productive. Not only does she add value to the team by being more productive in her job, she has probably saved the company over \$100,000 had the test tool fallen into non-use.

Larry had a reputation for being at work, but in body only. He seldom associated with co-workers after hours and was in rustout mode. One day, the team got a new leader, Mike. Right after the orientation with his new team, Mike asked Larry into his office. “Larry, I understand that you really don't seem to like it much around here,” Mike stated. “Yeah, I guess you could say that,” responded Larry with a definite feeling of “I wonder how long this meeting will last?” “Well, what do you like to do?” asked the team leader. “You mean at work or anywhere?” asked Larry. “Anywhere,” responded Mike. “I really like to work on PCs. That's my hobby. I have six under construction at home right now,” answered Larry. Mike's eyes lit up. “Is that a fact? Well, have I got a deal for you. As of today, you are in charge of configuring our test lab,” said Mike. “You're kidding. I've been wanting to do that for months, but nobody would take me seriously,” said Larry. “I am,” said Mike. “One other thing you need to know is that when I give someone

“Your challenge is to act like a coach and find what motivates or de-motivates every single person on your team.”

(Continued on page 3)

Increasing the Value of Every Member of Your Team Part 2

(Continued from page 2)

charge of something, I also give them responsibility and accountability. That means the lab is your baby. When we need to run a test, it had better be ready," emphasized Mike. "Hey, no problem," said Larry. As the weeks went by, people began to notice a change in Larry. He was the first one in and the last to leave on many days. Not only did Larry excel in maintaining the test lab, he designed the processes to make it run efficiently. He also became more likeable and even started hanging out with co-workers outside of work. Larry was working in his area of passion, but it took an observant and skilled leader to bring it out.¹

OK, you get the idea. Think about each person on your team. Do you know what excites them? What bores them? If not, it's time to build some relationships to find out.

To measure this area, give yourself a 20% increase of each team member that is working in their area of passion and motivation.

#3 - Make Each Team Member an Expert on Something

This may sound like a stretch at first, but this is a journey, OK? You may need to combine this item with #6, Constantly Work to Build and Reinforce Skills.

When each person on the team is an expert in something, it does some major things:

- 1) People feel respected – it really builds their self-esteem
- 2) People are motivated to learn more about their area of expertise
- 3) The team has an asset in knowledge that continues to grow
- 4) It's like having an on-staff consultant
- 5) You have a person that can handle a major crisis
- 6) There is less guesswork in how to solve problems

I'll never forget an assessment I performed at a company several years ago. The software development and testing at this company was in really

bad shape. I had never had any organization score a zero before on the scorecard! Yet in the midst of failure, there was one team that stood out. They always implemented projects on time, always pleased their users, delivered quality work, and...actually seemed to be having a good time in the process. After digging deeper into what made this team click, here's what I found:

- 1) Their manager was not technical. In fact, he was from marketing (I know all you Dilbert fans are cringing.)
- 2) Each person on the team was a resident expert at something. They had a requirements analyst, system designer, coder, tester, and a troubleshooter.
- 3) Anytime the occasion came up for an "expert opinion" the team deferred to the expert on the team. Whatever solution the expert advised was accepted as the appropriate course of action.
- 4) They did not get this way overnight. They had to build skills and develop trust as a team.

The question is, "How do you become an expert in something?" We'll examine part of the answer in #6. But for now, here's a couple of starters:

- 1) If you read thirty minutes a day on the same topic, in five years you'll be an expert in that topic. In fact, in some disciplines, it make even less time. Hey, I never promised this would be quick!
- 2) There's no substitute for experience! You learn much more from your mistakes than in the sterile classroom environment. Let people experiment when possible.

We'll also see how some of the other seven areas can add to this one.

#4 – Make Each Person on the Team a Mentor

A mentor is a skilled and trusted advisor, a personal trainer. Everyone should take someone else under their wing to train them in the subtleties of doing something. People on the team can be a mentor to someone, not necessarily on the team.

A mentor can:

- 1) Be a driver for continuous learning
- 2) Can promote what your team does to people outside of the team
- 3) Can pass along valuable lessons learned
- 4) Can personalize training to individuals

Add 20% value for each mentor on your team.

#5 – Make Sure Each Person Has a Mentor

The mentor is the bridge between concept and practice. The mentor gives encouragement when the theory doesn't seem to work in practice. If you're looking for true Just-in-time training, get a mentor.

The mentor can be on the team or elsewhere, either in the organization or in another organization. The key is to help your team find that one right person who has "been there" and knows all the tricks. Also, the mentor has to be willing to give of their time. Most people are open to being a mentor, as it provides a way to give back to those who many years ago helped them. Don't be afraid to ask someone to be your mentor. Even if they decline, it's an extreme compliment to be asked.

Give yourself a 20% increase in effectiveness for each team member that has a mentor.

#6 – Constantly Work to Build and Reinforce Skills

This does not have to be done through formal training, although training is the most straightforward approach. You can also perform and receive mentoring, receive just-in-time training (such as web-based training), or perform independent research. You can also read books and articles on your desired topics.

Here are some of my favorite books for building testing skills:

For beginners:

Software Testing by Ron Patton



(Continued on page 5)

Links

Sample Change Control Procedure Document

<http://cn.esoftbank.com/SoftwareEngineering/documents/approved/standard/CC.PDF>

Sample Security Policy

<http://www.csuohio.edu/security/changecontrol.htm>

Lots of change control tools

http://directory.google.com/Top/Computers/Software/Configuration_Management/Tools/

From Crosstalk:

Effective Configuration Management

<http://www.stsc.hill.af.mil/crosstalk/1998/feb/effectivecm.pdf>

The CM Database: To Buy or Build

<http://www.stsc.hill.af.mil/crosstalk/2000/jan/sorensen.asp>

A Configuration Manager's Perspective

<http://www.stsc.hill.af.mil/crosstalk/2000/jul/starbuck.asp>

Stop-Gap Configuration Management

<http://www.stsc.hill.af.mil/crosstalk/1998/feb/stopgapcm.asp>

Software Configuration Management: Don't Buy a Tool First

<http://www.stsc.hill.af.mil/crosstalk/1997/nov/configuration.asp>

Software Configuration Management: A Discipline with Added Value

<http://www.stsc.hill.af.mil/crosstalk/2001/jul/butler.asp>



Quotes

"You do not lead by hitting people over the head - that's assault, not leadership." - Dwight D. Eisenhower

"Inventories can be managed, but people must be led." - H. Ross Perot

"The key to successful leadership today is influence, not authority." - Ken Blanchard

"The most dangerous leadership myth is that leaders are born - that there is a genetic factor to leadership. This myth asserts that people simply either have certain

charismatic qualities or not. That's nonsense; in fact, the opposite is true. Leaders are made rather than born." - Warren Bennis

"Never criticize someone until you have walked a mile in their shoes. That way you are a mile away and plus, you have their shoes." - Garrison Keillor

"I believe in looking reality straight in the eye and denying it." - Garrison Keillor

"It's a shallow life that doesn't give a person a few scars." - Garrison Keillor

"God writes a lot of comedy... the trouble is, he's stuck with so many bad actors who don't know how to play funny." - Garrison Keillor

"Quality isn't something that can be argued into an article or promised into it. It must be put there. If it isn't put there, the finest sales talk in the world won't act as a substitute." - C. G. Campbell

"Fast is fine, but accuracy is everything." - Wyatt Earp

"If you think of standardization as the best that you know today, but which is to be improved tomorrow; you get somewhere." - Henry Ford



Questions From the e-Mail Bag

Q: I have some questions about user acceptance testing (UAT). Who does it? Is it the business users?

A: UAT should be planned, performed and evaluated by the people who will actually be using the software. The user may be the same people as the customer, but in large-scale information systems, the customer (sponsor) hardly ever uses the system they have built or bought.

One of the issues at this point is

the concern that perhaps the "real users" may not be skilled enough to plan and perform an adequate test. I usually respond, "OK, so you're telling me that you trust the users to conduct business with it, though?"

Q: Is the QA/testing team involved in UAT?

A: Sometimes the users need people to help them in a facilitation role. I just advise that the QA or test team do not end up performing the test for the users.

Some examples of tasks that may need facilitation are: defect reporting, test planning (scripting, etc.), identification of business or test processes for test scenarios.

Q: Who signs off on UAT?

A: This can vary from one organization to the next. In some places, people will not sign-off on anything. However, if a sign-off is required, it is often by the highest ranking business manager over the areas impacted by the system.

(Continued on page 5)



Questions from the Mail Bag

(Continued from page 4)

Here's two other important points regarding sign-offs:

There is a big difference between a user sign-off and an implementation decision. Just because the user deems the system acceptable does not mean it is ready to deploy. Have people been trained? Is the help desk ready? Is production support ready? The implementation decision should be a management level team decision. Information from UAT and other phases of testing are simply input to the decision.

Second, in many UAT efforts I have seen, the acceptance of the system is a foregone conclusion. The contract has been signed, dollars have been spent and the system is going in. The question is, "Where are the gaps between the business processes and the

system processes?" UAT gives an opportunity to identify and close the gaps before deploying the system. I see more of that kind of evaluation as opposed to a "thumbs up or thumbs down" acceptance.

Q: What is a Service Level Agreement?

A: A Service Level Agreement (SLA) is an agreement that specifies the level of service one organization will provide to another organization. The organizations may be internal to the same major organization, such as an IT department serving a business unit, or the organizations could be external. An example would be a software vendor or a web service providing services to a customer. The SLA can address many areas. Typically, you will find quality levels (such as defects), performance levels (such as response

time for interactive systems, or response time for dealing with problems). A SLA is very similar to a contract. The idea of SLAs internally give the look and feel of a contract without the force of law to deal with breaches.

Basically, the SLA puts an agreement in writing between two parties.

SLAs have been around for a long time, but have found a new life in areas such as web site up-time and vendor quality of COTS (Commercial-off-the-shelf) software.

For more information, check out the book: Foundations of Service Level Management by Rick Sturm, Wayne Morris, and Mary Jander



"Just because the user deems the system acceptable does not mean it is ready to deploy."



Book Review—Net Privacy

(Continued from page 2)

14. Protecting Corporate Desktop Privacy
15. Protecting the Privacy of the Road Warrior's Laptop
16. Protecting the Privacy of Remote Access and Telecommuters
17. The Future of Privacy Management

Good Companion Books

Web Security, Privacy and Commerce by Simon Garfinkel, Gene Spafford (Contributor), Debby Russell

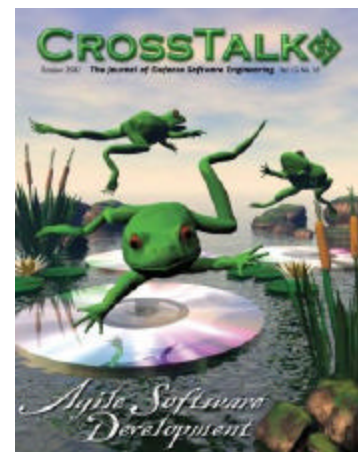
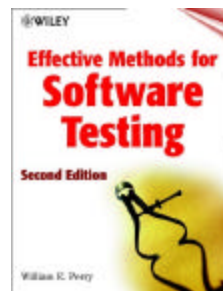
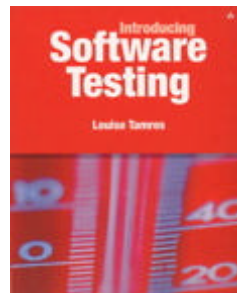
Summary

I can highly recommend this book to anyone in an organization that has been charged with the task of data privacy. This is a very readable book and is also well-suited for any level of management. Every organization has privacy concerns. Find out what your organization's and customers' privacy concerns are by getting this book and learning how to protect them.

Reviewed by Randall W. Rice

Increasing the Value of Every Member of Your Team - Part 2

(Continued from page 3)



For beginners to intermediate:

Introducing Software Testing by Louise Tamres
Effective Methods for Software Testing, 2nd Ed., by William E. Perry

My favorite journals and magazines are:

Crosstalk (www.stsc.hill.af.mil)
Application Development Trends
STQE (www.sqe.com)

Another way to build your skills is to study for an obtain certification in software testing. There are several sources for software quality certifications:

American Society for Quality (ASQ) Certified Software Quality Engineer (CSQE) – <http://www.asq.org/>
International Institute for Software Testing (IIST) Certified Software Test Professional

(Continued on page 7)

Change...A Process To Be Managed by Steve DeMarte

It has been said that the only thing in life that is constant is change. Whether one looks at the weather, the seasons, or the human condition, change is an integral part of the experience. This kind of change is usually welcomed only because we have come to accept it. In other circumstances, people do not readily accepted change. Change can often wreak havoc if people are not prepared for it and do not have ways to manage it. In the world of software projects, change can complicate and frustrate even the simplest of matters.

Just like the seasons change, changes will happen during the software development process. To successfully deal with change, you need adaptive techniques to manage it. These techniques range from simple to complex and sophisticated in nature. In this article, I will describe change in the context of issue management and workflow, and how these practices help manage change during a project.

What's Your Issue?

We all have issues. In software development, issues should be considered specifically. I like to think of issues as *defects* and *change requests*. These issue types:

- Can both initiate change
- Are inherent to any software development project
- Require a different type of management

As software testing takes place, defect and change request issues are identified. These issues must be managed through a process that provides details on each issue. The details on an issue usually include information that helps in the transition of the issue from one state to another. Detail items like "issue type", "severity", "description", "assigned to", etc., all help describe issues in a way that allow them to be managed throughout its lifecycle.

Differences Between Defect Reports and Change Requests

Before proceeding, let's discuss the difference between a defect report and change request. A defect report is an issue type that describes a problem with the application-under-test. It is a problem that exists *specifically* due to the lack of defined functionality. Defects (or bugs) should be described in terms similar to the requirements being tested against. If there are no requirements, then the defect is described in terms of function or attributes as best the tester can describe them at the time. This points out the need for defined requirements as one of the baselines for testing. It is also important to assign a defect issue a severity level. This helps greatly in determining prioritization of work

and aids in risk assessment and risk management.

A change request is an issue type that arises out of the need to add or modify functionality. Change requests are often the result of usability testing sessions where the user community wants something changed. From my experience, whether testing against detailed requirements or not, the user will want some sort of change during the project. It is as equally important to document these issues as well as defects. Change requests should be documented in a way to assure traceability to updated requirements.

There are two key parts in tracking and managing change requests:

- The capture of information for the change request
- The update to existing requirements

All work driven by the change should be traceable to the change request. We will examine why later in this article. Unlike defects, change requests have to be managed to prevent feature creep versus, as with defects, managed to prevent loss of functionality or regression defects. It is important to distinguish between defect reports and change requests in order to ensure that effective change management can take place. Two questions that deserve further consideration are:

- What are some of the aspects of change management?
- How do I know that change is being well managed?

Aspects of Change Management

Part of the change management process is to differentiate between defect reports and change requests. However, that is only a part of the process. Other key factors in change management are tracking, workflow, and change approval.

Tracking issues allows for the analysis of assigned work and helps facilitate issue resolution. It is very important that all issues transition eventually to some form of resolution. Issues, regardless of type, must be tracked through its lifecycle to a resolution.

Issues can be tracked manually through the use of issue reports or through the use of issue tracking and workflow tools. I strongly recommended the use of workflow tools as it lends itself to much better management. There are a host of tools available, but be careful about tool selection. Remember, a fool with a tool is still a fool with a tool. Your tool selection should make sense in your environment and should be accessible by everyone who will be submitting issues.

Most importantly, choose a tool that is practical for your organization. Some of the workflow tools are far easier to install and configure than others. Ironically, some of the defect tracking and workflow tools have bugs in them, so be sure and talk with current and past users of the tool in other organizations. Vendor support is also critical. Forums such as www.qaforums.com, www.stickymids.com and www.softwaredioxide.com are helpful places to get the real story about specific tools and vendors.

How to Know that Change is Well-Managed

An issue is generally tracked through various states until it is transitioned to some sort of resolution or closure. When an issue is initially reported, there should be in place a process of immediate review. This review is done by key stakeholders who usually have responsibility for the allocation of work. The review does a number of things:

- **First, it allows for agreement on issue typing and triggers some sort of work assignment.**

Assignment will generally be to developers, testers, or change approval staff, but could be to tech writers or anyone having responsibility for work on an issue. It is the process step where an issue state changes and begins its flow through its cycle. Once an issue is tracked to individuals the workflow process begins.

Workflow is a critical part of any change management scheme. No application development manager should be without a process of determining allocation and cost of work. Workflow enables the reporting and analysis of all issues relevant to a project. These issues should all be treated as units of work and tracked for project planning and costing purposes. I have been involved in many projects where managers are committed to the "drop dead date" and have never taken into account the work associated with outstanding issues. Project planning should always include task items specific to defect fixes and change request implementation. The only way to do this is to track and manage issues.

Workflow also allows for the transitioning of an issue from one state to another. For example, an issue initially enters the system in a "new" state. An individual or team is responsible for reviewing newly reported issues. Workflow dictates that an issue must transition to a state where the issue will be *satisfactorily* addressed and eventually resolved. Whether the work on the issue should begin, be deferred, or not be done at all, workflow must be in place to expedite the transition of the issue. Many times, issues are not easily resolved. Therefore, they must be resolved in some sort of forum. The resolution must result in approval, by a majority of those involved, of some sort of action. This dialog and control is a key factor in the change management process. All defect issues that cannot be resolved and all change request issues must go through change control.

(Continued on page 7)

Increasing the Value of Every Member of Your Team Part 2

(Continued from page 5)

(CSTP) – www.softdim.com
Quality Assurance Institute (QAI) Certified Software Test Engineer
(CSTE) – www.qaiusa.com
Quality Assurance Institute (QAI) Certified Software Quality Analyst
(CSQA) – www.qaiusa.com

The above certification programs vary widely in their approaches, investment and objectives for certifications, so **I advise you to carefully research and compare details about each program** to learn what it take to obtain and keep the certification.

The key is to have your own personal growth plan to build your skills – and to follow it!

Add 25% value for each person on your team that has and follows a skill-building plan.

#7 – Play to Your Strengths, Add to Fill Gaps

At first glance, it seems to make sense that we build up our weak areas. However, you will be much more effective in what you do if you focus on what you do best and continue to perfect those things. No one can do all things well, so it makes sense to find those people who can do the things well that you or others on your team don't do so well.

An Example

Just to see how powerful these adjustments can be, let's assume you have a 7 person team that performs 8 major functions:

- Each function can be performed by more than one person: +80%
- Each person working in their area of passion: +140%
- Each person is an expert on something: +70%
- Each person is a mentor: +140%
- Each person has a mentor: +140%
- Each person has a person growth plan: +175%
- Each person is working in their area of strength: +175%

The sum of the above items yield a 920% increase in value!

Don't Like My Numbers?

The percentages are based in my own estimation of what level of value added the above seven actions would bring to a team. If you think the numbers are too high or too low, then plug in your own estimates. The main thing is to define a way to measure to increase in effectiveness and value.

Summary

Effective teams don't just happen – they are cultivated. Even when you don't get to pick your team, you can still make it a success by leveraging your talents and passions. Teams are made up of people and people are your greatest asset. Try one, two or all seven of the value-added items and see what happens!

1. This is a true story. The names were changed to protect the innocent.

Change...A Process To Be Managed

(Continued from page 6)

• Second, reviews facilitate and monitor change control

Change control is another key part of the change management process. It should be done involving key stakeholders on any given project. Project managers, lead developers and testers, and key users are often members of a change control board (CCB). It is the responsibility of these people to resolve those defect issues that cannot be brought to satisfactory resolution through the standard workflow model. In other words, defect issues and change request issues are the sole responsibility of the change control board.

A common question is "why submit defect issues to the change control board? Development leads may want to defer some defect fixes based on work priority, while test team members might disagree with the decision to defer. These disagreements can be presented to the change control board for resolution. A decision could be made to defer a fix based on the relatively low severity of the issue. The change control board should agree on how to resolve the issue, and the issue should be documented accordingly. Defect issues are often interpreted by developers one way and another way by testers. What may appear to be 'functioning as designed' by one may be defective to someone else. Sometimes the cause for such disagreements are ambiguous requirements. The change control board must clear such ambiguities and enable the transition of the issue accordingly.

Change request issue types take on an altogether different flavor of responsibility. It should not be up to developers or testers to determine whether to do change request work, as change requests cost additional money! Plus, the cost of changes multiplies greatly toward the end of the project and can even put the project at risk of failure.

• Finally, reviews provide a way to prioritize and approve changes

It is important that change requests be analyzed for impact on the overall project. To manage change in this regard, is to manage cost. It is a fact of software development life that customers will want change. Project managers cannot simply say "No!", especially to the project sponsor. What should be said, though, is that change requests brought into scope by the change control board will impact delivery and cost. If the sponsor and other stakeholders agree to proceed with the change, adjustments to the project plan concerning delivery and cost should be made.

If the decision is not to proceed with the change, it is important to get agreement and document that without the change the product is still viable and acceptable to the user community. Change requests are often times deferred due to project deadlines and anticipated delivery dates. Deferral decisions are acceptable as long as key user stakeholders agree on them.

Summary

Software development organizations will always have to deal with change. Those that understand how to manage it will certainly do better at implementing projects. Issues that come up during the project lifecycle should be addressed as part of the development process. As issues work their way through the workflow process to resolution, effective change management practices will help minimize cost and risk associated with changes. As with the weather, and the seasons, change within our software organizations should be a welcomed one.

Steve DeMarte is a Quality Assurance and Testing consultant based out of Raleigh, NC. He is president of QA Insights and has experience in managing the change control process in organizations.



October, 2002

© 2002, Rice Consulting Solutions, LLC

P.O. Box 891284
Oklahoma City, OK 73189

405-793-7449
405-793-7454 Fax
rice@riceconsulting.com

"Test everything. Hold onto the good."
I Thessalonians 5:21

We're on the Web!
www.riceconsulting.com

Testing COTS-based Applications
by Randall W. Rice

Book Review — Questioning Extreme Programming
by Pete McBreen

Coming to Chicago!

October 30 — November 1, 2002

A Three-day course in Web Testing Techniques



Presented by Process
Management Group, Ltd.
(www.pmg ltd.com), the Midwest's Premier Provider of IT
Quality and Software Testing Services, and
Rice Consulting Services, LLC
(www.riceconsulting.com),
a world recognized leader in Quality and Testing Training.

See details and register at
http://www.riceconsulting.com/chicagoq4_2002.htm

Calendar of Events

A Three-day Course in Web Testing

Chicago, IL, October 30—November 1,
2002

Sponsored by the Process Management
Group, Ltd. And Rice Consulting Solutions,
LLC

[http://www.riceconsulting.com/
chicagoq4_2002.htm](http://www.riceconsulting.com/chicagoq4_2002.htm)

EUROStar Conference

November 11—15, 2002

Edinburgh, Scotland

[Http://www.testingconferences.com](http://www.testingconferences.com)

Randy will be presenting a one-day tutorial
on *Surviving the Top Ten Challenges of
Software Testing* and a keynote address on
*Getting the Most Value from Every Person
on Your Test Team*.

**We hope to see you at one of these
events!**



If you have a group of 12 or more
people in your city that would like to
sponsor a training event, contact
Randy Rice at rrice@riceconsulting.com
to find out how to book a special
presentation.