

The Software Quality Advisor Online

Increasing the Value of Every Member of Your Team Part 1

By Randall W. Rice, CSQA, CSTE



Increasing the Value of Every Member of Your Team

Organizations are keenly interested in how to make the most of budgets and get the best performance from their people. At the same time, studies show that people are underutilized to the point of rusting out instead of burning out!

In this article, the first of two parts, I will examine:

- The degree and nature of rustout
- How to Reverse the situation
- Maxwell's laws of teamwork.

In today's economy, it seems that management is keenly aware of the need to maximize the value of all resources, including human resources. At the same time, it seems that we are living in the land of Dilbert, where people show up for work, deal with bureaucratic systems that make little sense and consume huge amounts of time, and generally give up trying to be creative or to achieve high levels of personal effectiveness.

In this article, we will explore how to leverage your current team structure and talents to greatly increase your overall effectiveness, and therefore, your team's value to the organization.

It is important at the outset to understand that we will be discussing not working longer or harder, but rather unleashing the incredible potential within each person when they are allowed to work in their area of passion.

It is my observation in working with

organizations of all types and sizes that many people work well below their potential. I believe that by making minor changes to the team structure and giving people opportunities for growth, each person's increased values can be multiplied in measurable ways.

How Bad is the Problem?

In a recent CNN Moneyline web poll, 28% of respondents said they were satisfied with their jobs, 15% were somewhat satisfied, and the rest (57%) were just ready for 5:00 p.m. to arrive.



One of the pioneers of the quality movement worldwide, Dr. W. Edwards Deming, wrote in his landmark book, *Out of*

the Crisis (1986), "With the storehouse of skills and knowledge contained in its millions of unemployed and with the even more appalling underuse, misuse and abuse of skills and knowledge in the army of employed people in all ranks and industries, the United States may be today the most underdeveloped nation in the world."¹

That was Dr. Deming's subjective observation. Let's look at some other indicators of low value. One indicator is what people do while

(Continued on page 2)

Inside this issue:

Links, Quotes, and Questions from the Mailbag 4

Calendar of Events 7

Book Review—*Managing Software Acquisition: Open Systems and COT Products* by B. Craig Meyers and Patricia Obendorf

Format: Hardcover, 400pp.
ISBN: 0201704544
Publisher: Addison-Wesley
Pub. Date: June 2001

Overview

This book is aimed toward project managers and their staffs who are involved in projects that may involve open systems and Commercial Off-the-Shelf (COTS) products. This book focuses on principles of open systems and COTS, not detailed technical issues or checklists and templates. This approach makes sense as there are so many possible technical and business environments, each with their own requirements.

The book had its origin in seminars developed and taught at the Software Engineering Institute (SEI) at Carnegie Mellon University (www.sei.cmu.edu).

According to the authors, the objectives of the book are to:

- Define basic terms, concepts, and processes related to open systems and the use of COTS products
- Explain the potential benefits and difficulties of using an approach that relies on open systems and COTS products



buy it now

(Continued on page 2)

Book Review—Managing Software Acquisition: Open Systems and COTS Products

(Continued from page 1)

- Describe how open systems and COTS products affect the project manager and the project staff
- Illustrate how to incorporate open systems and COTS products in the acquisition process
- Highlight special concerns for government managers

What I Liked About This Book

- The Overall Approach and Message

I liked the overview approach and the organization of the book. The book does a good job of explaining basic terminology and processes without getting too complex. It is a fairly easy read and addresses many of the real world concerns of COTS.

- A Dose of Reality

This book drives home the point that acquiring software is not necessarily easier or less risky than building it. Many people falsely believe that because someone else develops the software your concerns will be fewer. This book makes the point that your concerns will be different, but not necessarily less than in-house developed systems.

- A Good Reality Check for Acquiring Software

I also like the message that just because you are acquiring software from an external source you don't need to be concerned with testing.

- Places to Reflect Upon and Apply Concepts

Each chapter ends with a "Food for Thought"

section that is a nice springboard for your own research and application.

- Considerations for Government Environments

Finally, I liked the special comments throughout the book aimed toward the government sector. Although the private sector certainly has its own share of issues, many books on IT topics do not address the unique concerns of perhaps the largest single customer of software, the government sector.

What I Would Have Liked to Have Seen

- More on Testing

Keeping in mind that I am a testing person, I was looking for more information on the testing

(Continued on page 5)

Increasing the Value of Every Member of Your Team Part 1

(Continued from page 1)

they are at work on non-work activities.

- Online industry analysts predict that Internet misuse will cost companies an estimated 1 billion dollars in lost productivity (Newsweek).
- In a recent survey of 224 firms that utilized monitoring software, 60 percent of the managers said they had disciplined employees for online misuse, and 30 percent had fired people for such behavior, which included downloading pornography and shopping and gambling online (Websense Security Software).²

Another indicator is the occurrence of "rustout," a workplace malady in which an employee's potential is underused and his/her performance is mediocre. Rustout is more subtle and insidious than its better known counterpart, employee burnout.

Four Signs of Rustout

You can often recognize rustout by what is called "The Four Dis's."

Disengagement - Quit mentally, but still on the payroll

Dis-identification - Feels they used to be somebody, but not anymore

Disorientation - Don't know where they fit in

Disenchantment - Doesn't feel valued or doesn't feel talents are recognized.

How to Reverse the Situation

I believe the cure to rustout and the way to see a team increase its value to an organization is to leverage what you already have. When the fulcrum of a lever is set in the wrong place, it actually takes more force and effort to do work. By finding the most effective place for the fulcrum, one person can do the work of many. That's what we're going to discuss in the rest of this article – how to know where to leverage your efforts, and then, how to make the leverage point a reality.

Imagine that you are trying to move a large boulder using a (Continued on page 3)

"I believe the cure to rustout and the way to see a team increase its value to an organization is to leverage what you already have. When the fulcrum of a lever is set in the wrong place, it actually takes more force and effort to do work. By finding the most effective place for the fulcrum, one person can do the work of many"

Leverage Your Efforts

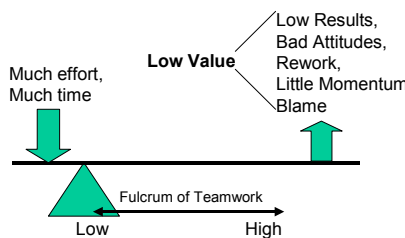


Figure 1—Poor leverage of effort by ineffective teamwork

Increasing the Value of Every Member of Your Team Part 1

(Continued from page 2)

lever. You would try to place the fulcrum of the lever as close to the rock as possible to get the greatest leverage. Likewise, with getting increased value from a team we want to place the leverage point at the most effective point. (Figure 2) Unfortunately, many organizations are so consumed with the crisis of the day that they fail to give thought to how to leverage the team's efforts. If you are in that situation, sit back and relax as you read the rest of this article. My desire is that you will get some ideas as you reflect on some of the seven ways I am going present to increase your team's effectiveness.



Figure 2 — Great leverage of effort by high levels of teamwork

I must be quick to point out that these are just things that have worked in my experience and that of others, but they may not be universal or a magic answer to your team. I advise trying these ideas one by one, but you will notice they have a synergistic effect. The more of the seven ideas you get to work, the more momentum you will achieve.

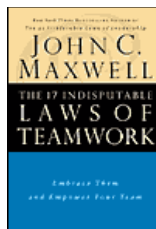
It is also very important to understand that the goal is not to work harder or longer, but to work smarter. This may be a radical thought in your organization if the prevailing management mindset is "If you're not doing anything, you're not being productive." I maintain that sometimes the most productive thing we can do sometimes is nothing! And by the way, it's really OK to have fun at work.

Maxwell's Laws



In John Maxwell's book, *The 17 Indispensable Laws of Teamwork*, he discusses some laws that apply very well here.

The Law of the Niche – "All players have a place where they add the most value."



In the law of the niche, the goal is to place the person where they add the most value. Although this seems obvious, in so many cases people are placed in situations where they do not perform well. The first solution people try to remedy a bad fit is to train them to fit. The problem is that many times the people don't want

to fit!

I have often found sports teams to be a good example of good and bad teams. A good coach spends time finding out what makes each player tick. Once the coach understands the player's temperament, skills, and talents, they can be placed in their best role on the team. Of course, we see just the opposite approach taken many time both in sports and in other organizations. In those cases the results are predictable: uninterested, unmotivated people on losing teams.

To find the right niche for someone you need to know the team, the situation the team is in, and the characteristics of each player.

As Vince Lombardi said, "The achievements of an organization are the results of the combined effort of each individual."

John Maxwell also comments on this law of the niche, "You may have a group of talented individuals, but if each person is not doing what adds the most value to the team, you won't achieve your potential as a team."



The Law of the Chain – "The strength of the team is impacted by the weakest link."



I can't help but think of the television program, "The Weakest Link", in writing about this law. The television show gets ruthless as people vote the person they see as the weakest link off the show. However, as the show progresses, the strategy shifts from voting off the weakest link to voting off the greatest threat. It's an interesting experiment in human behavior but a horrible example of teamwork!

Our goal in dealing with weak links on a team is to use an approach that builds the person and the team. It is helpful to understand at the outset that not every team is for every person. Some people simply don't want to be there and don't want to take the journey. Others shouldn't take the journey for a variety of reasons. Others can't take the journey. They lack the skills, the experience or the motivation to go on with the rest of the team.

As an old Boy Scout, I can tell you one thing for certain. The speed of the process depends on the slowest person. Sure, others can go on ahead but you don't arrive as a team.



So, there is an impact of weak links on the stronger links. The stronger links know who the weak ones are. The stronger members also have to help the weaker ones, which takes away from what the stronger links are working on. If left unaddressed, the stronger members can start to resent the weaker links. This situation can even cause the stronger team members to question the leader's ability. As Maxwell says, "You lose respect of the best when you don't deal properly with the worst."

There is no exact, repeatable rule for dealing with weak links. You have to consider the person, the team, and what's at stake to make a good decision.

The Law of the Bad Apple – "Rotten attitudes ruin a team"

Most people have seen this law at work on a team. Most of the responsibility is on the leader to deal with bad attitudes. There is no other way to deal with the issues than to discuss them with the

(Continued on page 4)

Links

The Software Testing Spot – A collection of testing links
<http://www30.brinkster.com/wvole/>

A really great slide show on review s
<http://www.stc-online.org/cd-rom/1998/slides/t5dcook.pdf>

A web site for software engineering
<http://www.software-engineer.org/index.php>

Book summaries on leadership
<http://home.earthlink.net/~denmartin/key-lead.html>

A white paper on software QA and testing
<http://www.aqtlab.com/Quality%20Assurance%20Reasons%20and%20Basics.pdf>

Optimizing software teamwork
<http://www.rational.com/products/whitepapers/100860.jsp>

Teaching teamwork (discusses the Team Software Process)
<http://www.computer.org/software/homepage/2002/05hil/print.htm>



Increasing the Value of Every Member of Your Team—Part 1

(Continued from page 3)

but give the person a chance to give their perspective first. If you are dealing with a group's bad attitude, start with the ringleader.

If the problem was due to a misperception, you can clarify it and go on. If there is truly an attitude problem, as a leader you need to clearly state your expectations and give the person a chance to change. If the person doesn't change for the better, then your only choice is to remove them from the team.

Next week, we will look at seven ways how you can increase the value of every member of your team in measurable ways.

1. Deming, W. Edwards, *Out of the Crisis*, 1986, pg 6.
2. Center for Online and Internet Addiction – <http://www.netaddiction.com/workplace.htm>

Quotes

"Sweat plus sacrifice equals success." - Charley Finley

"Our goals can only be reached through a vehicle of a plan, in which we must fervently believe, and upon which we must vigorously act. There is no other route to success." - Stephen A. Brennan

"If you're ridin' ahead of the herd, take a look back every now and then to make sure it's still there" - Will Rogers

"Always bear in mind that your own resolution to suc-

ceed is more important than any other one thing." - Abraham Lincoln

"Work is either fun or drudgery. It depends on your attitude. I like fun." - Colleen Barrett

"The greatest part of our happiness depends on our dispositions, not our circumstances." - Martha Washington

"It's hard to lead a cavalry charge if you think you look funny on a horse." - Adlai Stevenson



Questions From the e-Mail Bag

Comment: Just wanted to give you some feedback from your book, *Surviving the Top Ten Challenges of Software Testing*. There was a statement about how testers are educated and not born testers. I don't buy into all of this statement at all.

In my experience the best software testers I've come across are the type of people that can't leave stuff alone, they are always fiddling and trying to work out how stuff works. These guys are nightmares at times as they get distracted as soon as something flashes at them. "We were in Maastricht (The Netherlands)

working on a software project and this young Scottish guy was with us in this pub/bar. He saw the electronic darts scoring machine in the corner, he wasn't worried about playing darts, he just wanted to know how it worked, so after about 20 minutes he yelled "So, who wants to play darts - I'm scoring" These characteristics will certainly stand this guy in good stead within software testing as he had a natural aptitude for this work, BUT, I agree that without structured education this guy will do well but not brilliant. Education is often forgotten or certainly under realized especially in fast paced rapid development time

critical projects and this should not be the case in my humble opinion.

A: Actually, I agree with your comments about the natural attributes of a tester. I have tried to train some people to be testers who just were not cut out for it. The issue we were trying to address in the book was the attitude (by management mainly) that people can perform testing with their wits alone, without processes or training. I agree with you that I have known many testers that can go quite far on their own intuition. I have also been able to help people who didn't seem to be quite as natural in testing learn to think in ways that are similar to those with "the knack" for testing.

Thanks for reading the book and for your feedback. Perhaps one day there will be a second edition in which I can add to, clarify or restate some of the points.

Q: Our organization uses Test Director by Mercury Interactive to track defects. My manager is concerned about reports that some



(Continued on page 5)

Questions from the Mail Bag (Continued from Page 4)

(Continued from page 4)

people are not using the tool to report defects. What are your thoughts on how to handle this situation?

A: My solution (not fully understanding all of your issues there) would be to place controls on the defect tracking process, assign a defect tracking administrator to be a person in charge of the tool usage and performance of the process of reporting defects, and not count or even consider work to be tested that has not gone through the process using the tool.

The only exceptions would be those you mentioned in your e-mail. In other words, tool use would be mandatory except on special exceptions. It sounds like this is what your management is expecting, which means that management should also be making

that message. If management is not willing to take such a stand, then they get what they create - a situation where tool usage is optional.

Q: Who should ideally produce the acceptance test plans - the user or the analyst/designer?

A: In my opinion, the user should create the acceptance test plans and in fact "own" or control the test. The users may need assistance and facilitation from other groups in this process, but the entire point of the exercise is to test the application from the user perspective. My experience is that the only people who can truly take that perspective is the user community.

Q1: I'm a fresh grad in Computer Studies and unfortunately I don't have much experience in organizing testing..... And by now quite a

number of problems have arisen....

(1) at first, my boss requires me to write about all possible test case/situations to simulate the UAT. (everything is based on functionality instead of the business role....)

A1: OK, first of all, that's an impossible task you've been given. As much as we would like to have a complete test of functionality, there is not enough time in your life and your kids' lives...and even your grandkids' lives to test all possibilities, even once. Second, the whole point of UAT is to test business processes and business scenarios from a user point of view. Period. All other testing should have been done prior to this. Plus, UAT should not be verifying compliance to requirements, but fitness for business use and user needs. In other words, "Are we delivering the RIGHT system?"



(2) since it's not practical, I've advised to change the focus to write out the UAT scenarios instead

A2: Good job.

(3) now the problem is, my boss is really keen on the UAT cases (just concentrate on functionality) but he is not satisfied with the UAT scenarios (concentrating on the business role), he is always trying to add more and more cases to the scenarios which in my view is not sensible.... (especially when he's asking me to add more critical cases such as server breakdowns or power failure during backup/normal process etc...)

(Continued on page 6)

Book Review—Managing Software Acquisition: Open Systems and COTS Products

(Continued from page 2)

aspects of COTS. Most of the testing topics mention that it needs to be done and gives some general comments, but lacks any specific guidance. I would have also liked to have seen a discussion of user requirements in addition to the other kinds of requirements mentioned.

Organization

This book consists of five main parts, which contain related chapters, and four appendixes.

Part One, Getting Started, consists of the first four chapters, which introduce the basic elements of open systems and the use of COTS products. These chapters present an overview of acquisition, describe the promises of the open, COTS-based approach, explore the paradigm shift to open systems and COTS products, and present the elements of an open, COTS-based approach.

Part Two, Understanding the New World, explores various aspects of open systems and COTS products. Chapters 5-8 look at reference models and architectures, standards, commercial off-the-shelf (COTS) products, and acquisition roadmaps.

Part Three, Managing the Transition, provides information to help you maneuver successfully in the world of open systems and COTS products. Chapters 9-12 consider how open systems and

COTS products can change your business, discuss special concerns for managers, describe engineering practices, and discuss procurement practices.

Part Four, Considering Acquisition, focuses on the acquisition context for open systems and the use of COTS products. Chapters 13-15 describe an acquisition framework used to describe various acquisition models, particularly acquisition models for open, COTS-based systems.

Part Five, Closing Thoughts, consists of one chapter, which looks at anticipated future acquisition issues, both general and specific to the government.

Other information is provided in the appendixes: a glossary of terms, a list of acronyms used in the book and what they mean, sample questions to help you analyze your system, and references.

Audience

- Senior and Executive Management
- Project Managers
- Methodologists and Process Designers
- QA Analysts

Topics and Outline

Pt. 1 Getting Started

1. An Overview of Acquisition.
The Project Manager Perspective.
Acquisition Strategies.
Looking Ahead.
Summary.
Food For Thought.

2. Promises and Pitfalls.
Key Definitions.
The Promises.
The Pitfalls.
Summary.
Food For Thought.

3. The Paradigm Shift of Open Systems and COTS Products.
Essence of the Paradigm Shift.
Consequences of the Paradigm Shift.
Marketplace Considerations.
Importance of the Interface.
Product Quality Characteristics.
The Loss of Control.
Implications for the Government.
Summary.
Food For Thought.

4. Elements of an Open, COTS-Based Approach.
An Overview of The Approach.
The Elements of the Approach.
Iteration.
Summary.
Food For Thought

(Continued on page 6)

Book Review—*Managing Software Acquisition: Open Systems and COTS Products*

(Continued from page 5)

Pt. 2 Understanding the New World

5. Reference Models and Architectures.

Abstraction.
Reference Models.
Architectures.
A Comparison of Reference Models and Architectures.
Trends.
Summary.
Food for Thought.

6. Standards.

What Is a Standard?
Standards Organizations and Their Processes.
Characteristics of Standards.
Standards Maturity.
Profiles.
Conformance.
Sources of More Information.
Standards in Government.
Summary.
Food For Thought.

7. Commercial Off-The-Shelf (COTS) Products.

Kinds of Off-the-Shelf Components.
Key Characteristics of COTS Products.
Deciding to Use COTS Products.
Negotiating Between Requirements and the Marketplace.
Buyer Beware.
Government Policy Implications.
Summary.
Food for Thought.

8. Acquisition Roadmaps.

A General Approach.
Open Systems Highway.
Upgrades.
The Open, COTS-Based Path.
Integration.
Summary.
Food For Thought.

Pt. 3 Managing the Transition

9. How Open Systems and COTS Products Can Change Your Business.

Kinds of Changes.
Potential Changes.
Summary.
Food for Thought.

10. Special Concerns for Managers.

The Manager's Quest for Control.
Cost.
Schedule.
Performance.
People.
Transition Strategies.
Summary.
Food for Thought.

11. Engineering Practices.

Determine Concepts, Requirements, and Reference Models.
Define Architectures, Components and Interfaces.
Select Standards.
Select Implementations.
Acquire Implementations.
Integrate and Test.
Deploy and Support.
Summary.
Food for Thought.

12. Procurement Practices.

Contracting Strategies.
Contracting Documentation.
The Role of the Players.
Government Contracting Concerns.
Summary.
Food for Thought.

Pt. 4 Considering Acquisition

13. An Acquisition Framework.

Defining A Framework.
Acquisition Activities.
Acquisition Events.
Relations Among Activities and Events.
Timing Considerations.
Framework Summary.
Acquisition Strategies.
Summary.
Food For Thought.

14. Acquisition Models.

Characterizing Acquisition Models.
Waterfall Model.
Refined Waterfall Model.
Spiral Acquisition Models.
Summary.
Food For Thought.

15. Acquisition Models for Open, COTS-Based Systems.

The Overall Context.
Standards.
COTS Products.
Integration of Standards and COTS Product Acquisition Elements.
Acquisition Model Considerations.
Management Implications.
Multi-Project Acquisition.
Summary.
Food For Thought.

Pt. 5 Closing Thoughts

16 Looking Ahead

Summary

If you or your organization are planning to embark on a COTS or open systems strategy, this is a good book to understand the territory.

Reviewed by Randall W. Rice

Q&A

(Continued from page 5)

A3: Ahh...then you need to separate the tests and assign them to different teams. Of course, if you are the one responsible to plan the test, then you only have time to plan it from one perspective. I am assuming its getting close to deployment time. I think what may be causing the lack of confidence in the scenarios alone is that perhaps detailed functional tests may not have been done yet. If this is the case, I've got some good news for you. You can combine business scenario testing with detailed functional testing, but its a little tricky. The danger is that you may get bogged down into detail and miss the larger view of testing transactions and scenarios.

I would take a top-down test design approach. Start by mapping the business processes and business cases at a high level and then drill down into the detailed functionality. It's the best of both worlds, but it is time consuming. To deal with time constraints, you will need to prioritize which business processes are the most critical and focus on them first, then move to those with moderate criticality, and finally, to those with lower criticality.

(4) so finally, I've got to re-think/re-organize the direction to continue doing this UAT, can you suggest a better way to compromise or organize UATs in my case? (ie. how to group the cases with the scenarios? Or how should the scenario's length and characteristics be?)

A4: Well, this is a question that requires some professional judgment to answer. I would really need to see what you are working with. To understand my perspective, keep in mind that I teach a process for designing UAT test scenarios that is fairly rigorous and is based on business cases.

I can tell you some of my guidelines for test scripts and test scenarios:

- Keep the complexity low. If you find that a process has more than 5 or 6 decisions, either break it up or make it more general.
- Keep the decisions out of the test scripts. Decisions add complexity and confusion.
- Keep the scripts modular. Instead of having long scripts with many functions in each script, break them up into modules and combine the like building blocks. I really don't have a recommended script length, but try to avoid going over 2 pages.

I have many other practical examples in my online training course. You can find details at:

http://www.riceconsulting.com/training/web-based/online_uat.htm

You can even take a free demo there.



September, 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

October 2002 Issue:

Getting the Most Value from Every
Person on Your Test Team, Part 2
by Randall W. Rice

Book Review — Net Privacy

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

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*.

www.testingconferences.com

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 rice@riceconsulting.com to find out how to book a special presentation.