



# LISA '06

---

Adam Haberlach  
Google, Inc.  
December 7, 2006

# QA and The System Administrator

## Happy Users and Long Lunchbreaks

---

Adam Haberlach  
Google, Inc.  
December 8, 2006

# How did I get here?

---

## Didn't particularly study for a career in Quality Assurance

- Bachelors in Computer Engineering at Oregon State University
- Worked on-campus in a "Product Testlab"
- Joined the dotcom craziness at Be, Inc.
- Joined post-dotcom craziness at AirWave Wireless
- Attempted post-post-dotcom craziness of Law School♦
- ...and now I'm at Google

I'm a chronic organizer who doesn't complete large projects on his own, but can quickly understand what's going on and come up with ways to keep them headed in the right direction.

# Agenda

---

- What is QA? (And what isn't QA.)
- What have we done at Google?
- How you can get started with your organization
- QA Q&A

# Where does QA Fit in?

---

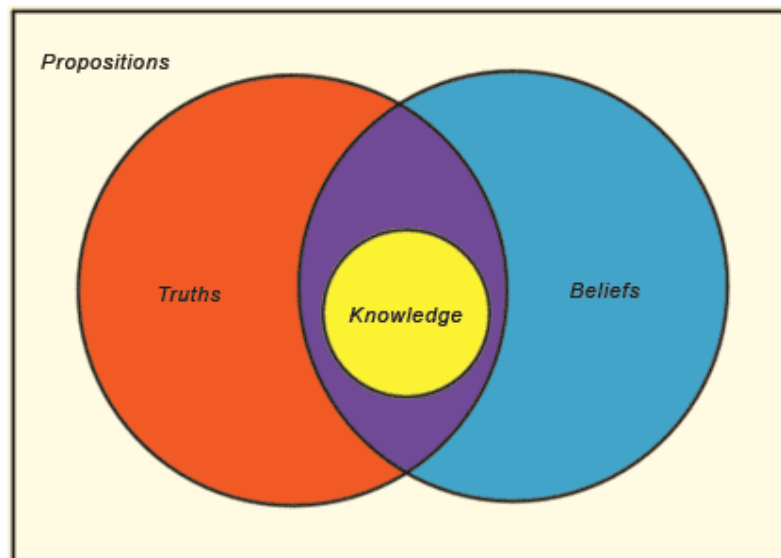
- Organize development and resources
  - but that may really be a Project Management job
- Process enforcement
  - but that may really be a Release Management job
- User Advocacy
  - but that may really be a Product Management job
- Testing Tools and Frameworks
  - but that may really be a Development job

Do I have to do everything around here?

# Quality Assurance Defined - In Abstract

---

QA: an epistemological pursuit of knowledge about the product.



# Quality Assurance Defined -- In Practice

---

## Some Answers to “What’s QA?”

- QA is how we know that we built what we said we where going to.
- Quality Assurance is there to prevent you from breaking your own service by introducing an "unwanted feature". They do this by focusing on the part of the job you find it easy to skip over: thorough and detailed testing before release.
- Quality Assurance is shipping stuff. \*

# Continued...

---

QA can't make bugs get fixed. QA can't find every bug. What QA can do is provide information - reliable, non-anecdotal information about the quality of the product. QA can't tell you whether to ship or not (though they can provide forceful opinions about this). What they can do is tell you the overall state of the project, from fit-and-finish to reliability. Decision makers go to market with the product they are comfortable with, based on this information.

QA shouldn't be a hurdle, Quality should be a gate to releasing, but QA shouldn't have to act as that gate. QA can work in a tight loop with the development team throughout product development, actually *\*accelerating\** the process of shipping quality software. If done correctly, QA is an obvious asset to a project, not a necessary evil that impedes shipping software.



# Operations QA

---

## Not So Traditional

- Dedicated resource for Corporate Sysops/Netops teams
  - Started just about a year ago
- Spun out of “Internal Systems QA” Team
  - Billing System
  - Internal HR Apps
- Viral Marketing Model, Best-Effort Resourcing

# Desktop Linux Distribution

---

Engineering time is one of our most important resources

Goal: a useful and stable desktop operating system that can provide engineers with what they need to get things done

- Support Build System
- Day-to-Day Productivity
- MP3 Playback

# Desktop Linux Distribution

---

## Tools

- Linux Testing Project (<http://ltp.sf.net>)
- QMTest (<http://www.codesourcery.com/qmtest/>)

## Release Process

- Unstable/Testing/Stable Branches
- Code Freezes

## Testlab

- One of each workstation used by Google anywhere, globally

# Server Linux / Windows Validation

---

## Linux Servers: similar to workstations, but different

- Less Hardware Variety
- Different Requirements
- More Load and Performance Focus

## Windows Workstation Validation

- Eggplant (<http://www.redstonesoftware.com/>)
  - GUI Focus
  - Uses VNC Connection
- Testlab: Again, one of every workstation

# LDAP Testing

---

## Comparison of 6 servers and several platforms

- Developed tool to playback real traffic based on logs
- Built suite of functional tests
- Set up and tore down a variety of systems

## Continued Validation of new builds

- Performance Testing/Tuning

# Internal Application

---

## Initially a Simple Ruby-on-Rails app

- “No, really, we’ve got unit tests”
- Started with 2 developers, now up to 4 -- first large project for all of them
- Integrates with LDAP, DNS, maybe more -- high risk

## QA Involvement

- Built Regression Suite
- Found innumerable UI annoyances -- served as customer proxy in Scrum Development Model

# Build Tools

---

## Not typically a Sysops function

- ...but it depends heavily on the operating system and we were testing that already
- Nobody else was doing it

## Overall Goal: to stop the “release and rollback” cycle

- A little education and organization goes a long way

# Dogfood

---

## Use your own products!

- Involved in roll-out of internal products
- First-stage triage of bugs and issues
- Pre-Dogfood-release testing

If it's good enough for customers, it's good enough for you!



# Suggestions

---

## Getting Started

- Look for long-running and stable projects
  - Does it have a development cycle?
  - Are the developers not on the same page as the users?
  - Is it more complex than one person can understand and keep sane?
- Be committed
  - Listen to QA or be vocal
  - QA is part of the design, development and release process, not just the end.

# Why bring in QA? Why not do it ourselves?

---

## Google already has a QA org

- Domain-specific knowledge
- Already have tools

## Resource allocation

- Always looking for more sysops
- (Always looking for more QA Engineers, too)

---

# QA Q&A

Adam Haberlach

[ahaberlach@google.com](mailto:ahaberlach@google.com)

(Yes, we're hiring for both QA Engineer and System Administrator)