

# Quality Cost Analysis: Benefits and Risks

Copyright © Cem Kaner  
January, 1996  
All rights reserved

“Because the main language of [corporate management] was money, there emerged the concept of studying quality-related costs as a means of communication between the quality staff departments and the company managers.”<sup>1</sup>

Joseph Juran, one of the world’s leading quality theorists, has been advocating the analysis of quality-related costs since 1951. Feigenbaum made it one of the core ideas underlying the Total Quality Management movement.<sup>2</sup> It is a tremendously powerful tool for software quality, as it is for product quality in general.

## What is Quality Cost Analysis?

**Quality costs** are the costs associated with preventing, finding, and correcting defective work. These costs are huge, running at 20% - 40% of sales.<sup>3</sup> Many of these costs can be significantly reduced or completely avoided. One of the key functions of a Quality Engineer is the reduction of the total cost of quality associated with a product.

Here are six useful definitions, as applied to software products. Figure 1 gives examples of the types of cost. Most of Figure 1’s examples are (hopefully) self-explanatory, but I’ll provide some additional notes on a few of the costs:<sup>4</sup>

- **Prevention Costs:** Costs of activities that are specifically designed to prevent poor quality. Examples of “poor quality” include coding errors, design errors, mistakes in the user manuals, as well as badly documented or unmaintainably complex code.

Note that most of the prevention costs don’t fit within the Testing Group’s budget. This money is spent by the programming, design, and marketing staffs.

- **Appraisal Costs:** Costs of activities designed to find quality problems, such as code inspections and any type of testing.

Design reviews are part prevention and part appraisal. To the degree that you’re looking for errors in the proposed design itself when you do the review, you’re doing an appraisal. To the degree that you are looking for ways to strengthen the design, you are doing prevention.

- **Failure Costs:** Costs that result from poor quality, such as the cost of fixing bugs and the cost of dealing with customer complaints.
- **Internal Failure Costs:** Failure costs that arise before your company supplies its product to the customer. These costs go beyond the obvious costs of finding and fixing bugs. Many of the internal failure costs are borne by groups outside of Product Development.

---

<sup>1</sup> Gryna, F. M. “Quality Costs” in Juran, J.M. & Gryna, F. M. (1988, 4<sup>th</sup> Ed.), *Juran’s Quality Control Handbook*, McGraw-Hill, page 4.2.

<sup>2</sup> Feigenbaum, A.V. (1991, 3<sup>rd</sup> Ed. Revised), *Total Quality Control*, McGraw-Hill, Chapter 7.

<sup>3</sup> Gryna, F. M. “Quality Costs” in Juran, J.M. & Gryna, F. M. (1988, 4<sup>th</sup> Ed.), *Juran’s Quality Control Handbook*, McGraw-Hill, page 4.2.

<sup>4</sup> These are my translations of the ideas for a software development audience. More general, and more complete, definitions are available in Campanella, J. (Ed.) (1990), *Principles of Quality Costs*, ASQC Quality Press, as well as in Juran’s and Feigenbaum’s works.

For example, if your company sells thousands of copies of the same program, you will probably print several thousand copies of a multi-color box that contains and describes the program. You (your company) will often be able to get a *much* better deal by booking press time with the printer in advance. However, if you don't get the artwork to the printer on time, you might have to pay for some or all of that wasted press time anyway, and then you may to pay additional printing fees and rush charges to get the printing done on the new schedule. This can be an added expense of many thousands of dollars.

Sometimes the programming group will treat user interface errors as low priority, leaving them until the end to fix. This can be a big mistake. The marketing staff (or packaging production staff) need pictures of the product's screen long before the program is finished, in order to get the artwork for the box into the printer on time. User interface bugs – the ones that will be fixed later – can make it hard for these staff members to take (or mock up) accurate screen shots. Delays caused by these minor design flaws, or by bugs that block a packaging staff member from creating or printing special reports, can cause the company to miss its printer deadline.

- **External Failure Costs:** Failure costs that arise after your company supplies the product to the customer, such as customer service costs, or the cost of patching a released product and distributing the patch.
- **Total Cost of Quality:** The sum of all the costs (Prevention + Appraisal + Internal Failure + External Failure).

**Figure 1. Examples of Quality Costs Associated with Software Products.**

<i>Prevention</i>	<i>Appraisal</i>
<ul style="list-style-type: none"> <li>• Staff training</li> <li>• Requirements analysis</li> <li>• Early prototyping</li> <li>• Fault-tolerant design</li> <li>• Defensive programming</li> <li>• Usability analysis</li> <li>• Clear specification</li> <li>• Accurate internal documentation</li> <li>• Evaluation of the reliability of development tools (before buying them) or of other potential components of the product</li> </ul>	<ul style="list-style-type: none"> <li>• Design review</li> <li>• Code inspection</li> <li>• Glass box testing</li> <li>• Black box testing</li> <li>• Training testers</li> <li>• Beta testing</li> <li>• Test automation</li> <li>• Usability testing</li> <li>• Pre-release out-of-box testing by customer service staff</li> </ul>
<i>Internal Failure</i>	<i>External Failure</i>
<ul style="list-style-type: none"> <li>• Bug fixes</li> <li>• Regression testing</li> <li>• Wasted in-house user time</li> <li>• Wasted tester time</li> <li>• Wasted writer time</li> <li>• Wasted marketer time</li> <li>• Wasted advertisements</li> <li>• Direct cost of late shipment</li> <li>• Opportunity cost of late shipment</li> </ul>	<ul style="list-style-type: none"> <li>• Technical support calls</li> <li>• Preparation of support answer books</li> <li>• Refunds and replacement with updated product</li> <li>• Lost sales</li> <li>• PR work to soften drafts of harsh reviews</li> <li>• Lost customer goodwill</li> <li>• Warranty costs and other costs imposed by law</li> </ul>

In his book, *Out of the Crisis*, W.E. Deming listed *Excessive costs of liability, swelled by lawyers that work on contingency fees* as one of the seven Deadly Diseases. I'm a CQE who recently became a lawyer. My objective is to use the legal system as a vehicle to improve software quality, either as a corporate counsel who works with Engineering in a proactive manner, or as a plaintiff's attorney who files expensive bug reports on a contingent fee basis. In my view, litigation over defective products puts pressure on companies who don't care about their customers. It empowers quality engineers. It is part of the cure, not one of the diseases.

Software quality is often abysmally low. It is impossible to fully test a software product, so all software is necessarily shipped with defects. (For a discussion of the problems of testing, and of the types of defects in software, see my book, *Testing Computer Software*, 2nd Ed., with Jack Falk & Hung Quoc Nguyen, Van Nostrand Reinhold, 1993.) Many companies ship software with significant, known defects. Others don't test the products well enough to discover the most serious problems.

As Quality Engineers, we study quality-related decision making from a financial viewpoint. Our objective is to minimize the cost of quality associated with each product. (See *Principles of Quality Costs*, 2nd Ed., Edited by Jack Campanella, ASQC Quality Press, 1990). Figure 1 provides some representative quality costs associated with the development of software products that will be sold to the public.

Figure 1. Examples of Quality Costs Associated with Software Products.

<b><i>Prevention</i></b>	<b><i>Appraisal</i></b>
<ul style="list-style-type: none"> <li>• Staff training</li> <li>• Requirements analysis</li> <li>• Early prototyping</li> <li>• Fault-tolerant design</li> <li>• Defensive programming</li> <li>• Usability analysis</li> <li>• Clear specification</li> <li>• Accurate documentation</li> </ul>	<ul style="list-style-type: none"> <li>• Design review</li> <li>• Code inspection</li> <li>• Glass box testing</li> <li>• Black box testing</li> <li>• Training testers</li> <li>• Beta testing</li> <li>• Test automation</li> <li>• Usability testing</li> </ul>
<b><i>Internal Failure</i></b>	<b><i>External Failure</i></b>
<ul style="list-style-type: none"> <li>• Bug fixes</li> <li>• Regression testing</li> <li>• Wasted in-house user time</li> <li>• Wasted tester time</li> <li>• Wasted writer time</li> <li>• Wasted marketer time</li> <li>• Wasted advertisements</li> <li>• Direct cost of late shipment</li> <li>• Opportunity cost of late shipment</li> </ul>	<ul style="list-style-type: none"> <li>• Technical support calls</li> <li>• Preparation of support answer books</li> <li>• Refunds and replacement with updated product</li> <li>• Lost sales</li> <li>• PR work to soften drafts of harsh reviews</li> <li>• Lost customer goodwill</li> </ul>

Unfortunately, the ASQC’s Quality Costs Committee omitted an important class of quality-related costs when they published *Principles of Quality Costs*. Look at Appendix B, the *Detailed Description of Quality Cost Elements*, and you’ll see that all of the costs listed are costs borne by the manufacturer / seller of the product.

The manufacturer and seller are definitely not the only people who suffer quality-related costs. The customer suffers quality-related costs too. If a manufacturer sells a bad product, the customer faces significant expenses in dealing with that bad product.

Figure 2 lists some of the external failure costs that are borne by customers, rather than by the company.

Figure 2. Examples of External Failure Costs Borne by the Buyer and the Seller

<b><i>Seller: external failure costs</i></b>	<b><i>Customer: failure costs</i></b>
These are the types of costs absorbed by the seller that releases a defective product.	These are the types of costs absorbed by the customer who buys a defective product.
<ul style="list-style-type: none"> <li>• Technical support calls</li> <li>• Preparation of support answer books</li> <li>• Refunds</li> <li>• Replacement with updated product</li> <li>• PR work to soften drafts of harsh reviews</li> <li>• Lost customer goodwill</li>   <li>• <b><u>Costs imposed by law</u></b></li> </ul>	<ul style="list-style-type: none"> <li>• Wasted time</li> <li>• Lost data</li> <li>• Lost business</li> <li>• Embarrassment</li> <li>• Frustrated employees quit</li> <li>• Failure of demos to customers and other tasks that could only be done once</li> <li>• Cost of replacing product</li> <li>• Cost of reconfiguring the system</li> <li>• Cost of recovery software</li> <li>• Cost of tech support</li> <li>• Injury / death</li> </ul>

Many of the external failure costs, such as goodwill, are difficult to quantify, and many companies therefore ignore them when calculating their cost-benefit tradeoffs. Other external failure costs can be reduced (e.g. by providing cheaper, lower-quality, post-sale support, or by charging customers for support) without increasing customer satisfaction. By ignoring the costs to our customers of bad products, quality engineers encourage quality-related decision-making that victimizes our customers, rather than delighting them.

The point of quality-related litigation is to transfer some of the costs borne by a cheated or injured customer back to the maker or seller of the defective product. The effect of this is to put pressure on the manufacturer to develop higher quality, safer products.

It is fashionable for companies to whine about their customers' ability to sue over defective products. It's not surprising to hear this -- we've all known executives who care more about this quarter's profits than about their longer term relationships with customers. And we've all heard them whine about those dratted nitpickers in QA. But it amazes me when I hear anger toward plaintiffs' lawyers at ASQC meetings -- fundamentally, we are on the same side.