Quality Assurance in a DevOps World

Enabling Trust

Kit Plummer

http://kitplummer.github.com
kplummer@maestrodev.com
DevOps is an ecosystem of...

- **People** (relationships)
- **Processes** (activities)
- **Product(s)**
Ecosystem...

• Applied on top of columnar organizations:
  – Development/SW eng./design/etc.
  – Quality assurance/testing/delivery
  – Product eng./operations/customers
Current DevOps thinking is...

• aligned with the Agile Manifesto,

• and prescribes itself as a **Cultural** (organizational) and **Professional** (personal) movement.
Trust is the cornerstone...

• Patrick Debois' math - based on Covey's "4 Cores":

• trust = f(character, competence)
• character = f(integrity, intent)
• competence = f(capabilities, results)
Trust is a QA deliverable...

- **Results** are the indicator
- **Testing** is the apparatus
- **Proof** of quality is the target
RATBERT, MY COMPANY IS HIRING FOR OUR QUALITY ASSURANCE GROUP. YOU'D BE PERFECT.
WHAT WOULD I HAVE TO DO?

YOU WOULD FIND FLAWS IN OUR NEW PRODUCT, THUS MAKING YOURSELF AN OBJECT OF INTENSE HATRED AND RIDICULE.

BUT THEN YOU'D FIX THOSE FLAWS... AND YOUR RESPECT FOR ME WOULD GROW INTO A SPECIAL BOND OF FRIENDSHIP, RIGHT?!

NO, THEN WE SHIP.
The Continuum (borked)
Proof == Trust
Quality Assurance

- **Product Control**
  - Requirements
  - Change
  - Release

- **Catch “bugs” before presentation**
  - Functional AND Non-Functional
Bi-Directional (required)
QA Frontend

• Agile
  – QA/Testers on team
  – Stories -> UAT
  – Continuous Integration
QA Backend

• Test harness(es)
  – Integration
  – Functional
  – Non-functional
QA’s DevOp Priorities

• Testing must be a first-class citizen
QA’s DevOp Priorities

• Testing must be a first-class citizen

“We’re not worried about testing...our biggest problem is automating deployment”
QA’s Priorities (cont.d)

• Release management
Driving Trust via Releases

• Proof (Testing)

Unit and Integration Libraries/Apps (promoted)

Functional & Non-Functional Integrated & Deployed

Functional Validated & Versioned (released)
Dev-QA-Ops Choreography
Push QA into Ops and Dev

• *DD
  – Behavior DD (Agile QA?)
  – Pulls non-functional from Ops
  – Pushes non-functional to Devs
• Less Unit, Mo Integration, Less Mock
• Automate functional, but to a point
• Ops and QA enable Dev testing
Push QA into Ops and Dev

• *DD

  “BDD has reduced our prerelease bugs by 80%, in a side-by-side comparo”
Alternative Perspective

• Building missiles
  – Platform shall have a maximum range of 100km
Validation and Verification

Modeling & Sims
Algorithms/Designs

Functional & Non-Functional
Integrated & Deployed

Functional Libraries
Apps/Services
Quality Quality

• Tests must be good
  – More to it than just assertions
• True negatives? False Positives?
  – Testing for Failure
• Brittle Tests? Avoid testing content
• Changes should come with tests
  (open sourcey way)
Truth isn’t Good Enough

- Regressions (read blame)
  - A single version change affects truth
- Measuring/Monitoring
  - Earlier the better
- Trending (continuous improvement)
  - Continuous improvement
Dev/QA Reporting
DevOps
Questions

http://devopsweekly.com/