

2009

PACIFIC NW SOFTWARE
QUALITY
CONFERENCE



MOVING
QUALITY
FORWARD

OCTOBER 27-28, 2009

Conference Paper Excerpt
From the

CONFERENCE
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

Quality Cost Management

Manage Your Quality Costs or Let Them Manage You

Abstract

Many of us QA professionals spend our entire careers managing defects and we're getting pretty good at it. But if defect management is our universe, we are not really *assuring quality*. We need to move away from finding, reproducing, documenting, prioritizing defects and validating fixes. We need to move to preventing defects.

Rising above the defects takes commitment to process improvement. What is stopping you, dear reader, from making serious software process improvements? My bet is that your answer is "Time." If you are a software quality professional, the only other valid reason is "Knowledge." Gaining process improvement knowledge is easy – great resources exist ... if only you had time.

This paper addresses the time issue. The answer is money. That's it. It's that simple and it's that hard. Getting money to improve processes require a convincing business argument. Executives understand money more than they understand software quality. Your task, should you choose to accept it, is to convince your management that preventing problems is more profitable than finding and fixing them. Only then will they fund your improvement efforts.

Here's your starting point: Proactively managing quality costs is a sounder business practice than reacting to product failures.

This paper will provide the theoretical foundation for actively managing quality costs to increase profits and reduce chaos. My goal is to help you convince your executives to invest in defect avoidance... with dedicated resources driven by data. Quality cost management, based of Crosby's cost-of-quality foundation, means reducing overall quality costs by eliminating sources of waste and failure.

We need not swim in defects forever. There is a better way.

Biography



Ian Savage

A quality/productivity practitioner and evangelist, Ian is a veteran software developer, quality assurance engineer, manager and executive with experience in the high tech manufacturing, financial services, construction services, and security software domains.

Ian learned about quality at Tektronix and has served PNSQC in several capacities. He serves on the boards of the Software Association of Oregon, the SAO QASIG, the Tech Alliance: Central Oregon's Technical Association (SAO chapter), and PNSQC in various roles.

Trained as a process engineer, he has contributed to the Agile Open Northwest, PNSQC, and ASQ's Certified Software Quality Engineering program - becoming one of ASQ's first CSQEs. As an ACM and Agile Alliance member, his current interests include applying best practices – including appropriate agile methods – at McAfee, Inc.

What is “Cost of Quality” and Why You Should Care

What is CoQ?

Cost of Quality (CoQ) is a framework that provides answers to these questions:

- How much does poor quality cost us?
- How much are we investing in quality?

The answers to those two questions allow us to express quality in terms that executives immediately understand: profit and loss. Once engaged in that conversation, executives view quality differently – as a strategic business driver.

In short, CoQ shows how much money is spent on quality; it is sound both in theory and in practice,¹ it requires quality cost assessments,² and it moves people to action. The total cost of quality (TCoQ) equals the costs of assuring quality (**quality system costs**) plus the costs of responding to poor quality (**failure costs**).

Definition: $TCoQ = \sum (\text{quality system costs} + \text{failure costs})$

In stark contrast to software quality costs, manufacturing costs are extremely low and that is reflected in low prices of hardware. Software companies are decades behind in adopting the techniques that enable low prices. True, software development is different from hardware manufacturing³ but CoQ measurement is applicable to the software business – without modifications.

What is QPIP?

Throughout this paper you will see references to “QPIP.” It is shorthand for “quality and process improvement projects.” It reflects the strong link between good processes and high quality.⁴

Managing quality costs requires combining CoQ with a QPIP program. CoQ quantifies your improvement efforts: It is difficult to point at bugs that you prevented, but it is easy to point at a TCoQ curve that is heading in the right direction.⁵ See Figure 1.

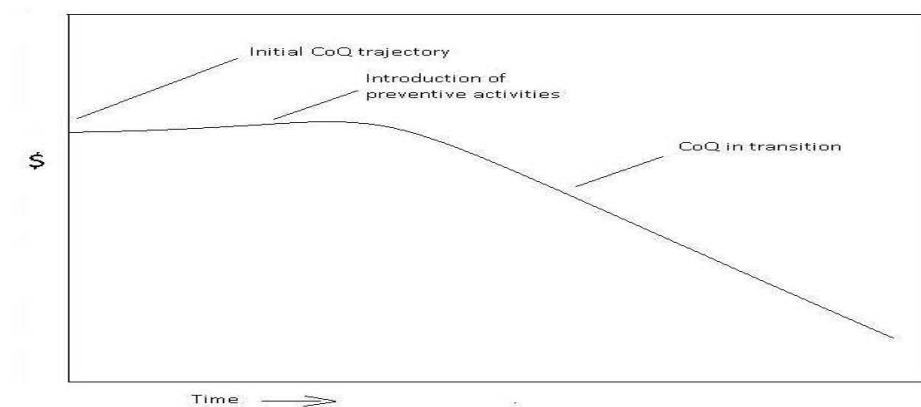


Figure 1: Over time, prevention yields lower Total CoQ

¹ See Jack Campanella’s “Principles of Quality Costs”, 3rd Edition, 1999, ASQ Quality Press.

² Several high-profile companies actively track quality-related time and expenses with CoQ Systems. See Part III.

³ Herb Krasner describes the salient difference elegantly in his 1998 Crosstalk article “Using the Cost of Quality Approach for Software.” Software, because it is soft, can more readily be modified – it’s malleable.

⁴ The more common term “software process improvement” (SPI) omits that link – and without it, executives can view process improvement as an easily axed academic exercise - not a strategic driver. To have political capital, you must be part of the solution.

⁵ If you don’t track TCoQ, you will find yourself relying on anecdotal evidence to justify your improvement efforts.

Here's the first pitch for your executives:

This is the value proposition of quality:

Strategic: World-class products and services are profitable and worthwhile

Operational: Failure prevention methods are easy to use and effective

Capture this in your elevator talk.

The goal of a proactive CoQ program is to minimize failures by using industrial-strength preventive techniques.⁶ The alternative to actively managing your quality costs is reacting to failures after they occur, enduring chronic pain and constantly playing catch-up. You can either manage your quality costs by investing in prevention or hope for the best, triage incoming problems, and get further behind.

Why You Should Care

Quality Analysts, Process Engineers, and Testers: Are you doing things of little or no value? Why? Can you tell your manager? Do you always ask yourself, "How could we have caught that problem earlier?" or do you wait for others to ask that? Do you always ask, "How could we prevent that sort of thing from happening again?" Can you take actions based on your answers?

Quality Managers: How effective are you? How effective is your team? How do you know? Are you leaving "gold in the mine"? How much time of your team's time is simply wasted?

Whatever your role in software quality, you can move quality forward in your organization by actively managing your quality costs. But it is not free. Indeed, your TCoQ will *increase* before it decreases but then it can decrease dramatically – and that's the goal for you and your executives.

The software industry is leaving gold in the mine.⁷ With a few notable exceptions, we have the opportunity to greatly increase profits by adopting the lessons from other industries. In his must-read book "Quality is Free" [Crosby79] Crosby says that the TCoQ is *at least* 20% for companies that do not track CoQ. My experience is that TCoQ in software companies is roughly twice that of manufacturing companies and Krasner's excellent survey supports that belief. In [Krasner99], he concludes that 10-70% of the global software development budget is spent on quality.

Here's the next pitch to the execs: **As you address root causes by proactively managing your quality costs, you...**

- **eliminate entire families of defects,**
- **improve your products and services,**
- **increase your organization's profit and market share, and**
- **protect your job and many others.**

Minimizing mistakes in your sphere of influence helps your managers succeed. Reducing costs responsibly also helps our industry serve society better through lower prices, higher reliability, and better performance.

QPIP is also good for you professionally. The nature and impact of your work changes: It becomes productive and predictable, routine and dynamic, fun and scary. You have more interpersonal interactions but fewer interruptions. You collaborate often and feel a strong dedication to cause. You spend less amounts of time fighting fires and no time covering your tail. As you reach for the brass ring – world class quality – you spend less time hassling with bugs and more time on value-added activities, you gain experience with high-ROI projects and your influence and marketability grow. Testers may find careers evolving into process engineering, auditing, management, or quality systems analysis. Managers may

⁶ This is also the essence of lean manufacturing. See RAND article: http://www.rand.org/pubs/research_briefs/RB80/index1.html.

⁷ Dr. Joseph Juran used this phrase to sell CoQ to management in his Quality Control Handbook, McGraw Hill, 1951. Other authors refer to the unseen and underreported costs as the "hidden factory".

find higher visibility and influence in the organization and quicker promotions. All will find work more enjoyable.

There will always be some need for manual functional testers but as the software quality profession is following the same trajectory as other industries (notably manufacturing, community policing, and health care), fewer testers will be needed. Quality professionals find our roles and responsibilities change as our quality organizations mature and their missions evolve (as proposed in Table 1).⁸

Growth Phase	Level	Primary Activity/Mission
initial	0	Help make the software work
	1	Break the software (find bugs)
awakening	2	Assure feature functionality / Assess functionality
	3	Assure software attributes / Assess quality
maturity	4	Improve processes / Prevent problems
	5	Audit processes

Table 1: Software Quality Organization Maturity Levels.

As you become more immersed in QPIP, your organization matures.

How to Manage Quality Costs

This section addresses the question, “Okay, how am I going to do it?” The quest is to find time. The quest starts with a charter for your active CoQ and QPIP efforts. Here’s a sample, generic QPIP charter:

Maximize ROI on available quality resources by using preventive measures to eliminate entire families of defects and waste.

Next, familiarize yourself with Krasner’s research into ROI on process improvement efforts.⁹ Any effort that returns a 20-30% premium is worthwhile but efforts that return 200% or 1000% are slam dunks. You will have executive attention.

Convincing management to undertake serious quality improvement requires learning some jargon. The next section introduces the main terms you’ll need.

Definitions: System and Failure Costs

Many software activities are directly related to quality. Active CoQ addresses those quality-related activities (see the Appendix for examples). Quality-related costs and activities are normally divided into two broad categories: *System* and *Failure* which both have two subcategories as shown below and in the Appendix.

System costs ~ Those things you choose to do to prevent defects or detect them.

Prevention ~ everything associated with avoiding failures and waste.

Appraisal ~ everything associated with test and measurement.

Failure costs ~ Those things you must do to recover from creating or shipping defects:

Internal failures ~ defects found *internally* (before shipping)

External failures ~ defects found *externally* (by customers or others after shipping)

With those basic definitions, we can move on to our improvement strategy.

⁸ While much is written about organizational maturity and its link to software quality, the maturity of quality organizations is largely absent from the literature other than the current paper and <http://www.aof.mod.uk/aofcontent/tactical/quality/downloads/gmmm.doc>.

⁹ See Krasner’s 2001 report <http://www.compaid.com/cailInternet/casestudies/krasner-spiROI.pdf> for ROI data in the public domain.

General CoQ Improvement Strategy

The necessary and sufficient tasks to manage quality costs, then, are these:

1. Get support – find a champion
2. Assess your major quality activities and other quality costs
3. Propose/Select improvement projects
4. Execute and monitor
5. Repeat and institutionalize

Note: Size matters. Small organizations can just do it – review quality successes and failures at each planning meeting – daily, sprint/scrum/iteration, and release – to take corrective actions. Larger organizations with committed resources can also transform into a world-class¹⁰ software shops by managing quality costs but the journey is necessarily more structured and formal. The rest of this section focuses on those larger shops.

Get Support – Find a Champion

First, you need support for making quality costs visible and for implementing corrective actions. So find a senior executive responsible for allocating money to support the effort. She needs to represent your efforts to other executives and defend your efforts over time. You'll need a great elevator talk and a perhaps a presentation about eliminating families of defects and increasing market share through better quality. It may be a tough sell: when quality costs are high, research funds are slim.

If you can break through her expectation of QA to just to find bugs, she may invest in defect prevention. But a certain organizational maturity is required to embark on a quality improvement project even when cost avoidance is the goal. Gaining a champion can be tough but once you clear this hurdle, you are well on your way to continuous improvement, higher profitability, and better days.

The first thing you need to sell to management is that you feel strongly that you can help her save tons of money. All you want initially is to conduct a CoQ survey to show her how much your organization is spending on quality. You can tell her later that you want to do those surveys repeatedly.

Assess Your Major Quality Costs

There are two main ways of determining your quality costs:

- 1) A series of surveys or
- 2) A more formal tracking system.¹¹

Warning: Your tracking system cannot become part of the problem. It is not a formal accounting system. It needs to be lightweight and show the quality costs in each category. Because quality improvements result in large improvements, your CoQ system can paint broad strokes.

Here are two low-cost options for the initial CoQ survey:

- Zoomerang
 - Pros: quick and cheap
 - Cons: spotty results, relies on recollections, responses may not be statistically significant
- Interviews and Excel
 - Pros: more detailed, better consistency
 - Cons: labor intensive so more expensive

¹⁰ Reviewers have suggested that “world-class” is ambiguous. For this paper, it equates to Crosby’s “Quality Certainty” stage and the equivalent. For background information, see:

- [Crosby's Stage V - Certainty](http://en.wikipedia.org/wiki/Quality_Management_Maturity_Grid): http://en.wikipedia.org/wiki/Quality_Management_Maturity_Grid
- [Bhote's Ultimate Six Sigma](http://www.amacombooks.org/book.cfm?isbn=9780814407592): <http://www.amacombooks.org/book.cfm?isbn=9780814407592>
- [Zero Defects](http://www.managers-net.com/zerodefects.html): <http://www.managers-net.com/zerodefects.html>
- [TQM](http://en.wikipedia.org/wiki/Total_Quality_Management): http://en.wikipedia.org/wiki/Total_Quality_Management
- [Project Management for Adults](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=589226): http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=589226

¹¹ Such as an Activity-Based Costing (http://en.wikipedia.org/wiki/Activity-based_costing)

Cost management systems are more rigorous and statistically defensible but they are more expensive. Put such a system on your wish-list for later – after your QPIP efforts have shown their worth.

Your first CoQ survey could use Zoomerang and ask just four questions of randomly selected employees:

1. How many hours did you spend last week on field failures and customer complaints?
2. How many hours did you spend last week diagnosing and fixing bugs?
3. How many hours did you spend last week doing testing and evaluations of any kind?
4. How many hours did you spend last week preventing problems from reoccurring?

Warning: You may get questions about how you intend to use the data. Assure the surveyed people that the survey results will be free from all identifying information. You can also expect widely divergent definitions of “field failures” and other terms. Include some examples in your survey.

Your results may vary but do not be surprised if the results look something like those in Figure 2.

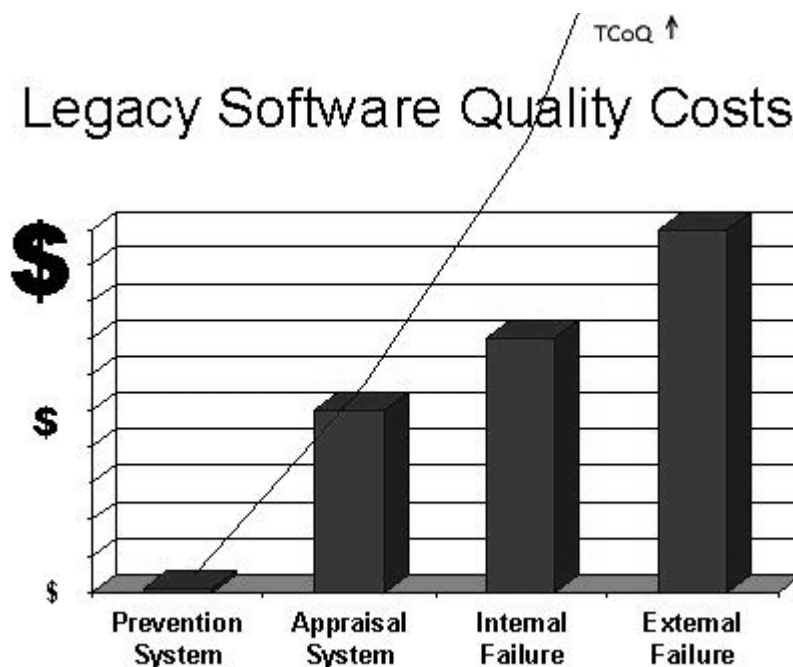


Figure 2: Typical legacy software quality costs – note that TCoQ is off the chart

Propose/Select Improvement Projects

The next thing is to select an improvement target. Your people usually know what changes can lead to big improvements. Your bug tracking database can be extremely useful in finding opportunities and patterns of failure. Running a few queries uncovers low-hanging fruit. Since bugs tend to cluster, you may identify components that need refactoring.

Warning: At the beginning, there will be lots of ideas and consensus may be hard to reach. Stick with it. Like any other team a QPIP team must form, storm, and norm before it can really perform.¹² Focus on those things that are causing the most pain for your customers and other stakeholders. Do not try to boil the ocean. You cannot address every concern at once nor find a solution that will fix all problems. Start

¹² Attributed to Bruce Tuckman, 1965: <http://en.wikipedia.org/wiki/Forming-storming-norming-performing>

with important issues and keep it simple. You need to show early successes, collect organizational capital, not cure everything in one go. QPIP requires continuously honing ones soft skills.

You are looking for your biggest problems for which you can take corrective actions and generalize the solutions. Let's take those in order...

Identify your biggest problems

Your champion can be an excellent source for improvement ideas. Involving her has the additional benefits of aligning your activities with company goals, keeping the discourse in management terms (cost avoidance), and keeping senior management in the loop.

Sometimes the target is painfully clear, for instance when your company has suffered an embarrassment in the market. But more often you will have multiple improvement options. In those cases, this is a useful QIP planning method:

1. Brainstorm the important problems – everyone contributes at least one “if only X, we could have Y”,
2. Affinity grouping – wherein like items are grouped together, and
3. Perform a Pareto¹³ analysis – wherein you converge on the “significant few” problems.

Two warnings when discussing patterns of failure:

1. Studiously avoid talking about people. As Deming taught us, 85% of root causes are process-related and only 15% are people-related. Direct the discussion toward process improvement even if everyone knows that Joe needs training.
2. Use the term “failure” infrequently. In a supportive, non-threatening way have the team suggest *improvement ideas*. That's much less threatening than talking about patterns of failure. Make it clear to everyone that you are not looking for scapegoats. Instead you are looking to refine (or implement) effective processes.

Take corrective actions

Once you have selected the one or two most important problems, dig for root causes. Ask this Deming-inspired question “What part of the process or system allowed this problem to happen?” Answering that question allows you to make process improvements that eliminate entire families of errors.

The term “corrective action” means effectively remediating root causes. Too often people feel that they have taken corrective action when they have found a design or coding fault. But without addressing the causal factors, other instances of that same fault will occur – and no long-term benefits are gained.

Figure 3 shows one useful framework for brainstorming possible root causes.

¹³ Also known as the 80/20 rule. See http://en.wikipedia.org/wiki/Pareto_principle.

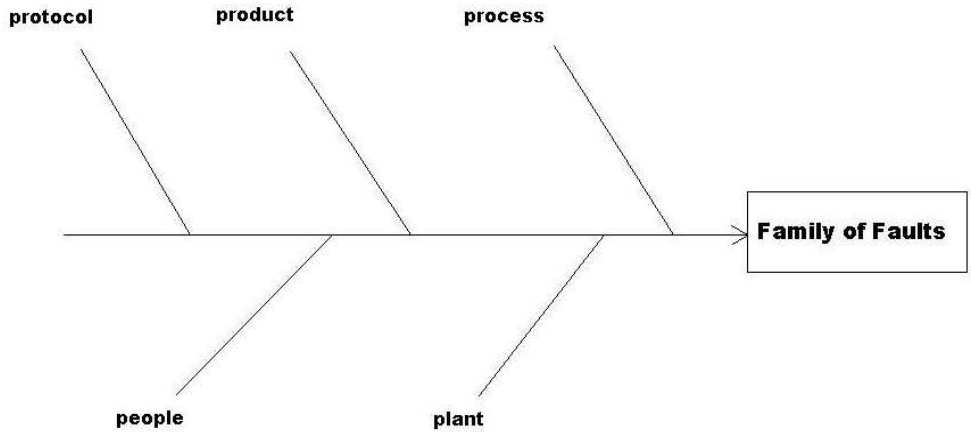


Figure 3: A generic cause-and-effect diagram (aka Ishikawa or fishbone diagram).

Be sensitive to the differences between symptoms, problems, causes, and solutions. In IEEE terms:

- Symptoms are FAILURES. These are the effects manifested during product use.
- Problems are FAULTS. These are the underlying issues with the code or the design.

Causes are those things that allow the faults to occur. These root causes are the real targets of CoQ/QPIP. Identifying the most influential root causes is vital. Labeling each cause as *necessary* or *sufficient* allows you to rank which causes to address first. Careful selection of the causal factors leads to the best ROI.

Note: You cannot address *every* causal factor. For instance, some causal factors may be outside of your sphere of influence others may require more resources than readily available. Using the S.M.A.R.T. goal-setting technique will help. To increase your effectiveness your atomic CoQ/QPIP goals, like all good goals, should be **s**pecific, **m**easurable, **a**ttainable, **r**elevant, and **t**ime-boxed.

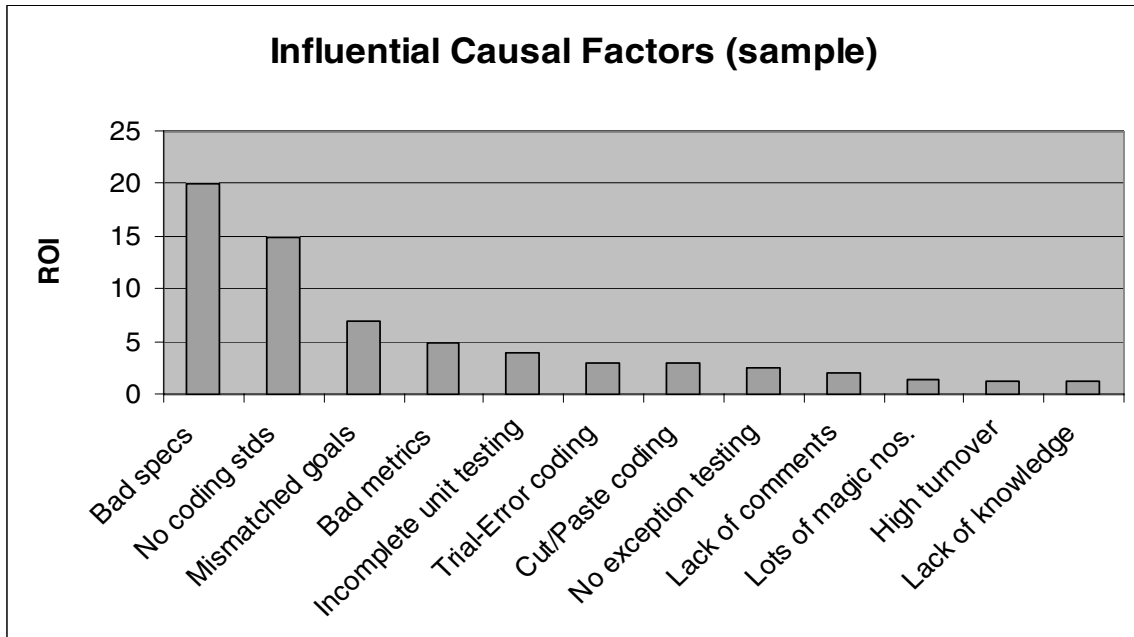


Figure 4: A sample Pareto chart of causal factors

Generalize solutions

Okay, you have identified and ranked your problems, determined the most influential causes, and created a corrective action implementation plan. But you're not done. Just as *faults* can cause multiple *failures*, *solutions* can solve multiple *root causes*.

Before you present your plan to your management champion, consider whether your proposed corrective actions will have multiple benefits. That is, will you get even better ROI than documented in your plan? If so, you have two options: 1) a simple notation in your plan that other beneficial side effects are probable or 2) revisit and expand your plan. This latter option takes you back to the affinity grouping step (above).

Generalizing the solutions helps you and your sponsor keep the overarching organizational goals in focus and further fund your CoQ/QPIP efforts.

Incidentally, the time you spend discussing CoQ/QPIP with your champion, the time you spend constructing and administering the surveys, the time you spend ranking problems and solutions, and the time you spend reading this paper – those are all Prevention costs. With respect to the late Phil Crosby, quality is not free: Managing quality costs is an expense but the payback can be huge.¹⁴

Execute and Monitor

Nothing will kill a QPIP quicker than lack of follow-through. Collaborate with your champion to assure that the action plans you design are well-received in your organization. Train your people how to respond to questions about the CoQ/QPIP. Keep all stakeholders fully apprised and involve as many of them as possible. Definitely involve the affected people when defining and implementing a new process.

A monitoring mechanism must be part of your action plan. Once your changes are implemented, stay in close touch with the affected people. Be prepared to change your solution. Be prepared to start over.

For you as a quality professional, once you've implemented a few process improvements, 1) you will be an effective change agent with corresponding growth in influence and 2) your negotiation and facilitation skills will be improving.

Repeat and Institutionalize

A successful CoQ/QPIP program is a journey that never ends. But one major milestone is met when QPIP is institutionalized... when it becomes part of the organization's DNA.

One way for large organizations to close the loop – or “ratchet” as Juran called it – is to establish a quality steering team. Your quality champion can lead that team but it also must include your most-senior executive. Their mission is to integrate quality/process improvement into every employee's work. They fund and focus QPIP teams and give the CoQ/QPIP program legitimacy. The very existence of a top-level quality steering team tells the world “we are serious about quality.”

One successful model is that the steering team meets regularly with smaller action teams (2-4 members), empowers them and holds them responsible for continuous improvement in certain domains such as the Capability Maturity Model's Key Process Areas.

Implementing Quality Circles is another way to institutionalize quality improvement. This, though, is a major undertaking that can only succeed when the entire organization appreciates the value of QPIP. It is not a year one objective in large software organizations.

¹⁴ The claim that “quality is free” arises from Phil Crosby's recognition that defect prevention pays for itself many times over.

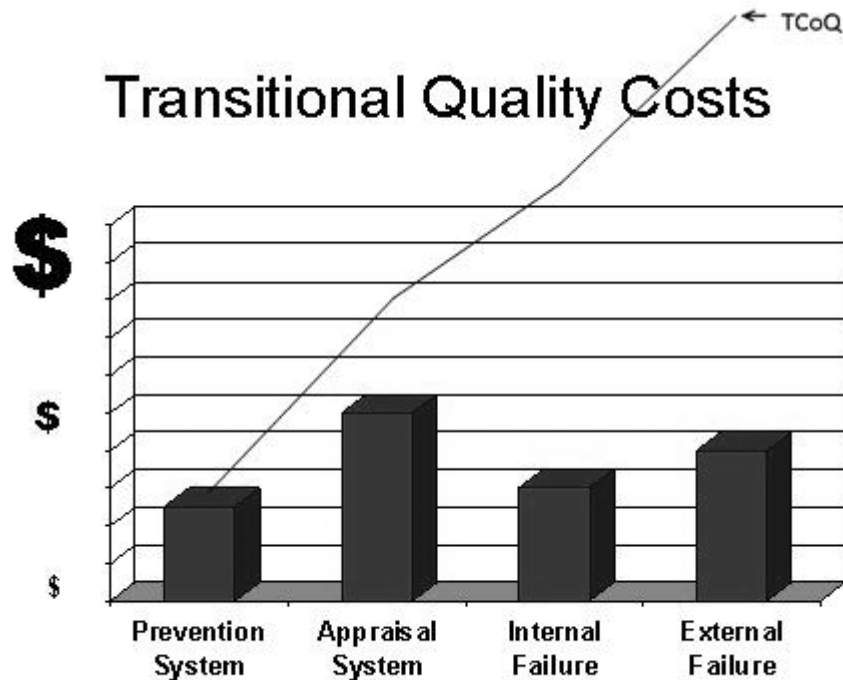


Figure 5: Possible quality cost profile after one year – note TCoQ

Let us fast forward one year beyond your initial TCoQ survey: Your executives are seeing that QPIP is profitable. The quality cost profile has changed considerably (Figure 5), people are happier, schedules are more predictable, and estimates are better. Products are selling better and profit margins are improving. The testing department is no longer frantic and they are now working 40 hour weeks. TCoQ reports are now reviewed at every board meeting. The company is awakening to the strategic value of quality.

You might think that finding good improvement projects would get harder after one year. But it doesn't for two reasons:

- 1) The environment keeps changing - processes don't fit forever, the competition doesn't stand still.
- 2) Your culture is changing. With quality as a strategic driver, people are seeing real improvements in their operations, they are re-invigorated and engaged. This generates new improvement ideas.

Learn, Practice and Preach

Beat the drum. As you get feedback about your process improvements, share that with the stakeholders. Let them know how much money you have saved the organization. That builds your political capital and makes further improvements easy to justify. At this stage of the software industry's evolution, the consummate quality professional is an educator and a practitioner.

Your organization's new focus on preventive actions allows you to accelerate the improvements. What does it really mean to invest in prevention? It means:

- Training: read books, go to conferences, lead discussion groups, mentor one another
- Product planning: involve customers in product evolution including end-of-life planning
- Quality planning: expand testing to include all quality attributes, establish tradeoff protocols
- Documenting: move away from tribal knowledge and its reliance on warriors and heroic efforts
- Analyzing failures: use each failure as a learning experience to improve processes

Push the Envelope

Keep running... Even the Toyota Way is constantly changing. Read, do experiments, collaborate, develop and apply new ideas. Success breeds success. Talk to your internal and external customers and suppliers. Know their needs and methods. Think in terms of systems and processes. Talk to your sponsor about sharing the wealth. At least one company shares the cost savings accrued during the first year with the responsible contributors.¹⁵ Monetary incentives aid the institutionalization and keep the focus on money – the language of management.

After you spend a couple years seriously removing root causes of failures, you don't have as many product failures. The code is cleaner. In the Bach brothers' parlance – you will have more T and less BS (more *testing* and less hassling with *bugs* and *setup*)¹⁶. Do you need to test your software as much? This is a non-trivial question. Would reductions in testing introduce too much risk?

The manufacturing world moved to small sample sizes long ago. A modern high tech manufacturing cost profile looks like Figure 6. Krasner¹⁷ tells me that General Dynamics Defense Systems, Motorola Cellular Systems, Qualcomm, Honeywell Building Control Systems, and Xerox Office Products have thriving CoQ programs. How long will it take us software people to catch up with the hardware people?

World Class Quality Costs

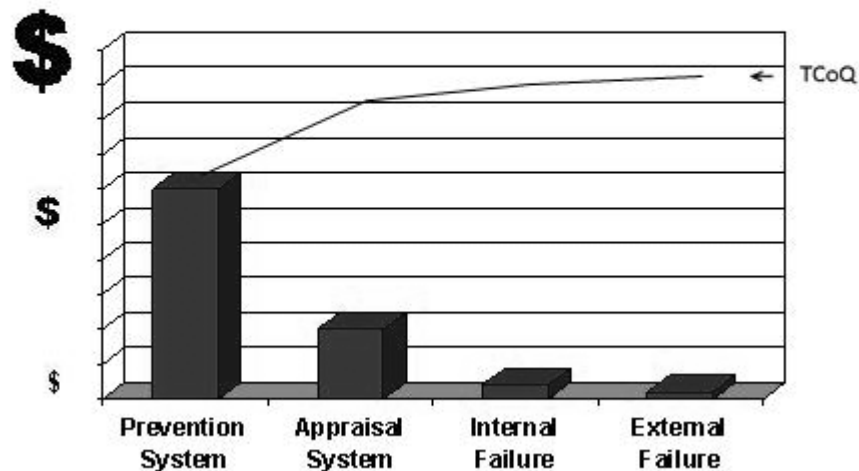


Figure 6: Quality system costs far exceeding failure costs – note TCoQ

¹⁵ This is another cost of quality improvement – another Prevention cost – one that the company is happy about.

¹⁶ Presented by Jonathan Bach at a Software Association of Oregon event, Wilsonville, 2007.

¹⁷ Private communication June 2009. See also "Accumulating the Body of Evidence for the Payoff of Software Process Improvement": <http://www.compaid.com/cailInternet/casestudies/krasner-spiROI.pdf>

Conclusion

You will find time for quality/process improvements once you have started the QPIP journey.

Many quality professionals are one executive decision away from tough times. But layoffs are avoidable for two reasons:

- 1) Companies that prevent problems prosper.
- 2) Companies that allow problems to reoccur are fertile grounds for QPIP.

In most software organizations, radical improvements are possible and you can contribute in ways large and small. But the improvements must be visible and CoQ tracking is an excellent mechanism for that.

Our times are marked by economic upheaval. That turmoil is not entirely outside the scope of this paper. Some current issues are definitely quality related – Ford, Chrysler and GM are playing quality catch-up to Japan's car companies. It's extremely expensive for them.

Like high tech quality, **software quality is not a matter of luck**: When you don't prevent recurrences, you are bound to see the same mistakes. That is a waste of time and talent. And if your organization wastes too much time dealing with avoidable problems, eventually it is forced into cost-cutting measures (e.g. layoffs). "Cost cutting" should have a different connotation – profitability improvement through defect prevention.

In my group, that means further reducing our incoming defect density by using more agile development methods. We have sufficient autonomy – and have made sufficient progress – to take some risks. In short, we are in the process of moving beyond being a test organization and becoming a quality organization. Just like PNSQC is a quality conference – not just a testing conference.

You Can Choose Your Future

We will always need testers
...but not as many
We always need improvement.

This isn't some esoteric or academic theory. You have practical choices to make. You can proactively manage your quality costs on a micro level. And you can influence quality costs at a macro level. When you find a good bug or when you let a bug escape to the customers, you have a chance to significantly improve your products and processes. You can collaborate with your colleagues to determine root causes and act to address them. Or you can duck; keep your head down, and hope that nobody traces the problem back to you.

If you choose the low road, you will find that you and your companies never really prosper.

If you choose the high road, you embark on a challenging, rewarding and exhilarating journey into group dynamics, organization maturation, and sustained improvements. Prevention as a way of life... When that becomes your mantra, you, your colleagues, and your company will win big.

Toyota has chosen the high road. They *want* people to find problems. They literally applaud people who stop the production line. They train people to find root causes. They continually evolve and they now sell more cars than anyone else. It is time for the software industry to catch up.

Acknowledgements

The author would like to thank these people for their thoughtful reviews and/or other feedback: Cynthia Gens and Jonathan Morris – my official PNSQC Reviewers; Jim Fudge, Patt Thomasson, Josh Eisenberg, Ellen Ghiselli, Patricia Lee, Justin Kelly, and Robert Dufur – McAfee colleagues and friends. Thanks to all for your valuable feedback.

Appendix – Some Quality-related Software Activities

<i>Quality Systems</i>		<i>Failures</i>	
Prevention System	Appraisal System	Internal Failure	External Failure
training reading seminars conferences formal education failure analysis corrective actions quality improvement process improvement quality planning product planning documenting commenting code writing process docs joint specs definition QPIP capacity planning	test planning acceptance testing verifications validations test planning inspections design reviews code reviews assessments audits unit testing	technical debt code corrections design corrections unexpected refactoring waste unproductive testing abandoned efforts ignored plans debugging added operations rework: re-planning re-designing re-verifying re-validating re-inspecting lost time unneeded testing waiting for meetings waiting for hardware cancelled projects	technical debt field failure phone support triage defect management patches and hot fixes lost market share firefighting damage control lost business bad press litigation / settlements returned products

The four quality cost categories – Prevention, Appraisal, Internal Failure, and External Failure – are sufficiently distinct and you can determine the quality category to which an activity belongs by asking: “What is the **primary reason** for doing this?”

For instance, the reason for a refactoring task may be to eliminate structural weaknesses and, if so, it is a prevention cost.

If the same refactoring is done to correct several reported bugs, it is a failure cost. If the bugs were reported by customers, it is external failure; if reported by a testing group, internal failure.

If the same refactoring is done to abstract useful classes, it is a normal development cost and not something to include in the TCoQ.

If you are extending your test harness by refactoring locally-written library routines, it is an appraisal cost because the primary reason for doing it is to test the customer-deliverable software.

But don't get too caught up with categorizing your quality costs. Remember that CoQ tracking is a broad brush approach – it is not an accounting system that needs to zero-out to the penny.

Also, many activities have multiple quality-related reasons but, again, don't get bogged down in arguing minutia. Doing so takes energy from the goal of reducing overall quality costs.

Bibliography

Crosby, Phil; 1979, Quality is Free, Penguin Books

Juran, Joseph; 2005, Juran's Quality Handbook – Fifth Edition, McGraw-Hill

Kaner, Cem; 1996, "Quality Costs Analysis: Benefits and Risks" <http://www.kaner.com/qualcost.htm>

Krasner, Herb: 1999, "Using the Cost of Quality Approach for Software",
http://sunset.usc.edu/cse/pub/event/archives/pdfs/Herb_pt2.pdf

Krasner, Herb: 2001, "Accumulating the Body of Evidence for the Payoff of Software Process Improvement" <http://www.compaid.com/caiInternet/casestudies/krasner-spiROI.pdf>