

# Reducing Your Cost of Quality

By Alan S. Koch, PMP

How high is your Cost of Quality? The answer might surprise you. Yes, it includes reviews, the QA infrastructure and preparing tests, those are your "Appraisal Costs". But how high are your "Failure Costs" the cost of defects?

Your engineers spend time in diagnosis and rework, development schedules slip, support costs climb, and your company's and products' reputations sink. These Failure Costs, which are the more significant Cost of Quality, are beyond your direct control. But you can gain control over them indirectly, by investing in Appraisal Costs that minimise Failure Costs, reducing your total Cost of Quality and making it more predictable



## Cost of Quality: Appraisal vs. Failure Costs

The Cost of Quality is a significant cost on any project, so prudent managers look for ways to keep those costs in check. The Quality costs we can control are things like performing reviews, preparing tests, and maintaining our QA infrastructure, Appraisal Costs. But there are also the Quality costs we cannot control.

Failure Costs are the ones that happen to us. We incur these "Costs of Poor Quality" every time a defect comes to light, both during testing and after release. Failure costs take many forms:

- The effort that our developers spend investigating and diagnosing defects, and then reworking designs and code to correct them.
- Slips in our schedules, as testing uncovers defects that require rework and re-testing.
- Our customer support costs, most of which are for helping customers to deal with all of the defects we shipped to them, while developers spend even more time in investigation and rework.
- But, the biggest Failure costs are nearly impossible to quantify; loss of customer good will, tarnished reputation in the market, and loss of product momentum.

Since some components of the Cost of Quality are under our direct control and others are not, it seems to make sense to reduce those costs that we can, and hope for the best with those that we cannot control. Unfortunately, a focus on reducing Appraisal costs can increase our total Cost of Quality, because it is likely to result in an even larger increase in Failure costs.

As reported consistently in our industry, Failure Costs rise exponentially as the project progresses. Reducing Appraisal activities delays the detection of defects, ensuring that they are much more expensive to address when they are detected.

## Leveraging Appraisal to Reduce Failure Costs

Most organisations depend upon the compiler and various types of testing to remove most or all of the defects from their products. But as we can see from Figure 1, these are not the most effective methods of removing defects. They each tend to detect no more than 50% of the defects in the product, and often do much worse than that. In addition, they happen late in the project life cycle, when defects are the most expensive to fix. These activities are classified as 'Failure Cost of Quality' because the vast majority of the time is spent dealing with failures.

<b>Activity</b>	<b>Cost of Quality</b>	<b>Effectiveness</b>
Structured Personal Reviews	Appraisal	-----
(Fagan) Software Inspections	Appraisal	-----
Informal Peer Review	Appraisal	-----
Compiling	Failure	-----
Unit Testing	Failure	-----
Integration	Failure	-----
Beta Testing	Failure	-----
System Testing (and performance & other testing)	Failure	-----
Acceptance Testing	Failure	-----
Walkthroughs	Appraisal	-

*Figure 1: Appraisal vs. Failure Activities*

Contrast this with the various kinds of Reviews and Inspections. They are relatively more effective, not only because they can detect 60-80% of the defects in the product, but also because those defects are detected earlier, when they cost much less to correct. They are classified as 'Appraisal Cost of Quality' because only a small proportion of the time is spent responding to failures. So Appraisal activities tend to remove many more defects for each engineer-hour spent than do the Failure activities.

## Notes on Figure 1

The 'Cost of Quality' column indicates whether the majority of the time in that activity is spent appraising or dealing with failures. (It is important to realise that only the first compile is not failure-related. If there were no defects, we would have to run the compiler only once. By the same token, only the first run of any test is not failure-related. All diagnosis, rework and re-testing is necessitated by failures.)

The 'Effectiveness' column indicates both the percentage of defects that are likely to be detected and the total cost in engineer-hours to detect, diagnose, and remove each defect.

The placement of Reviews and Inspections in the list is based on best practices. In some cases, their effectiveness is quite low.

The order of Compiling, Unit Testing, Integration, Beta Testing, System Testing and Acceptance Testing in the above list really does indicate their relative efficiency in removing defects (not just their life cycle order).

The placement of Unit Testing in the list is based on best practices. Many developers have never been trained in testing, and are ineffective at it.

Walkthroughs are more effective for training purposes than for defect removal.

These economics point us toward the principle of leveraging Appraisal Costs (the ones we directly control) in order to reduce Failure Costs (the ones that are less controllable). And this principle leads us to the counter-intuitive proposition that if we wish to achieve dramatic reductions in our total Cost of Quality, we must increase Appraisal Costs dramatically. Does this proposition really work?

Consider that all ALL of our Failure costs (every dollar of them) are caused by a finite number of defects in our software. Every defect that we can remove more economically than we currently do represents

money on our companies' bottom lines. Every defect we can remove in a more timely way represents hours or days (or weeks!) of schedule saved. Every defect that we avoid shipping to our customers reduces support costs, and every useful feature that we do ship is priceless good will that builds our companies' reputations and market share.

Reviews and Inspections are the most economical way to detect and remove defects. Testing is a relatively less effective way to remove defects, but it is still a necessary part of our development life cycle. Rather than continuing to make it our main defect removal mechanism, we would do better to use it to verify the effectiveness of our earlier primary defect removal activities: reviews and inspections

## **Being in Control**

The only way to be in control of our total Cost of Quality is to shift it from the uncontrollable Failure Costs to the controllable Appraisal Costs. With each incremental increase in Appraisal activities like reviews (assuming they are done well), we can expect a corresponding and larger reduction in our Failure activities.

If we carry this principle to its logical conclusion, we will find ourselves in the enviable position of having shifted the majority of our Cost of Quality to the Appraisal side of the equation! This will mean that our Cost of Quality will not only be reduced significantly, but it will also be more predictable and more manageable. Instead of happening to us, our Quality Costs will be a tool that we can wield to control our projects and assure their success.

---

*Alan S. Koch, PMP, author of Agile Software Development: Evaluating the Methods for your Organisation, speaks, writes, and consults on effective Project Management. Visit <http://www.ASKProcess.com> to learn how to improve the return on your software investment by focusing on the quality of both your software products and the processes you use to development them.*