Black Box Software Testing

Spring 2005

Part 4 -- QUALITY COST ANALYSIS

by
Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology
and
James Bach
Principal, Satisfice Inc.

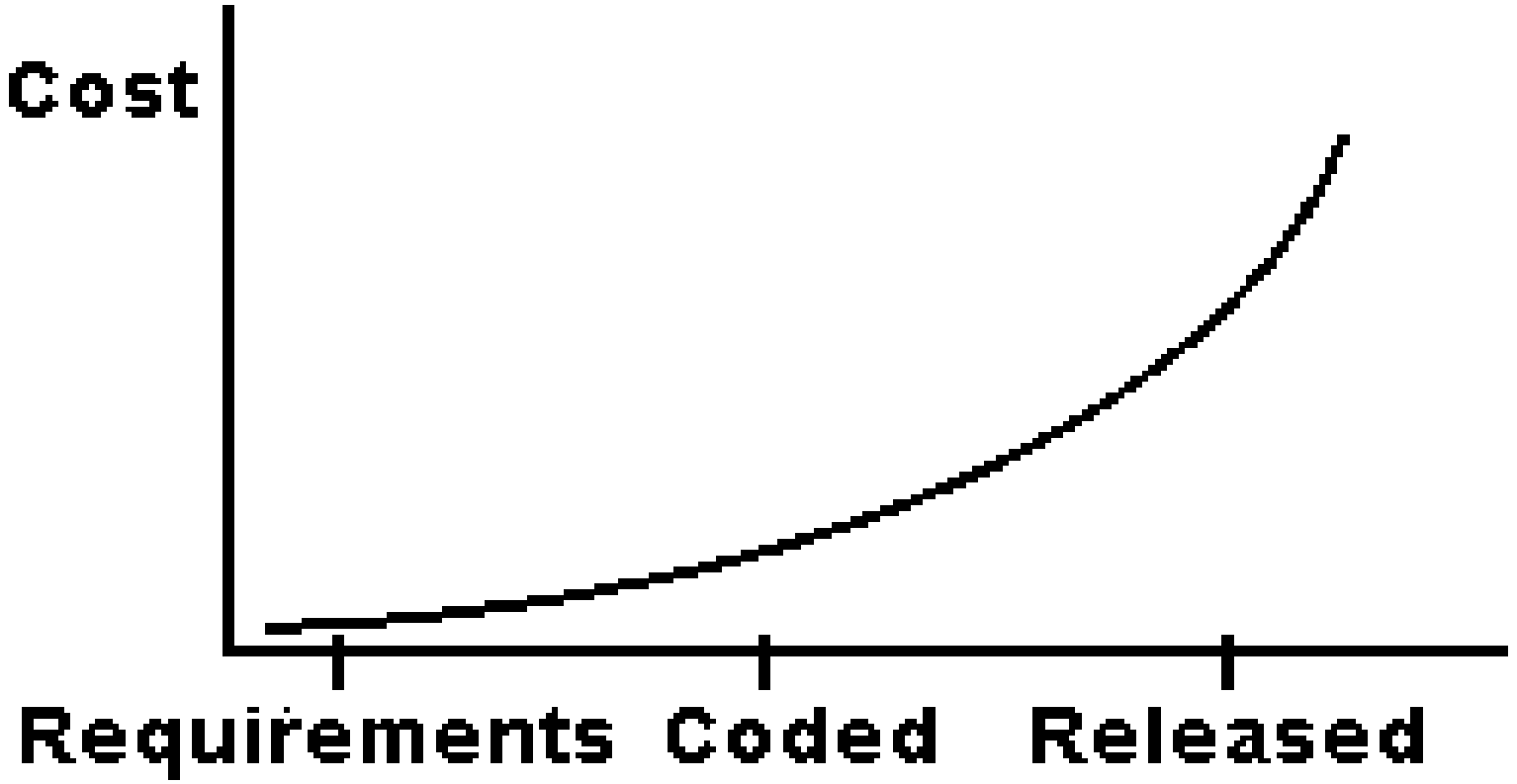
These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Advocating for fixes by alerting people to quality related costs

Suggested Reading:

Kaner, *Quality Cost Analysis: Benefits & Risks.*

Money Talks:
Cost of Finding & Fixing Errors



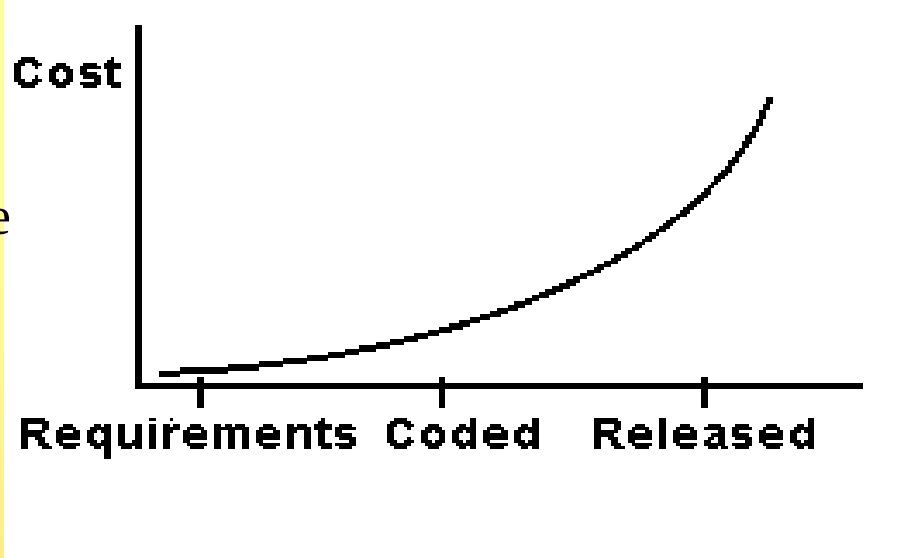
This curve maps the traditionally expected increase of cost as you find and fix software errors later and later in development.

Money Talks:
Cost of Finding & Fixing Errors

This is the most commonly taught cost curve in software engineering.
Usually people describe it from the developers-eye view. That is, the discussion centers around

- how much it costs to find the bug
- how much it costs to fix the bug
- and how much it costs to distribute the bug fix.

But sometimes, it pays to adopt the viewpoints of other stakeholders, who might stand to lose more money than the development and support organizations.

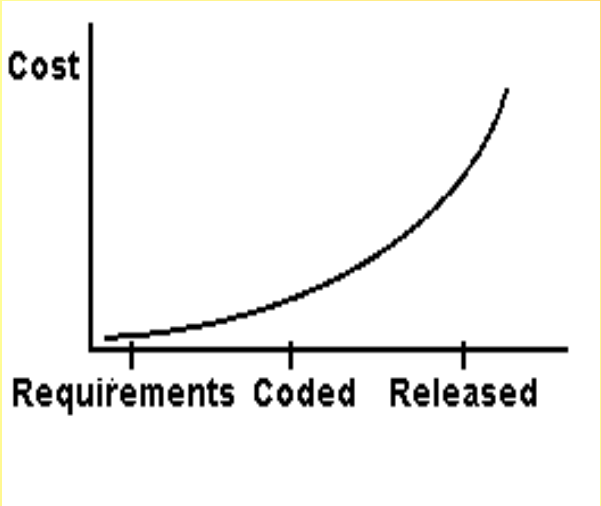


Money Talks:
Cost of Finding & Fixing Errors

From the programmers' viewpoints:

- if we find bugs in requirements, we can fix them without having to recode anything;
- programmers who find their own bugs can fix them without taking time to file bug reports or explain them to someone else;
- it is hugely expensive to deal with bugs in the field (in customers' hands).

Consider too the effects on other stakeholders. Costs escalate because more people in and out of the company are affected by bugs, and more severely affected, as the product gets closer to release. For example, think of the marketing assistant who wastes days trying to create a demo, but can't because of bugs.



Money Talks: Cost of Finding & Fixing Errors

It is important to recognize that this cost curve is predicated on a family of development practices.

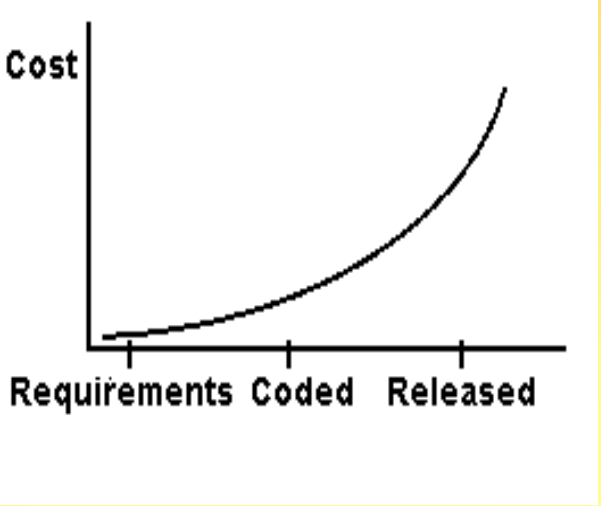
When you see a curve that says,

"Late changes are expensive"

you can reasonably respond in either of two ways:

- Make fewer late changes.
  - *This is the traditional recommendation*
- **Make it cheaper to make late changes.**
  - *This is a key value of the agile development movement (see Beck's Extreme Programming Explained, or go to www.agilealliance.org)*

In this course, I push you to find ways to find bugs earlier, but my development philosophy is agile.



Quality Cost Analysis

- Quality Cost Measurement is a cost control system used to identify opportunities for reducing the controllable quality-related costs
- The *Cost of Quality* is the total amount the company spends to achieve and cope with the quality of its product.
- This includes the company's investments in improving quality, and its expenses arising from inadequate quality.
- A key goal of the quality engineer is to help the company minimize its cost of quality.

» Refer to the paper, *Quality Cost Analysis: Benefits & Risks.*

Quality-Related Costs

Prevention	*Appraisal*
Cost of preventing customer dissatisfaction, including errors or weaknesses in software, design, documentation, and support.	**Cost of inspection (testing, reviews, etc.).**
Internal Failure	*External Failure*
Cost of dealing with errors discovered during development and testing. Note that the company loses money as a user (who can't make the product work) and as a developer (who has to investigate, and possibly fix and retest it).	**Cost of dealing with errors that affect your customers, after the product is released.**

Examples of Quality Costs

Prevention	*Appraisal*
• Staff training • Requirements analysis & early prototyping • Fault-tolerant design • Defensive programming • Usability analysis • Clear specification • Accurate internal documentation • Pre-purchase evaluation of the reliability of development tools	• Design review • Code inspection • Glass box testing • Black box testing • Training testers • Beta testing • Usability testing • Pre-release out-of-box testing by customer service staff
Internal Failure	*External Failure*
• Bug fixes • Regression testing • Wasted in-house user time • Wasted tester time • Wasted writer time • Wasted marketer time • Wasted advertisements • Direct cost of late shipment • Opportunity cost of late shipment	• Lost sales and lost customer goodwill • Technical support calls • Writing answer books (for Support) • Investigating complaints • Supporting multiple versions in the field • Refunds, recalls, warranty, legal costs • Interim bug fix releases • Shipping updated product • PR to soften bad reviews • Discounts to resellers

Customers' Quality Costs

Seller: external costs	Customer: failure costs (seller's externalized costs)
These illustrate costs absorbed by the seller that releases a defective product. • **Lost sales and lost customer goodwill** • **Technical support calls** • **Writing answer books (for Support)** • **Investigating complaints** • **Refunds, recalls, warranty, legal costs** • **Government investigations** • **Supporting multiple versions in the field** • **Interim bug fix releases** • **Shipping updated product** • **PR to soften bad reviews** • **Discounts to resellers**	*These illustrate costs absorbed by the customer who buys a defective product.* • **Wasted time** • **Lost data** • **Lost business** • **Embarrassment** • **Frustrated employees quit** • **Failure during one-time-only tasks, e.g. demos to prospective customers** • **Cost of replacing product** • **Reconfiguring the system** • **Cost of recovery software** • **Tech support fees** • **Injury / death**

Influencing Others Based on Costs

- It's often impossible to fix every bug. Sometimes the development team will choose to not fix a bug based on their assessment of its risks for them, without considering costs to other stakeholders.
  - Probable tech support cost.
  - Risk to the customer.
  - Risk to the customer's data or equipment.
  - Visibility in an area of interest to reviewers.
  - Extent to which the bug detracts from the use of the program.
  - How often will a customer see it?
  - How many customers will see it?
  - Does it block any testing tasks?
  - Degree to which it will block OEM deals or other sales.
- To argue against a deferral, ask yourself which stakeholder(s) will pay the cost of keeping this bug. Flag the bug to them.