Continuous Delivery Requires Radical Changes for Testers

Jeff “Cheezy” Morgan
@chzy
Cheezy@IndustrialLogic.ca
Continuous Delivery?
For us it means...

Every line of code written by all developers goes to production with each commit into the source code repository.

Goal is to run experiments in production with our customers in order to build better products.
Join us for Storytime
Things Change

World

Product Owner

Team
But why???

• Faster time to market

• Low risk releases

• Higher quality

• Lower cost

• Better products

• Happier teams
Isn’t this talk about Testing

- [x] Checking
- [x] Testing
- [x] Verification
- [x] Validation
There is no Testing phase
What is testing?

• Testing the behaviour of the system
• Testing the non-functional aspects of the system
• Driving risk out of the system
Testing the Behaviour
Tests at right level

- Unit Tests
- Integration Tests
- User Tests
- Exploratory

Automated

Majority are here ->

Push it down!
User Interface

Unit Test

Class

Class

Class

Class

Class

Our Services

Unit Test

Class

Class

Class

Class

Class

Backend

???
User Interface

Our Services

Backend

Not Too Many
Majority of your system will be testing with Unit Tests
40,000 to 2,000
Who writes these tests?
Don't put what you want to do before how you need to do it
What are they?

• Visual Appearance
• Accessibility
• Security
• Load / Performance
• Compliance
Team Shares
But what about Security and Capacity
Shares knowledge and helps teams maintain scripts
Risk
In the code...

- No Branches
- Feature Toggles
- Automate Database Changes
- Zero Defect Policy
Deployment Pipeline
Stage 1 runs all unit tests, static code analysis, and runs smoke tests

Stage 2 runs all acceptance tests and additional automation on one browser

Stage 3 runs subset from Stage 2 across many browsers and mobile devices

Stage 4 runs compliance, static and dynamic security tests

Stage 5 runs performance tests

Stage 6 deploys the application to production

If Stage 1 passes

If Stage 2 passes

If Stage 3 passes

If Stage 4 passes

If Stage 5 passes

Every checkin the entire application is tested and a battery of automated tests are run to ensure quality

Each time developer or tester checks in code
Separate Release from Deployment
Blue Green Deployment

- Easiest to implement
- Least Benefit
Dark Deployment

Product Owner makes release decisions
Environments

- No manual changes
- Every change is code and in source control
- All environments are identical
- Changes traverse the pipeline
Back to the tester...
• Over time developers and testers become indistinguishable
• Many orgs decide to only have developers that understand testing
• “Traditional Manual Testing” has no place
• No Automation Team
Recap

• Test the behaviour with automated tests and a little bit of exploratory testing

• Continuously test the non-functional

• No branches - use feature toggles

• Zero Defect policy

• Pipeline drives the risk out of application and environment changes
Thank You

Cheezy@IndustrialLogic.ca

@chzy