Continuous Delivery

Jez Humble
@jezhumble

http://thoughtworks-studios.com/
agile 101

The "last mile"
web 2.0

disrupting traditional businesses

Flickr was last deployed 12 hours ago, including 9 changes by 5 people.

In the last week there were 54 deploys of 636 changes by 23 people.

http://code.flickr.com/
releasing frequently

1. build the right thing

You can't just ask customers what they want and then try to give that to them. By the time you get it built, they'll want something new.

Steve Jobs
scientific method

create hypothesis

deliver **minimum viable product**

get feedback

(repeat)

---

Eric Ries, *The Lean Startup*
“How long would it take your organization to deploy a change that involved just one single line of code? Do you do this on a repeatable, reliable basis?”

Mary and Tom Poppendieck, Implementing Lean Software Development, p59.
releasing frequently

1. build the right thing
2. reduce risk of release

John Allspaw: “Ops Metametrics” http://slidesha.re/dsSZIr
releasing frequently

1. build the right thing
2. reduce risk of release
3. real project progress
agile manifesto

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Fast, automated feedback on the production readiness of your applications every time there is a change - to code, infrastructure, or configuration.
Software always production ready
Releases tied to business needs, not operational constraints
continuous delivery

automation

patterns and practices

collaboration
ingredients

configuration management

continuous integration

automated testing
Develop

Local Workstation

Mainline Server

Build

pull

Build

push

✔

Done!
Everyone Commits To the Mainline Every Day
build quality in

“Cease dependence on mass inspection to achieve quality. Improve the process and build quality into the product in the first place”

W. Edwards Deming
different kinds of testing

- **Business facing**
  - Parliamentary tests
  - Showcases
  - Usability testing
  - Exploratory testing

- **Technology facing**
  - Support programming
    - Unit tests
    - Integration tests
    - System tests
  - Critique project
    - Non-functional acceptance tests (performance, scaling, ...)

Diagram invented by Brian Marick
deployment pipeline

an automated implementation of your system’s build, deploy, test, release process

visibility

feedback

control
Deployment pipeline

- **Delivery team**
- **Version control**
- **Build & unit tests**
- **Automated acceptance tests**
- **User acceptance tests**
- **Release**

- Check in → Trigger → Feedback
- Check in → Trigger → Feedback
- Check in → Trigger → Feedback
- Check in → Trigger → Feedback
- Approval → Approval → Approval
deployment pipeline

- Commit stage:
  - Compile
  - Unit test
  - Analysis
  - Build installers

- Acceptance test stage

- User acceptance testing

- Performance testing

- Production

- Increasing confidence in build's production readiness

- Environments become more production-like

- Faster feedback
deployment pipeline

<table>
<thead>
<tr>
<th>Revision</th>
<th>Build</th>
<th>Container</th>
<th>Browser</th>
<th>UAT</th>
<th>Prod</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.33</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>manual</td>
<td>manual</td>
</tr>
<tr>
<td>1.2.32</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>manual</td>
<td>manual</td>
</tr>
<tr>
<td>1.2.31</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>manual</td>
<td>manual</td>
</tr>
<tr>
<td>1.2.30</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>manual</td>
<td>manual</td>
</tr>
<tr>
<td>1.2.29</td>
<td>auto</td>
<td>auto</td>
<td>auto</td>
<td>manual</td>
<td>manual</td>
</tr>
</tbody>
</table>

Subversion - http://chistdcrsdmo01/svn/demo/trunk/

demo #12 Made some changes to import
Revision 13 days ago by demo
reducing release risk

devops

incrementalism

decoupling deployment and release
devops

culture
automation
measurement
sharing
Low risk releases are incremental. Blue-green deployments, canary releases, feature toggles, dark launching, production immune system.
blue–green deployments
release != deployment
feature toggles

Config File

[featureToggles]
wobblyFoobars: true
flightyForkHandles: false

some.php

<?if ($wobblyFoobars) {?>
    ... various UI elements
<?}?>

other.php

$fork_handle = ($featureConfig->isOn('flightlyForkHandles')) ?
    new flightyForkHandler(aCandle) :
    new forkHandler(aCandle);

Stolen from Martin Fowler [http://martinfowler.com/bliki/FeatureToggle.html](http://martinfowler.com/bliki/FeatureToggle.html)
Dark Launching
enterprise governance

- risk management
- SOX, ITIL, COBIT
- segregation of duties
- change management
- auditing and compliance
people are the key

inceptions, showcases, retrospectives

rotate people, share the pain

big, visible displays

continuous improvement (kaizen)
questions

jez@thoughtworks.com
http://continuousdelivery.com/

@jezhumble #continuousdelivery