

Balancing Quality with Constraints

Success and Failure in Software Delivery



RBCS

**TIME TESTED.
TESTING IMPROVED.**

www.RBCS-US.com



Achieving a Successful Balance

- ❖ In a perfect world, we would have boundless time and resources to deliver systems with plentiful features, each feature having perfect quality
- ❖ In the real world, we must balance schedule, budget, features, and quality
- ❖ When schedule, budget, and features get the attention, we have a quality quandary
- ❖ What exactly is the problem?
- ❖ How can we achieve a successful balance?
- ❖ Let's look at what works and what doesn't...

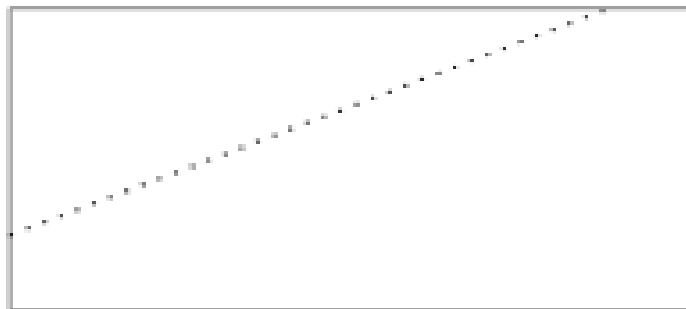


What Tradeoff?

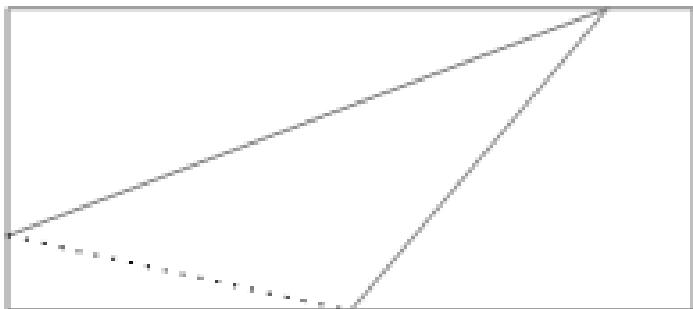
Define Features



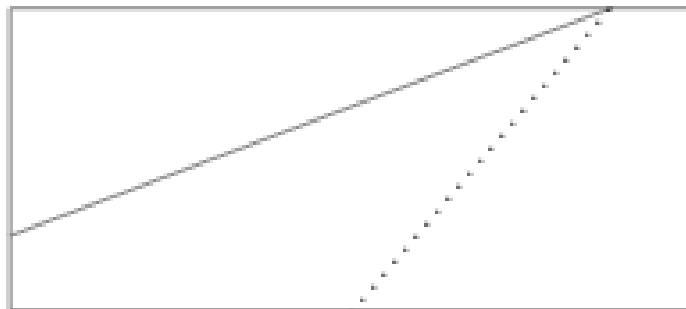
Select Schedule



Accept Quality?



Select Cost





Life Involves Trade-offs

- ❖ Quality versus schedule
 - ❖ Discovering numerous, unknown bugs late in the project increases testing costs, risks of project delay
 - ❖ Without addressing quality throughout, the schedule, quality, and features are unpredictable
- ❖ Features versus schedule
 - ❖ Without realistic estimation, features can be dropped at the end of a project – sometimes the wrong features
 - ❖ Fixing the right bugs (not necessarily all bugs) is also critical
- ❖ Plans and estimates must be robust and realistic, given the known unknowns and unknown unknowns that always exist, but good plans and estimates are not enough



Five Elements of Quality Balance

- ✦ Shared vision
- ✦ Disciplined management
- ✦ Quality in, and bugs out, throughout
- ✦ Focused testing
- ✦ Send the right message



Shared Vision of Product

- ❖ “A camel is a horse designed by a committee”
- ❖ It's not the diverse perspectives that cause the problem, it's the lack of coalescence
- ❖ The shared vision must include the proper balance between budget, schedule, feature, and quality elements
- ❖ An endeavor without shared objectives cannot succeed



Good Example: Apple iPhone, iPad, iPod

- ❖ Apple's iPhone, iPad, and iPod products have been wildly successful
- ❖ Part of the success has been an uncompromising commitment to key design and quality attributes
- ❖ While these products have bugs (especially compatibility, reliability, and maintenance), they clearly reflect a shared vision of each product and the product line as a whole



Bad Example: FBI

- ❖ Over a 10 year period starting in the late 1990s, the US FBI tried to automate its files
- ❖ The project meandered, picking up new features and changing scope
- ❖ Consulting companies made a lot of money from change orders
- ❖ Ultimately, the project was killed without any deliverable



Disciplined Management

- ⊕ Know what features to leave out
- ⊕ Know when to say “no”
- ⊕ Know which quality attributes matter to the users and customers
- ⊕ Don't shortcut the parts of the software process that build quality
- ⊕ Everyone has a role to play in project success, including product quality
- ⊕ Everyone has a voice in the project management team, including the test manager



Good Example: IBM z/OS

- ✦ The OS/360 project built IBM's mainframe OS (now z/OS)
- ✦ Code has continued to mature over 40 years
- ✦ Over 90 percent of Fortune 500 companies still use z/OS for core business
- ✦ What did the Brooks and his team do?
 - ❑ Requirements reviews, including independent testers
 - ❑ Unit testing and system testing
 - ❑ Unified design (chief programmer teams)
 - ❑ Understood which quality characteristics were key



Do the IBM Lessons Still Apply?

- ❖ An RBCS banking client follows many of the IBM practices
 - ❖ Requirements reviews lead by the test team
 - ❖ Unit testing and component integration testing by developers
 - ❖ System and system integration testing by independent testers
 - ❖ Key quality characteristics identified during project initiation
- ❖ They deliver innovative technology for the bank (a strategic goal) along with very high quality

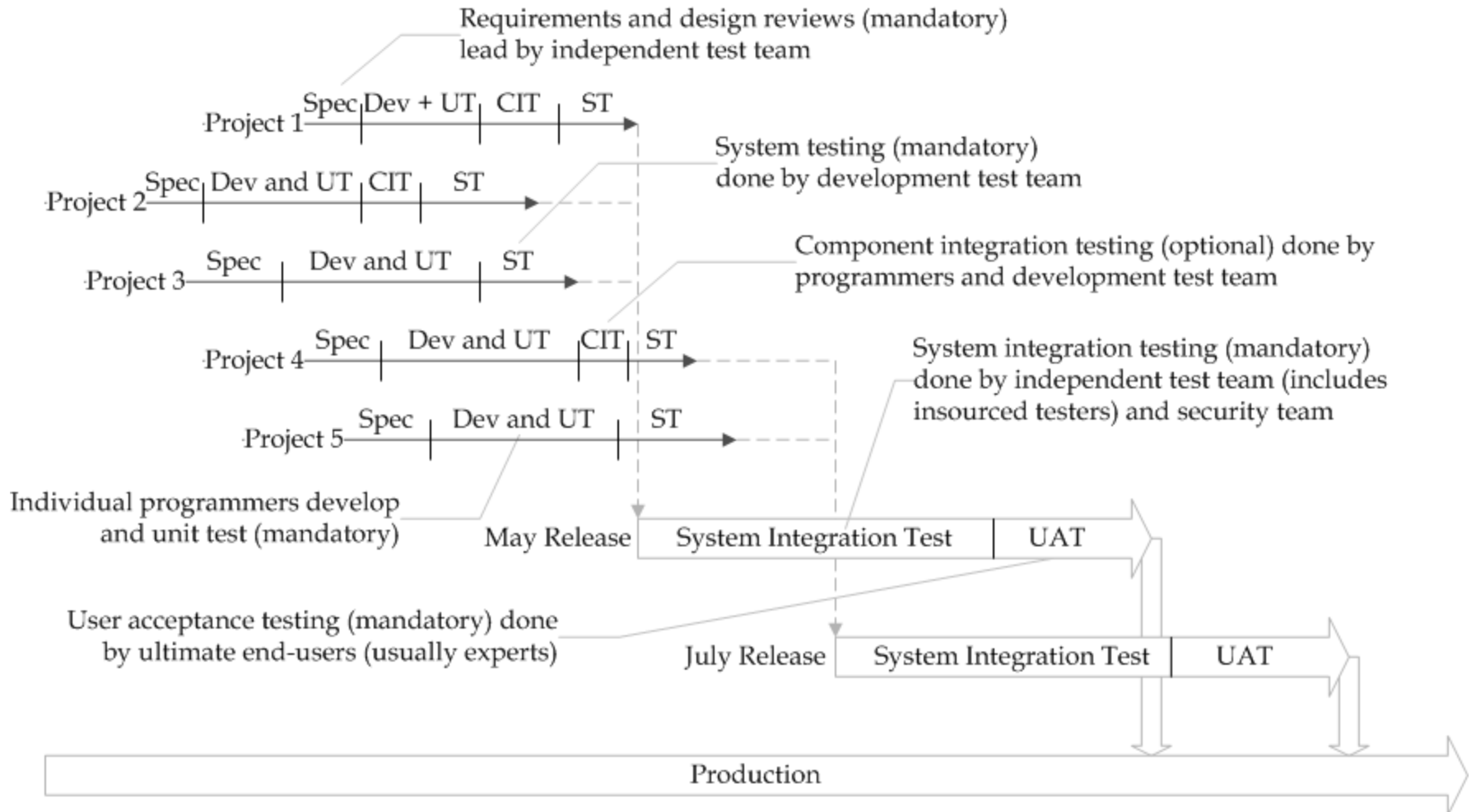


Quality In, Bugs Out, Throughout

- ❖ One sure way to provoke a quality crisis for a product is to forget quality...until the very end
- ❖ The entire project team must take responsibility for quality
- ❖ Build all work products with an emphasis on quality, and remove bugs early in the lifecycle
- ❖ Good phase containment will lower cost and shortening schedules
- ❖ This applies to sequential and iterative lifecycles (including Agile)



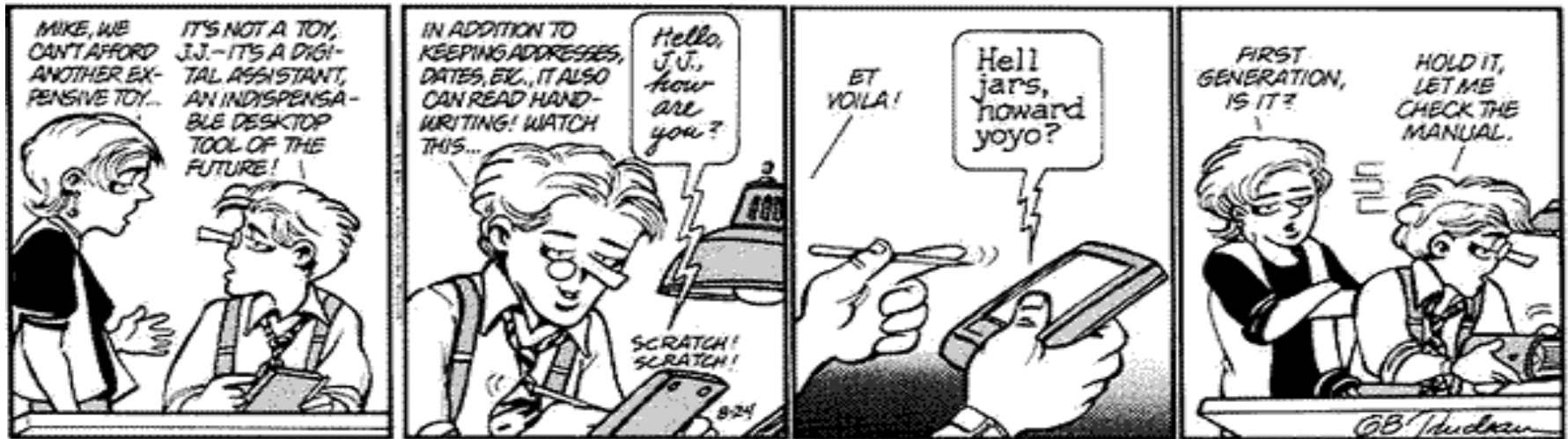
Good Example: Quality Throughout





Bad Example: Apple Newton

- ✦ The late Steve Jobs lead a renaissance for Apple
- ✦ During Jobs' absence, the focus on quality wasn't as high
- ✦ Consider the Newton, released buggy
- ✦ First to market doesn't help if the feature doesn't work!
- ✦ The Newton cost millions by effectively keeping Apple out of the PDA market for years



www.rbc-us.com



Focused, Well-Managed Testing

- ❖ As shown earlier, testing is a key part of a good quality strategy (but not the only part)
- ❖ Testing is more than just a geeky bug hunt
- ❖ Testing needs to focus on the important attributes of the system
- ❖ Testing needs clearly defined objectives and proper management towards those objectives
- ❖ The bugs found by testing must be triaged based on priority, making the fix or defer decision on costs, benefits, risks, and opportunities



Bad Example: Finding the Wrong Bugs

- ❖ In our assessments, we check to see if the testing is focused
- ❖ Part of the evaluation involves looking at the defect detection effectiveness
- ❖ We want $DDE (all) < DDE (high\ priority)$
- ❖ This usually doesn't occur unless risk based testing is used

$$DDE = \frac{\textit{test bugs}}{\textit{test bugs} + \textit{production bugs}}$$



Good Example: Clearly Defined Objectives

Test Level	Owner	Objectives
Unit	Development	<ul style="list-style-type: none">•Early detection of bugs in code units•Reduce risk of unit failures in production•Unit testing run before CIT•Unit tests results documented in the feature turnover
CIT (Comp Int Test)	Development	<ul style="list-style-type: none">•Early detection of bugs in unit interfaces•Reduce risk of dataflow and workflow failures in production•Ensure development ownership for delivering quality features
CIT	QA	<ul style="list-style-type: none">•Early QA validation of completed features
System/SIT	QA	<ul style="list-style-type: none">•Detect bugs, reduce risk, build confidence in use cases and end-to-end scenarios•Detect bugs, reduce risk, build confidence in user workflows
Beta	Customer	<ul style="list-style-type: none">•Detect bugs related to customer deployment•Reduce risk of failing customer business requirements•Demonstrate readiness for deployment

www.rbc-us.com



Send the Right Message

- ❖ Does quality matter?
- ❖ What are we ready to do to achieve quality?
- ❖ Management needs to talk about quality – in balance
- ❖ Management must avoid sending the wrong message with incentives
- ❖ When you hear people talking about quality and doing good work, you know the message is being sent



Bad Examples: Wrong, Dropped Message

- ✦ Assessment question: “In terms of priorities, rank quality, schedule, budget, and features”
 - ❑ Line management and individuals: Schedule, budget, features, quality
 - ❑ Senior management: Quality is always first
- ✦ Looks like the message isn't getting through
- ✦ Assessment questions
 - ❑ To non-test staff: Q: What is the value of testing? A: “Testing has no value”
 - ❑ To test staff: Q: What role do other teams play in quality? A: “They are lazy and don't care about quality”
- ✦ Looks like the wrong message is being sent



Good Example: Quality is Everyone's Job

- ❖ In a recent assessment, many of the programmers talked about using open source tools for unit testing and static analysis
- ❖ One programmer said, "We spend a lot of time here talking about how to write better quality code"
- ❖ Programmers work collaboratively with testers to achieve higher levels of quality



Still No Silver Bullets

- ❖ Lifecycles don't matter as much as the hype says they do
- ❖ New languages and platforms have boosted productivity (in terms of features per developer month), but not quality
- ❖ Many organizations have excuses for their quality problems, but excuses are a poor substitute for success
- ❖ Most of what we know about building quality systems within constraints, we have known for 20 years or more
- ❖ Fads have come and gone, but best practices abide



Conclusions

- ❖ Organizations can balance quality with features, budget, and schedule through: a shared vision; disciplined management; quality in, bugs out, throughout; focusing testing; and, sending the right message
- ❖ Examples abound of success with these methods, and failures when not observed
- ❖ All projects will involve tradeoffs...the only question is whether those will be made consciously, carefully, and in a mature fashion
- ❖ There are no magic solutions to the quality quandary...just disciplined application of known best practices



To Contact RBCS

For almost twenty years, RBCS has delivered consulting, outsourcing and training services for software and hardware testing and quality. Employing the industry's most experienced and recognized consultants, RBCS conducts product testing, builds and improves testing groups and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbc-us.com.

Address: RBCS, Inc.
31520 Beck Road
Bulverde, TX 78163-3911
USA

Phone: +1 (830) 438-4830
Fax: +1 (830) 438-4831
E-mail: info@rbc-us.com
Web: www.rbc-us.com