

Test Estimation and the Art of Negotiation



STARWEST 2011

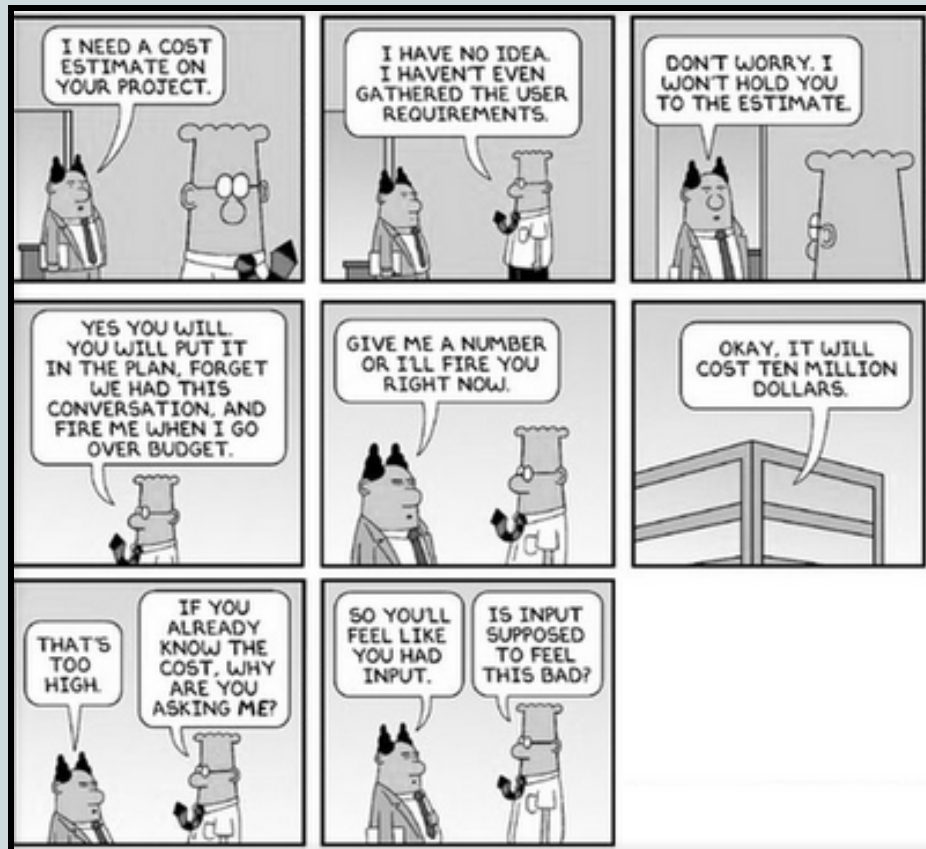
Anaheim, CA

Lynn McKee & Nancy Kelln
Quality Perspectives, Unimagined Testing



@lynn_mckee @nkelln

Estimation Reality?



Testing & Time



How long will
it take to test?

Behind the Question



- How much time is really needed for testing?
- How long would it take for complete testing?
- What is the least amount of time we could allocate to testing?
- Do we need to test at all?
- How much testing is needed to *assure* a high quality solution?
- How many test cases will be needed?
- How many testers are required?
- What testing skills are required?
- How much will testing cost?
- *and more...*

Common Test Estimation Models



Estimation Models



- Complex, scientific models are used to calculate “accurate” estimates
- Simplistic, heuristic models are used to craft a best guess
- Problem is we are often fooled by the intricate models and lulled into impossible commitments even with the simplistic ones
- No matter which model we use, there is pressure to improve our estimates by reviewing the project details more carefully, assessing and forecasting more risks, assuming greater dependencies, and adding more contingency

Critical Planning Tactic



- Test estimates are often prefaced with exhaustive lists of:
 - Assumptions
 - Constraints
 - Dependencies
 - Risks
- These become justification for inaccurate estimates

Mitigation Planning Tactic



- Test estimates are often heavily padded to offset concerns with the assumptions, constraints, dependencies and risks
- Tasks are padded with intent that some portion of the tasks will take less time therefore offsetting those that may end up taking longer
 - The asymmetry of a few overdue tasks overshadows the gains of many early tasks
- Estimates can only be buffered for the risks you foresee
- Consistent padding leads to reduced value, increased skepticism, and potential trust issues

Unknown Unknowns & Black Swans



- It is impossible to strategically plan for or mitigate the occurrence of Black Swans and Unknown Unknowns
- Unknown Unknowns
 - Things that can not be planned for because by their very nature we do not know about them
- Black Swans
 - A Black Swan is an improbable and unexpected event with three characteristics:
 - ✦ Black Swans take us completely by surprise, typically because they are outside of our models
 - ✦ Black Swans have a disproportionately large impact
 - ✦ After a Black Swan, people have a tendency to say that they saw it coming

- The Black Swan, Nassim Nicholas Taleb
- Five Orders of Ignorance, Phillip G. Armour

Dangers of Optimism



- We have tendency to be overly optimistic
- People consistently under estimate their own work and over commit
- Desire to believe a few late tasks are generally offset by early tasks
- Day to day activities such as setup, bug investigation, and reporting have a significantly under anticipated impact on the available testing time

Impact of Bugs on Estimates



- Consider the effect of finding bugs on our testing efforts
 - Suppose testing a feature takes 2 minutes (highly arbitrary and artificial assumption – that is, it's wrong but we are using it to model a point)
 - Suppose it takes 8 minutes to investigate and report a bug (another sweeping generalization in service of the point)
 - Suppose a testing session lasts 90 minutes, we can run 45 feature tests – *as long as we don't find any bugs*

- Rapid Software Testing Course, James Bach & Michael Bolton

Impact of Bugs on Estimates



Day 1

* Assume all tests would find bugs if bugs were present.

Module	Bug Reporting/ Investigation	Test Design & Execution	Number of Tests
A (good)	0 minutes (no bugs found)	90 minutes (45 tests)	45
B (okay)	10 minutes (1 bug, 1 test)	80 minutes (40 tests)	41
C (bad)	80 minutes (8 bugs, 8 tests)	10 minutes (5 tests)	13

- Rapid Software Testing Course, James Bach & Michael Bolton

Impact of Bugs on Estimates



Day 2

* Assume 6 minutes per bug fix verification.

Mod	Fix Verifications	Bug Reporting/ Investigation	Test Design & Execution	New Tests Today	Total Over 2 Days
A	0 min	0 minutes (no bugs found)	90 minutes (45 tests)	45	90
B	6 min	10 minutes (1 bug, 1 test)	74 minutes (37 tests)	38	79
C	48 min	40 minutes (4 bugs, 4 tests)	2 minutes (1 tests)	5	18

- Rapid Software Testing Course, James Bach & Michael Bolton

The Best Test Estimate is...



No Estimate at All!

Many hours are often spent developing estimates that would be better spent actually testing!

What are we Estimating?



- **Let's step back and reconsider what we are estimating:**
 - What is the testing service being provided?
 - When does this service begin?
 - When does this service end?
 - Who are the customers of this service?
 - How do we determine the value of this service?

Impossible Estimating



Test Estimation is Really Negotiation



- **Need to negotiate:**
 - The Testing Service
 - The Available Time for Testing
 - The People, Resources, and Budget

Negotiating the Testing Service



- Gaining clarity on the expectations of the testing effort
- Advocating for testing as a service and helping management understand:
 - The role of testing as an information service
 - The information service is provided in support of business critical decisions about the product's readiness
 - Testing is a combination of exploration, experimentation, discovery, investigation and learning
 - Testing is never an independent project
 - The notion of the Testing Phase is misleading as testing is ongoing; we do not start and stop “seeing”
 - Pre-emptive and prescriptive test planning is ineffective

Negotiating the Testing Service Cont.



- The delineation of Development and Testing phases is nonsense
 - ✦ Programmers are always programming and testers are always testing; the difference is the team's focus on new development vs. fixing
- Measuring the testing effort in test cases is extremely poor given the range of variance in the definition of test cases and the relative time to execute
- Testing does not serve as the quality police or production gate keepers especially by measuring expected levels of quality in terms of execution percentages or defect counts
- The impossibility of complete testing
- The decision to stop testing is a business decision not a technical one

Negotiating the Available Time for Testing



- Time available for testing is often obscured but already determined by the project dates
- Business initiatives *always* have target dates
- Majority of projects are based on right to left planning
- Some projects are work within canned timeframes
 - Projects may be have set timeframes for the collective delivery
 - Agile projects using sprints of 2 – 3 week defined time frames
 - The time available for testing is already determined by the iterations
- Clarify the communication strategy for keeping management informed on the product quality and how decisions will be made to extend (or shorten) the project if required to address quality concerns

Negotiating the People, Resources, & Budget



- The amount of testing possible within the available timeframe can be amplified by many factors:
 - Number of appropriately skilled testers
 - Available resources (environments, tools, data)
 - Availability of support personnel (programmers, business analysts, DBA's, network analysts, etc)
- As the project progresses, be cautious of “Brook’s Law” that “adding manpower to a late software project makes it later”

Revealing the Trade-offs



Business Drivers

+

Quality Expectations

+

Business Risks

+

Technical Risks

+

\$\$\$

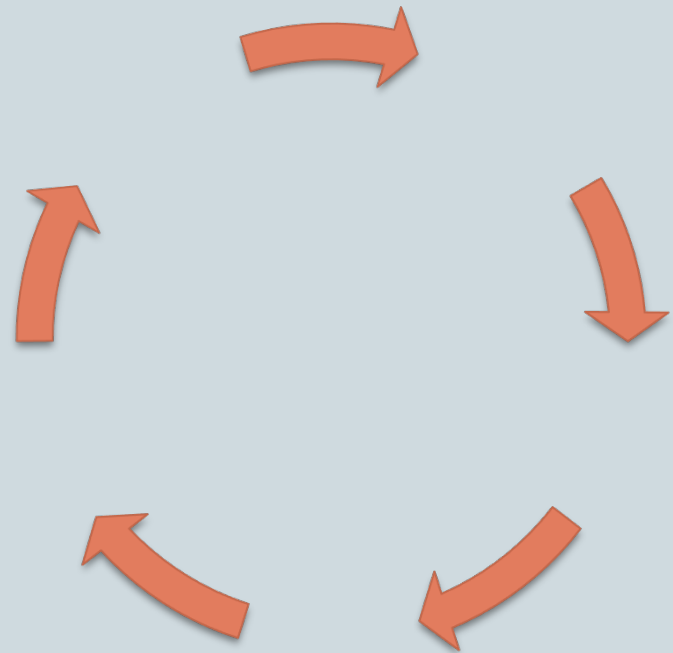
=

Testing: Service, Time, Budget, People & Resources

Iterate to Meet the Project's Changing Needs



- Iterate discussions on the testing service, time, budget, people, and resources as needed:
 - Changes in scope
 - Unknown unknown is realized
 - Black Swan occurs
 - Schedule delays
 - Quality concerns



Role Play #1 – How Long Will Testing Take



Role Play #3 – Communication w/Leadership



Summary



- We are often fooled by intricate estimation models and lulled into impossible commitments even with simplistic ones
- Developing exhaustive lists of assumptions, constraints, dependencies and risks only provides justification for estimates being inaccurate
- Mitigation planning with heavy padding only leads to low value estimates, skepticism and potential trust issues
- Unknown Unknowns and Black Swans are unavoidable and have significant impacts on our estimates
- Optimism can blind us from seeing the impact of late tasks and the identification of bugs

Summary



- **Stop estimating and begin negotiating**
 - Negotiate the testing service, available time for testing, and people, resources and budget
 - Communicate with management to help reveal the trade-offs for the amount of testing on your project
 - Iterate to meet the project needs when changes arise in scope, unknown unknowns are realized, a Black Swan occurs, the schedule is delayed, etc

Questions?



Lynn McKee
Quality Perspectives
Calgary, AB Canada

lynnmckee@qualityperspectives.ca

www.qualityperspectives.ca



Nancy Kelln
Unimagined Testing
Airdrie, AB Canada

nancy.kelln@unimaginedtesting.ca

www.unimaginedtesting.ca

References



- **Books**

- The Black Swan by Nassim Nicholas Taleb
- The Mythical Man Month by Frederick P. Brooks
- Quality Software Management, Vol. 1: Systems Thinking by Gerald M. Weinberg
- Quality Software Management, Vol. 2: First-Order Measurement by Gerald M. Weinberg
- Quality Software Management, Vol. 3: Congruent Action by Gerald M. Weinberg
- Quality Software Management, Vol. 4: Anticipating Change by Gerald M. Weinberg
- Perfect Software by Gerald M. Weinberg
- Lessons Learned in Software Testing by Cem Kaner, James Bach, Bret Pettichord

References



- **Articles**

- “Staying on the Critical Path” by Michael Bolton,
<http://www.developsense.com/articles/2005-09-StayingOnTheCriticalPath.pdf>
- “Swan Song” by Michael Bolton,
<http://www.developsense.com/articles/2010-01-SwanSong.pdf>
- “Elemental Models” by Michael Bolton,
<http://www.developsense.com/articles/2005-10-ElementalModels.pdf>
- “The Case Against Test Cases” by James Bach,
<http://www.satisfice.com/presentations/againststestcases.pdf>

References



- Webpages & Blog Posts

- “Estimating Projects: A History of Failure” by Gerald M. Weinberg, <http://secretsofconsulting.blogspot.com/2006/04/estimating-projects-history-of-failure.html>
- “Test Project Estimation, The Rapid Way” by Michael Bolton, <http://www.developsense.com/blog/2007/01/test-project-estimation-rapid-way/>
- “Test Estimation is Really Negotiation” by Michael Bolton, <http://www.developsense.com/blog/2009/08/test-estimation-is-really-negotiation/>
- “Project Estimation and Black Swans (5 part series) by Michael Bolton, <http://www.developsense.com/blog/2010/10/project-estimation-and-black-swans/>
- “Why is Testing Taking So Long (Part 1 & 2)” by Michael Bolton, <http://www.developsense.com/blog/2009/11/why-is-testing-taking-so-long-part-1/>
- “More Responses to the Agile Testing Questions” by Michael Bolton, http://www.developsense.com/blog/archive/2006_05_01_archive.html

References



- **Webpages & Blog Posts Continued**
 - “Test Effort Estimation: Getting it Wrong Before Getting it Right” by Pradeep Soundarajan,
<http://testertested.blogspot.com/2010/01/test-effort-estimation-getting-it-wrong.html>
 - “The Zeroth Law of Quality” by Tony Bowden,
<http://dustyvolumes.com/archives/505>
 - “Session Based Test Management” by James Bach,
<http://www.satisfice.com/sbtm/>
 - “Negligence and Testing Coverage” by Cem Kaner,
http://www.kaner.com/pdfs/negligence_and_testing_coverage.pdf
 - “The Impossibility of Complete Testing” by Cem Kaner,
<http://www.kaner.com/pdfs/imposs.pdf>
 - “Testing vs. Checking” by Michael Bolton,
<http://www.developsense.com/blog/2009/08/testing-vs-checking/>
 - “Ludic Fallacy”, http://en.wikipedia.org/wiki/Ludic_fallacy