Continuous Delivery
Bridging Quality Between Development and Customers

John Ruberto, Director of Quality Engineering, Intuit, Inc.
Great Innovative Place to Work

FORTUNE
100 Best Companies to Work For
#8  Intuit Inc.

Our 13th Year!!


FORTUNE
Most Admired Software Companies
1  salesforce.com   CA
2  Adobe Systems   CA
3  Intuit         CA
Intuit’s mission

To *improve our customers’ financial lives* so profoundly… they can’t imagine going back to the old way

**CONSUMERS**
- Quicken
- TurboTax
- mint.com

**SMALL BUSINESSES**
- QuickBooks
- Payroll
- QuickBase
- Intuit Payment Solutions

**ACCOUNTING PROFESSIONALS**
- Lacerte
- ProSeries
- Tax Online
I thought we were agile enough

• 3 Week release cycles
• Business team wanted to move faster
  – Faster innovations
  – Lots of change – complete redesign
  – Complex release environment (many products)
• We moved to Continuous Delivery
  – Refreshed technology, team, and site
  – When a story is complete – it gets deployed.
SLDC

Define

Code & Unit Test

Integration Test

System Test

Deploy
Continuous Integration

Define

Code & Unit Test

Integration Test

System Test

Deploy

✔

✔
Continuous Delivery

- Define
- Code & Unit Test
- Integration Test
- System Test
- Deploy

✔ ✔ ✔
Continuous Deployment

- Define
- Code & Unit Test
- Integration Test
- System Test
- Deploy
3-Week Release Cycle

Iteration N

<table>
<thead>
<tr>
<th>Define</th>
<th>Code &amp; UT</th>
<th>Sys Test</th>
</tr>
</thead>
</table>

Iteration N+1

<table>
<thead>
<tr>
<th>Define</th>
<th>Code &amp; UT</th>
<th>Sys Test</th>
</tr>
</thead>
</table>

3 Weeks

Deploy

Deploy
Benefits of Continuous Delivery

- Increase Velocity of Innovation
- Planning Flexibility
- Avoiding Technical Debt
- Teamwork
Use Caution

• Requires a low cost of deployment
• You control deployment
• Regulatory environment might not be compatible
• Difficult to retrofit into existing applications
• Applications that operate extensively on data
QuickBooks Support site

• Web-based support for 4 million small business users and accountants
• Design refresh across our product eco-system
• Ease is vital
• Old site:
  – Lots of tech debt (50% of effort in each release)
  – 3-week release cycle - Scrum
The Case for Change

• Quicker delivery = fewer customer frustrations
• Technology refresh was required to make this happen
  – Good for long term
  – Good for engineering team
  – Good for business
People Investments

OLD
• Structured Hand-off
• Build then test
• QA Tests
• Tests Reveal Problems
• Manual Deployments

NEW
• Collaboration
• Test First (TDD)
• Everyone Tests
• Accountability with each build
• Deployment automation built in
Technology Investments

• Ruby / Ruby-on-Rails
• Rspec: BDD Framework
• Jenkins
• Chef
• Static Analysis/style checker
• Amazon AWS
• Selenium Webdriver
Deployment Pipeline

Deployment Pipeline

Developer Machine → Integration → Pre-Production → Production

Practices

- Code Review
- Unit Test
- Accretive Test Coverage
- Static Analysis
- Automated UAT
- Performance
- Deployment Validation
- Business Decision to Release
Tips

• Make sure of business need & business is committed
• Is your system compatible with CD
• Does your team have the Agile mindset?
• Your technology should support CI & automated deployment
• Start small and build incrementally
Q&A

Contact me:

- John_Ruberto@intuit.com
- Blog.ruberto.com
- @JohnRuberto
- Linkedin.com/in/ruberto