‘Lean QA’ : Enabling Quality, through tools and technology

Wildcard Conference, Jurmala, Latvia, Friday Sept. 13 2013

http://wildcardconf.com

11:00 to 11:45 Testing Track

by Gilb

Copyright: © Gilb 2010-2013,
Slide owner: Kai@Gilb.com @kaigilb
Tom@Gilb.com @imtomgilb

www.gilb.com

These slides will be at:
Main Take-away Points

Quality Assurance is far more than ‘test’, and it can be far more cost-effective

‘Quality’ is far more than ‘bugs’

You probably have a lot to learn, if you want real competitive quality
Begin:

Quality Assurance is far more than ‘test’

and it can be far more cost-effective
a story
Inspection Effectiveness

Capers Jones
Regression test?

15% to 30%
Integration test?

25% to 40%
<table>
<thead>
<tr>
<th>Test Type</th>
<th>Duration (% of Project)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit test</td>
<td>15% to 50%</td>
</tr>
<tr>
<td>New function test</td>
<td>20% to 35%</td>
</tr>
<tr>
<td>Performance test</td>
<td>20% to 40%</td>
</tr>
<tr>
<td>System test</td>
<td>25% to 55%</td>
</tr>
<tr>
<td>Acceptance test (1 client)</td>
<td>25% to 35%</td>
</tr>
<tr>
<td>Low-volume Beta test (&lt; 10 clients)</td>
<td>25% to 40%</td>
</tr>
<tr>
<td>High-volume Beta test (&gt; 1000 clients)</td>
<td>60% to 85%</td>
</tr>
</tbody>
</table>
Inspections?

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informal design reviews</td>
<td>25% to 40%</td>
</tr>
<tr>
<td>Formal design inspections</td>
<td>45% to 65%</td>
</tr>
<tr>
<td>Informal code reviews</td>
<td>20% to 35%</td>
</tr>
<tr>
<td>Formal code inspections</td>
<td>45% to 70%</td>
</tr>
</tbody>
</table>
Best Practice Testing Combined
Little hope of ‘zero defects’

“Between 8 and 10 defect removal stages required to achieve removal effectiveness of 95%”
IBM Defect Avoidance Experience

Defect Prevention Effectiveness

- In Field
- Test
- Inspection

Year:
- 1st year
- 2nd year
- 5th year
1. Electric motor (15kW / 210Nm).
2. Hydraulic torque converter with lock-up-clutch.
3. 8-speed automatic transmission.
You don’t get quality by testing it in
but by ‘Engineering’ Quality In

Work hours

$€\text{ Kr.}$

Reliability
Performance
Security
Usability
Maintenance
Setting Quality Goals

Usability.Learn

Scale: average time to Learn how to operate the computer, from .. to ..

Status [today] 3 hours
Goal [next year] 10 min.
Designing to meet Quality within Costs

<table>
<thead>
<tr>
<th>Product Quality Requirements</th>
<th>Estimated Impact</th>
<th>Estimated Impact</th>
<th>Estimated Impact</th>
<th>Estimated Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past</td>
<td>Units</td>
<td>Units</td>
<td>Units</td>
<td>Units</td>
</tr>
<tr>
<td>User-Friendliness.Learn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>by a year</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>20</td>
<td>25</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>by a year</td>
<td>23%</td>
<td>29%</td>
<td>0%</td>
<td>12</td>
</tr>
<tr>
<td>Style</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0,5</td>
<td>-0,5</td>
</tr>
<tr>
<td>by a year</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Sum of Benefits</td>
<td>23%</td>
<td>29%</td>
<td>7%</td>
<td>12</td>
</tr>
<tr>
<td>Development Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project-Budget</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1000</td>
<td>1700</td>
<td>3000</td>
<td>2000</td>
</tr>
<tr>
<td>4500</td>
<td>1%</td>
<td>2%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>140000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1E+05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum of Development Resources</td>
<td>1%</td>
<td>2%</td>
<td>3%</td>
<td>2%</td>
</tr>
<tr>
<td>Benefits / Development Resources</td>
<td>22,21</td>
<td>16,33</td>
<td>2,12</td>
<td>5,55</td>
</tr>
</tbody>
</table>
Quality Assurance is far more than ‘test’

and, QA can be far more cost-effective
Quality is far more than ‘bugs’
System Performance

- Capacity: ‘How Much’
- Quality: ‘How Well’
- Resource Saving: ‘Efficiency’
Qualities are many and variable

Usability
- Learning
- Doing
- Error Rate

Adaptability
- Portability
- Enhancability
- Compatibility

Integrity
- Threat Type and Frequency
- Security Mitigation

Availability
- Reliability
- Maintainability (fault fix speed)
Quantify the Quality to ‘Assure’ It

I often say that
when you can **measure**
what you are speaking about,
and **express it in numbers**, you know something about it;

but when you **cannot measure** it,
when you **cannot express it in numbers**, your knowledge is of a meagre and unsatisfactory kind;

- Lord Kelvin, 1893
Whittaker, Google, is now experimenting in real Google projects.

He has **totally eliminated** the use of **professional testers** on his team, replacing them with a set of **more cost effective means** for ‘testing’ the software. (Construx Summit Talk, Oct 2011, Seattle)

James Whittaker
Engineering Director
Google

If following my work appeals to you:
+docjamesw (Google+)
@docjamesw (Twitter)
googledevspot.blogspot.com
gooqletesting.blogspot.com
Competitive Lean QA methods to Learn
1. Stakeholders Decide Qualities
2.

**Motivation**
- Contracting for results
- Paying Contractors for results
- Reward teams for results achieved
- Motivate Nerds towards Business

**Analysis**
- Comparative Evaluation
- Deadline Completion Estimation
- Data Collection & Learning
- Research

**QC**
- Quality Requirement Testing
- Design Inspections and Reviews

**Requirements**
- Communication of Primary Requirements
- Simplify requirements to Top Ten Critical Ones

**Management**
- Project Management

**Quality Quantification**
“As I see it Tom Gilb was the inspiration for much of what is defined in CMM Level 4.”

Ron Radice (CMM Inventor at IBM) 1996 Salt lake City (agreed orally by Watts Humphreys - his IBM Boss)

stt@stt.com, www.stt.com
Lack of clear top level project objectives has seen real projects fail for $100+ million: personal experience, real case

Bad Objectives, for 8 years

1. Central to The Corporation's business strategy is to be the world's premier integrated <domain> service provider.

2. Will provide a much more efficient user experience

3. Dramatically scale back the time frequently needed after the last data is acquired to time align, depth correct, splice, merge, recompute and/or do whatever else is needed to generate the desired products.

4. Make the system much easier to understand and use than has been the case for previous system.

5. A primary goal is to provide a much more productive system development environment than was previously the case.

6. Will provide a richer set of functionality for supporting next-generation logging tools and applications.

7. Robustness is an essential system requirement (see partial rewrite in example at right).

8. Major improvements in data quality over current practice

Quantified Objectives (in Planguage), Robustness, Testability:

Type: Software Quality Requirement.
Version: 20 Oct 2006-10-20
Status: Demo draft,
Stakeholder: {Operator, Tester}.
Ambition: Rapid-duration automatic testing of <critical complex tests>, with extreme operator setup and initiation.

Scale: the duration of a defined [Volume] of testing, or a defined [Type], by a defined [Skill Level] of system operator, under defined [Operating Conditions].

Goal [All Customer Use, Volume = 1,000,000 data items, Type = WireXXXX Vs DXX, Skill = First Time Novice, Operating Conditions = Field, {Sea Or Desert}. <10 mins.]
VALUE CLARITY:
Quantify the most-critical project objectives on day 1

P&L-Consistency&T P&L: Scale: total adjustments btw Flash/Predict and Actual (T+1) signed off P&L per day. Past 60 Goal: 15

Speed-To-Deliver: Scale: average Calendar days needed from New Idea Approved until Idea Operational, for given Tasks, on given Markets.
Past [2009, Market = EURex, Task = Bond Execution] 2-3 months?
Goal [Deadline = End 20xz, Market = EURex, Task = Bond Execution] 5 days

Operational-Control: Scale: % of trades per day, where the calculated economic difference between OUR CO and Marketplace/Clients, is less than “1 Yen” (or equivalent).
Past [April 20xx] 10% change this to 90% NH Goal [Dec. 20xy] 100%

Operational-Control.Consistent: Scale: % of defined [Trades] failing full STP across the transaction cycle. Past [April 20xx, Trades=Voice Trades] 95%
Past [April 20xx, Trades=eTrades] 93%
Goal [April 20xz, Trades=Voice Trades] <95 ± 2%
Goal [April 20xz, Trades=eTrades] 98.5 ± 0.5%

Operational-Control.Timely.End&OvernightP&L Scale: number of times per day the intraday P&L process is delayed more than 0.5 sec.
Operational-Control.Timely.Trade-Bookings Scale: number of trades per day that are not booked on trade date. Past [April 20xx] 20?

Front-Office-Trade-Management-Efficiency Scale: Time from Ticket Launch to trade updating real-time risk view
Past [20xx, Function = Risk Mgt, Region = Global] ~ 80s +/- 45s ??
Goal [End 20xz, Function = Risk Mgt, Region = Global] ~ 50% better?
Managing Risk – Accurate – Consolidated – Real Time

Risk.Cross-Product Scale: % of financial products that risk metrics can be displayed in a single position blotter in a way appropriate for the trader (i.e. – around a benchmark vs. across the curve).
Past [April 20xx] 0%
Goal [Dec. 20xy] 100%

Risk.Low-latency Scale: number of times per day the intraday risk metrics is delayed by more than 0.5 sec. Past [April 20xx, NA] 1% Past [April 20xx, EMEA] ??% Past [April 20xx, AP] 100%
Goal [Dec. 20xy] 0%
Risk.Accuracy

Risk.user-configurable Scale: ??? pretty binary – feature is there or not – how do we represent?
Past [April 20xx] 1% Goal [Dec. 20xy] 0%

Operational Cost Efficiency Scale: <Increased efficiency (Straight through processing STP Rates )>
Cost-Per-Trade Scale: % reduction in Cost-Per-Trade
Goal (EOY 20xy, cost type = I 1 – REGION = ALL) Reduce cost by 60% (BW)
Goal (EOY 20xy, cost type = I 2 – REGION = ALL) Reduce cost by x %
Goal (EOY 20xy, cost type = E1 – REGION = ALL) Reduce cost by x%
# Example of Estimating the Value of a Technical IT System Improvement (20xx)

## TIME.HEDGE - Time for hedge execution of average-sized trade

<table>
<thead>
<tr>
<th>Ambition:</th>
<th>Reduce the average time taken from verbal agreement (&quot;done&quot;) to hedge execution of an &lt;average-sized&gt; trade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale:</td>
<td>Seconds</td>
</tr>
<tr>
<td>Past:</td>
<td>[2Q10; Region=NA] 30 seconds</td>
</tr>
<tr>
<td>Goal:</td>
<td>[2Q12; Region=ALL] 3 seconds</td>
</tr>
<tr>
<td>Business Value:</td>
<td>[Type=Revenue; Reason=Improved Hedging P&amp;L; Goal Scale=3 seconds; Region=Global] Revenue= +$1mm to +$2mm</td>
</tr>
</tbody>
</table>

## SPEED.CODE – Mean elapsed time for code changes

<table>
<thead>
<tr>
<th>Ambition:</th>
<th>Reduce the mean elapsed time for code changes from business request to end-user go live</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale:</td>
<td>Mean time in calendar days over &lt;three&gt; months</td>
</tr>
<tr>
<td>Past:</td>
<td>[2009; Market=Eurex; Task=Bond execution] &lt;60 - 90&gt; days</td>
</tr>
<tr>
<td>Goal:</td>
<td>[2Q12; Market=Eurex; Task=Bond execution] 5 days</td>
</tr>
<tr>
<td>Business Value:</td>
<td>[Type=Revenue; Reason=Earlier P&amp;L from faster time to Market; Goal Scale=5 days; Region=Global] Revenue= +$2mm to +$5mm</td>
</tr>
</tbody>
</table>

This is an example made to reason about specification standards and is not supposed to be a real spec. Just realistic.
3. Assuring that Designs give Qualities

- 10 min. = 33% of total

Solution

Past
35 Minutes

Usability

Goal
5 Minutes
4. Measure Quality Levels in Specifications with Inspection
## ROI Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Costs</th>
<th>Benefits</th>
<th>B/CR</th>
<th>ROI%</th>
<th>NPV</th>
<th>BEP</th>
<th>Cost/Person</th>
<th>Risk</th>
<th>ROA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Methods</td>
<td>$188,199</td>
<td>$4,321,798</td>
<td>23:1</td>
<td>2.196%</td>
<td>$3,554,026</td>
<td>$8,195</td>
<td>$47,050</td>
<td>52.19%</td>
<td>$4,175,664</td>
</tr>
<tr>
<td>Inspections</td>
<td>$82,073</td>
<td>$2,767,464</td>
<td>34:1</td>
<td>3.272%</td>
<td>$2,314,261</td>
<td>$51,677</td>
<td>$20,518</td>
<td>26.78%</td>
<td>$2,703,545</td>
</tr>
<tr>
<td>PSPsm</td>
<td>$105,600</td>
<td>$4,469,997</td>
<td>42:1</td>
<td>4.133%</td>
<td>$3,764,950</td>
<td>$945</td>
<td>$26,400</td>
<td>6.44%</td>
<td>$4,387,756</td>
</tr>
<tr>
<td>TSPsm</td>
<td>$148,400</td>
<td>$4,341,496</td>
<td>29:1</td>
<td>2.826%</td>
<td>$3,610,882</td>
<td>$5,760</td>
<td>$37,100</td>
<td>37.33%</td>
<td>$4,225,923</td>
</tr>
<tr>
<td>SW-CMM®</td>
<td>$311,433</td>
<td>$3,023,064</td>
<td>10:1</td>
<td>871%</td>
<td>$2,306,224</td>
<td>$153,182</td>
<td>$77,858</td>
<td>83.51%</td>
<td>$2,828,802</td>
</tr>
<tr>
<td>ISO 9001</td>
<td>$173,000</td>
<td>$569,841</td>
<td>3:1</td>
<td>229%</td>
<td>$320,423</td>
<td>$1,196,206</td>
<td>$43,250</td>
<td>98.66%</td>
<td>$503,345</td>
</tr>
<tr>
<td>CMMI®</td>
<td>$1,108,233</td>
<td>$3,023,064</td>
<td>3:1</td>
<td>173%</td>
<td>$1,509,424</td>
<td>$545,099</td>
<td>$277,058</td>
<td>100.00%</td>
<td>$2,633,052</td>
</tr>
</tbody>
</table>

### Return on Investment (ROI)

- **Agile Methods**
- **Inspections**
- **PSPsm**
- **TSPsm**
- **SW-CMM®**
- **ISO 9001**
- **CMMI®**
A Recent Example

Source Eric Simmons, erik.simmons@intel.com 25 Oct 2011
Personal Public Communication

Application of Specification Quality Control (Gilb Inspections) by a SW team resulted in the following defect density reduction in requirements over several months:

<table>
<thead>
<tr>
<th>Rev.</th>
<th># of Defects</th>
<th># of Pages</th>
<th>Defects/ Page (DPP)</th>
<th>% Change in DPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>312</td>
<td>31</td>
<td>10.06</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>209</td>
<td>44</td>
<td>4.75</td>
<td>-53%</td>
</tr>
<tr>
<td>0.6</td>
<td>247</td>
<td>60</td>
<td>4.12</td>
<td>-13%</td>
</tr>
<tr>
<td>0.7</td>
<td>114</td>
<td>33</td>
<td>3.45</td>
<td>-16%</td>
</tr>
<tr>
<td>0.8</td>
<td>45</td>
<td>38</td>
<td>1.18</td>
<td>-66%</td>
</tr>
<tr>
<td>1.0</td>
<td>10</td>
<td>45</td>
<td>0.22</td>
<td>-81%</td>
</tr>
<tr>
<td></td>
<td>Overall % change in DPP revision 0.3 to 1.0:</td>
<td></td>
<td></td>
<td>-98%</td>
</tr>
</tbody>
</table>

Downstream benefits:
• Scope delivered at the Alpha milestone increased 300%, released scope up 233%
• SW defects reduced by ~50%
• Defects that did occur were resolved in far less time on average
5a. Numeric Quality Gateways
5a. Numeric Quality Gateways Improve Quality of work

Defects/Page

80 Majors Found (~160-240 exist!)

February
April

Inspections of Gary’s Designs

“Gary” at McDonnell-Douglas
DPP (=CMM 5) Improves Quality by 10x: Raytheon

Start of Effort

The individual learning curve ??

CONC
Cost of Rework (non-conformance)

COC
Cost of Conformance

Bad Process Change

www.sei.cmu.edu/publications/documents/95.reports/95.tr.017.html
Frequent feedback and improvement assure quality

- 2 Kinds of Feedback from Stakeholders, when value increment is really exploited in practice after delivery.
- Combined with other information from the relevant environment. Like budget, deadline, technology, politics, laws, marketing changes.

- [http://rsbatechnology.co.uk/blog](http://rsbatechnology.co.uk/blog)
- Back in 2004, I was employed by a large investment bank in their FX e-commerce IT department as a business analyst.
- The wider IT organisation used a complex waterfall-based project methodology that required use of an intranet application to manage and report progress.
- However, its main failings were that it almost totally missed the ability to track delivery of actual value improvements to a project’s stakeholders, and the ability to react to changes in requirements and priority for the project’s duration.
- The toolset generated lots of charts and stats that provided the illusion of risk control, but actually provided very little help to the analysts, developers and testers actually doing the work at the coal face.
- The proof is in the pudding:
  - I have used Evo (albeit in disguise sometimes) on two large, high-risk projects in front-office investment banking businesses, and several smaller tasks.
  - On the largest critical project, the original business functions & performance objective requirements document, which included no design, essentially remained unchanged over the 14 months the project took to deliver,
  - but the detailed designs (of the GUI, business logic, performance characteristics) changed many many times, guided by lessons learnt and feedback gained by delivering a succession of early deliveries to real users.
  - In the end, the new system responsible for 10s of USD billions of notional risk, successfully went live over over one weekend for 800 users worldwide and was seen as a big success by the sponsoring stakeholders.
    “I attended a 3-day course with you and Kai whilst at Citigroup in 2006”
Value Management Process

Decompose → Develop → Deliver → Measure → Learn → Stakeholders → Values → Solutions
Identify Stakeholders
Who and what cares about the outcome of our project?
Solution Prioritization
Find, Evaluate & Prioritize Solutions to satisfy Requirements.
Evo Cycles
Decompose the winning Solutions down into smaller entities, then package them so they deliver maximum Value.
Develop
Develop the packages that deliver the Value.
Deliver
Deliver to Stakeholders
improved Value,
(not always a thing or code)
Measure Change
Measure how much the Values changed.
Learn & Change
Learning is defined as a change in behavior.
Value Management Process

- Stakeholders
- Values
- Solutions
- Deliver
- Decompose
- Develop
- Measure
- Learn
Competitive Lean QA methods to Learn
What you can do immediately

① Identify the 5 most critical qualities of your system.

② Quantify the 5 qualities.

③ For each quality,
   ① set a Current level
   ② and a Goal level
Main Take-away Points

Quality Assurance is far more than ‘test’, and it can be far more cost-effective

‘Quality’ is far more than ‘bugs’

You probably have a lot to learn, if you want real competitive quality
Thanks!

Discussion After lecture, by email, all during the conference.

Tom@Gilb.com
Mobile: +47 920 66 705
www.Gilb.com

Copy of these slides will be in Gilb.com Downloads/Slides:

http://gilb.com/tiki-list_file_gallery.php?galleryId=14
The Lean Quality Assurance Methods

• Everything ‘not adding value to the Customer’ is considered to be waste.
  – This includes:
    • unnecessary code and functionality
    • Delay in the software development process
    • Unclear requirements
    • Bureaucracy
    • Slow internal communication
  – Amplify Learning
    • The learning process is sped up by usage of short iteration cycles – each one coupled with refactoring and integration testing. Increasing feedback via short feedback sessions with Customers helps when determining the current phase of development and adjusting efforts for future improvements.
  – Decide as late as possible
  – Deliver as fast as possible
  – Empower the team
  – Build integrity in
    • separate components work well together as a whole with balance between flexibility, maintainability, efficiency, and responsiveness.
  – See the whole
    • “Think big, act small, fail fast; learn rapidly”