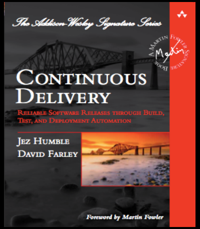
An Introduction to Continuous Delivery


The Addison-Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

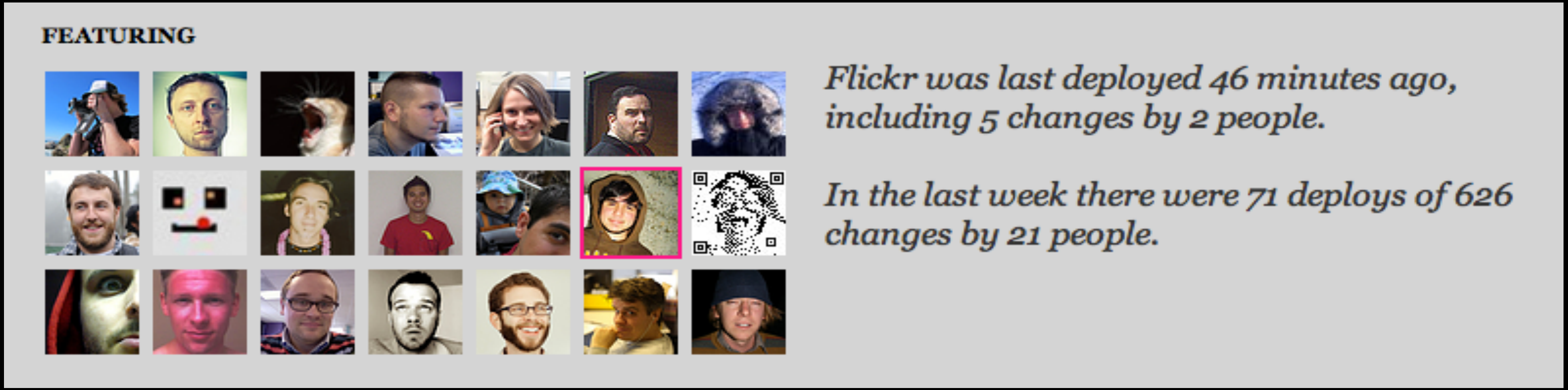
rolf russell
continuous delivery practice lead

ThoughtWorks®

LAST MILE
ROAD

conan the deployer

getting it in front of users quickly

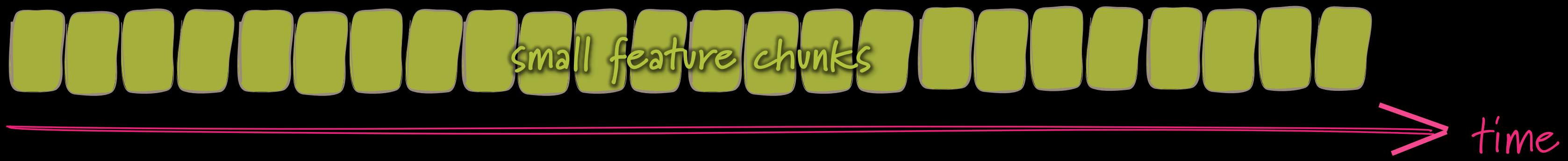


FEATURING

Flickr was last deployed 46 minutes ago, including 5 changes by 2 people.

In the last week there were 71 deploys of 626 changes by 21 people.

http://code.flickr.com

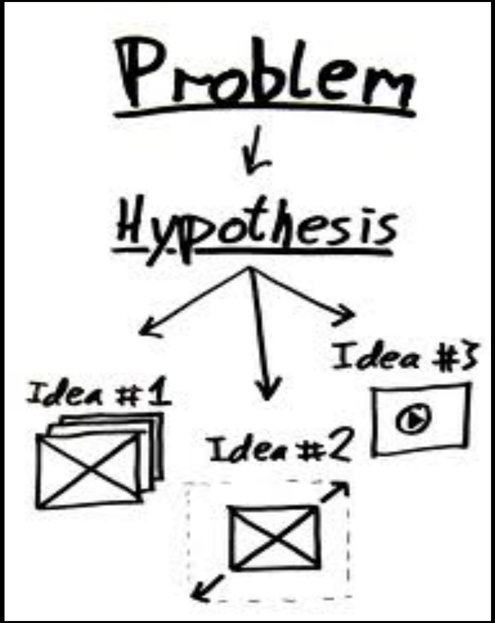


constant flow of new features into production

incremental release of small changes

Why

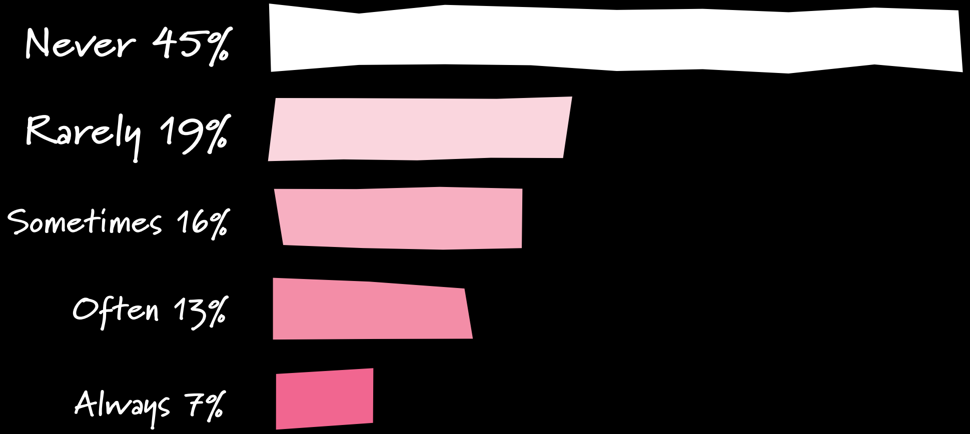
build the right thing

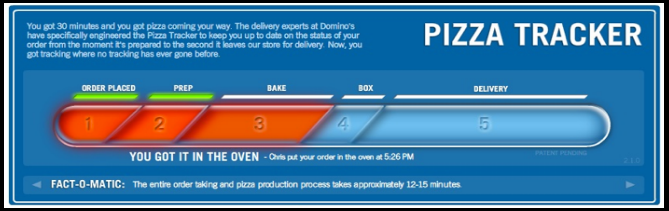


every business idea is a hypothesis until you get user feedback

corollary: don't waste money on the wrong thing

Standish Group: how often features are used





PIZZA TRACKER

You got 30 minutes and you got pizza coming your way. The delivery experts at Domino's have specifically engineered the Pizza Tracker to keep you up to date on the status of your order from the moment it's prepared to the second it leaves our store for delivery. Now, you got tracking where no tracking has ever gone before.

YOU GOT IT IN THE OVEN - Chris put your order in the oven at 5:26 PM

FACT-O-MATIC: The entire order taking and pizza production process takes approximately 12-15 minutes.

constant user connection

by releasing everyday:

your users can be delighted by new stuff all the time

your users get the feeling you are reacting to what they want

ability to move quickly

react to the market

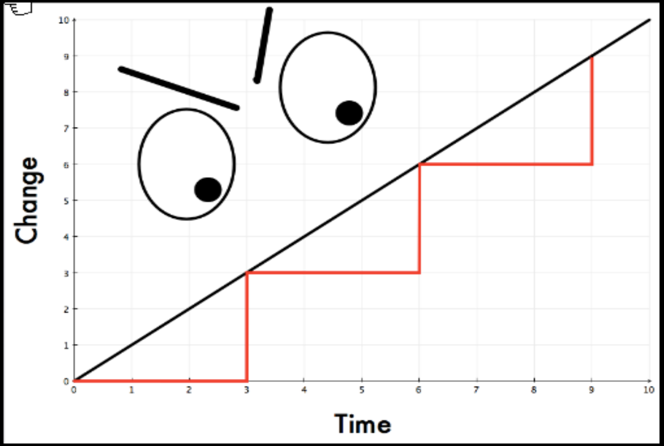
explore new revenue streams

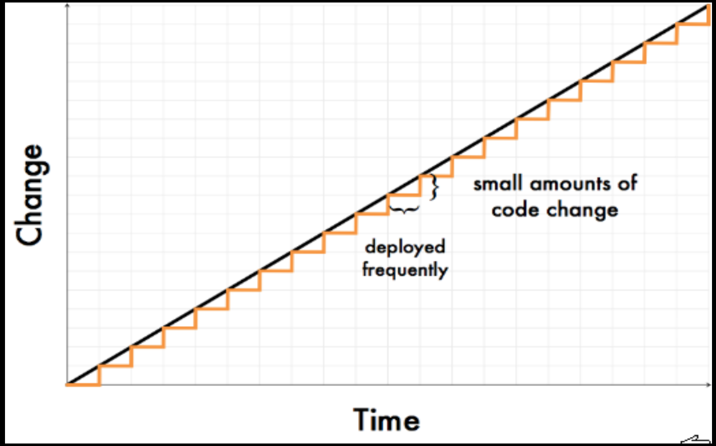
better aligned people

development & operations close to the business

IT no longer perceived as
the bottleneck

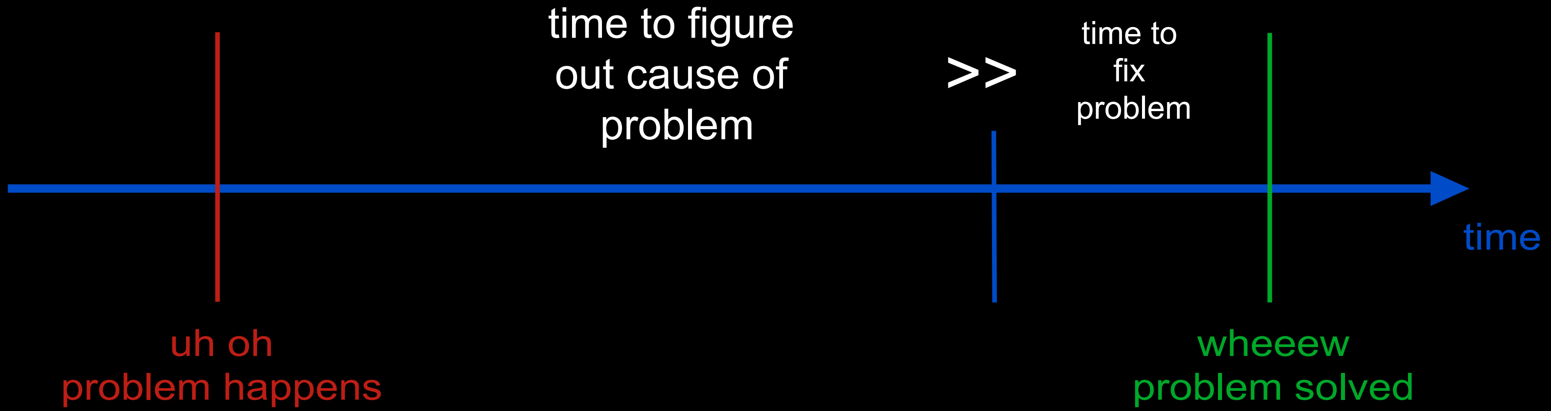
reliability & stability





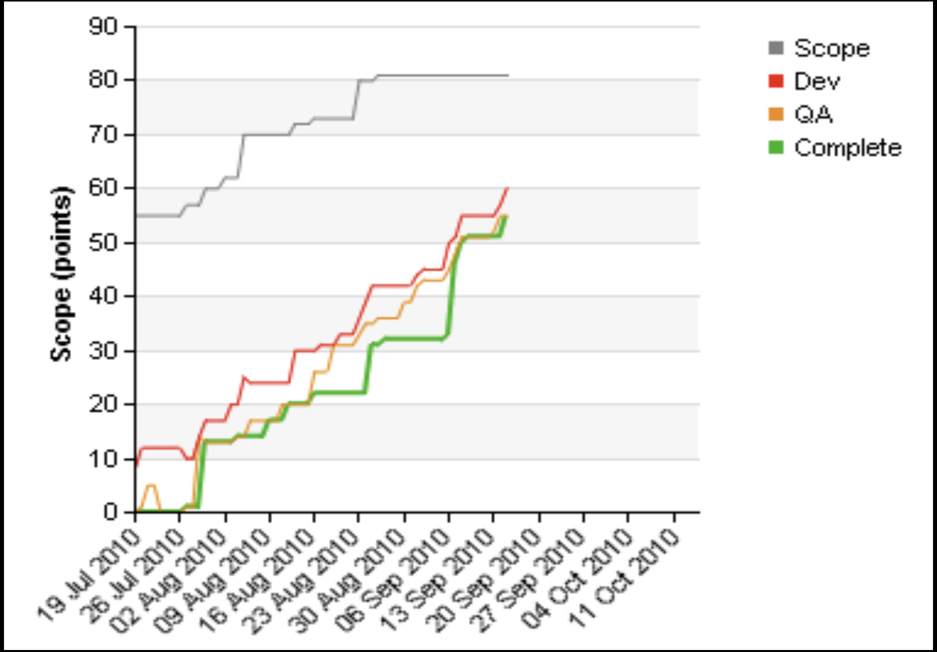
John Allspaw: “Ops Metametrics” http://slidesha.re/dsSZIr

TTR (time to recover)



adapted from John Allspaw

progress


Scope (points)
0
10
20
30
40
50
60
70
80
90
19 Jul 2010
26 Jul 2010
02 Aug 2010
09 Aug 2010
16 Aug 2010
23 Aug 2010
30 Aug 2010
06 Sep 2010
13 Sep 2010
20 Sep 2010
27 Sep 2010
04 Oct 2010
11 Oct 2010
Scope
Dev
QA
Complete

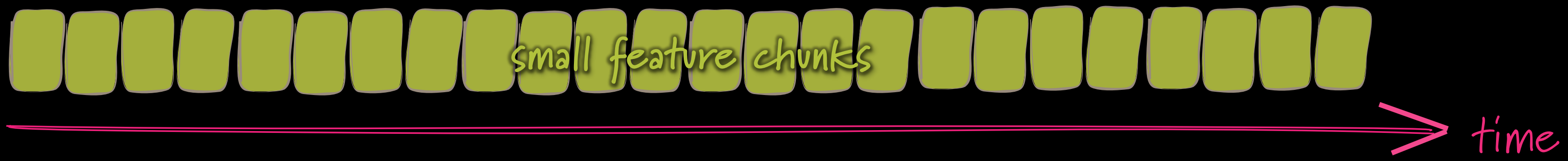
How

fast, automated feedback on the production readiness of your applications every time there is a change

whether code, infrastructure, configuration or database

Jez Humble

continuous delivery



software always production ready

releases tied to business needs, not IT constraints

minimize the lead time from idea to live

concept to cash

systems thinking is [a philosophy] based on the belief that the component parts of a system can best be understood in the context of relationships with each other and with other systems, rather than in isolation.

value stream mapping

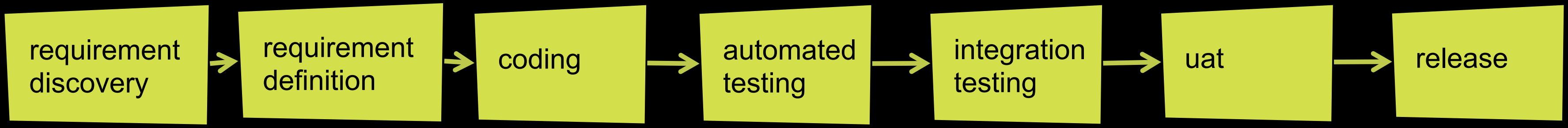
process tool for improving **the flow from a customer request to the fulfillment** *(concept to cash)*

emphasis is on **improving the whole not just the parts**

uses **quantitative data** to identify waste and inefficiency

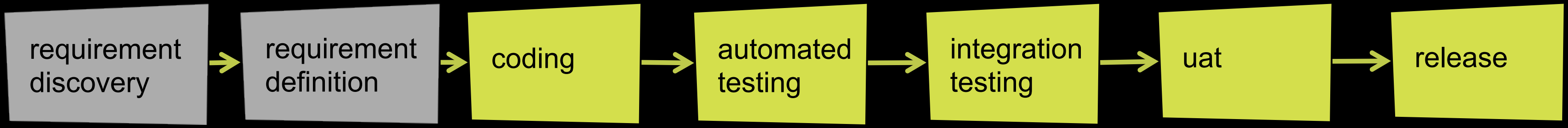
originally developed by toyota to assist in the improvement of manufacturing and supply-chain processes

value stream mapping



lead-time	4 wks	4 wks	6 wks	1 wk	4 wks	1 wk	1wk
value-add-time	2 days	1 wk	4 wks	2 days	1 wk	1 day	4 hrs
complete & accurate	30%	50%	25%	40%	80%	90%	80%

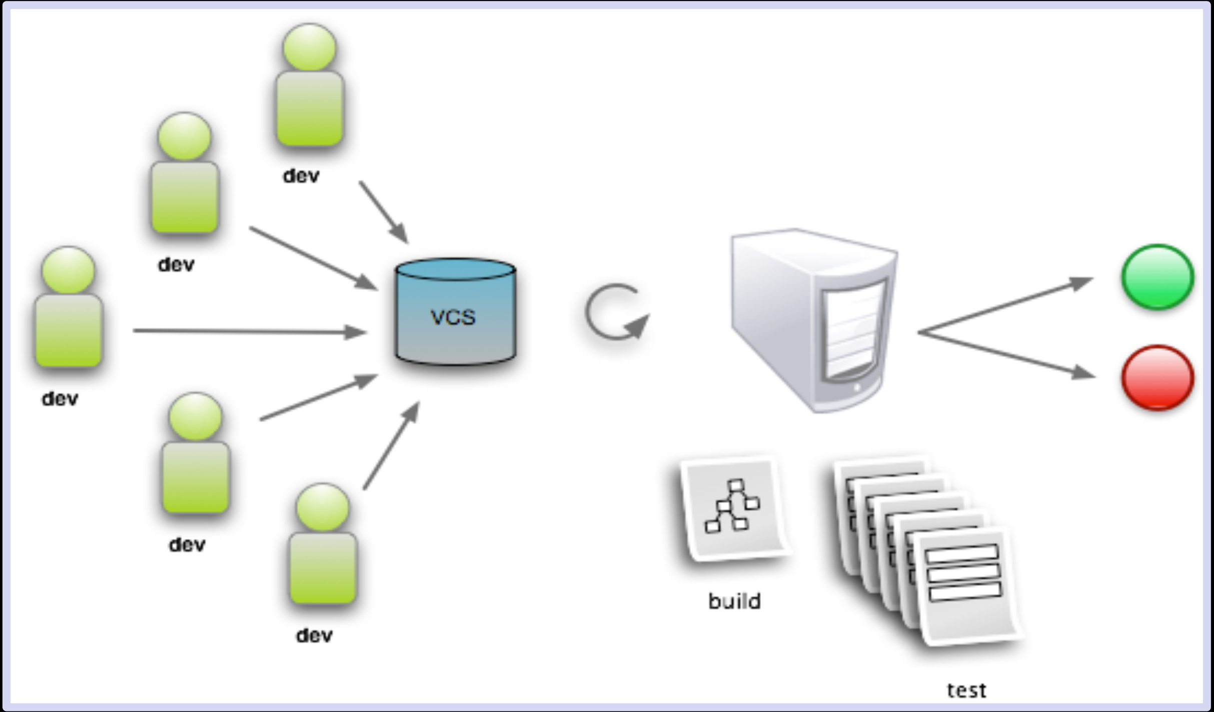
value stream mapping



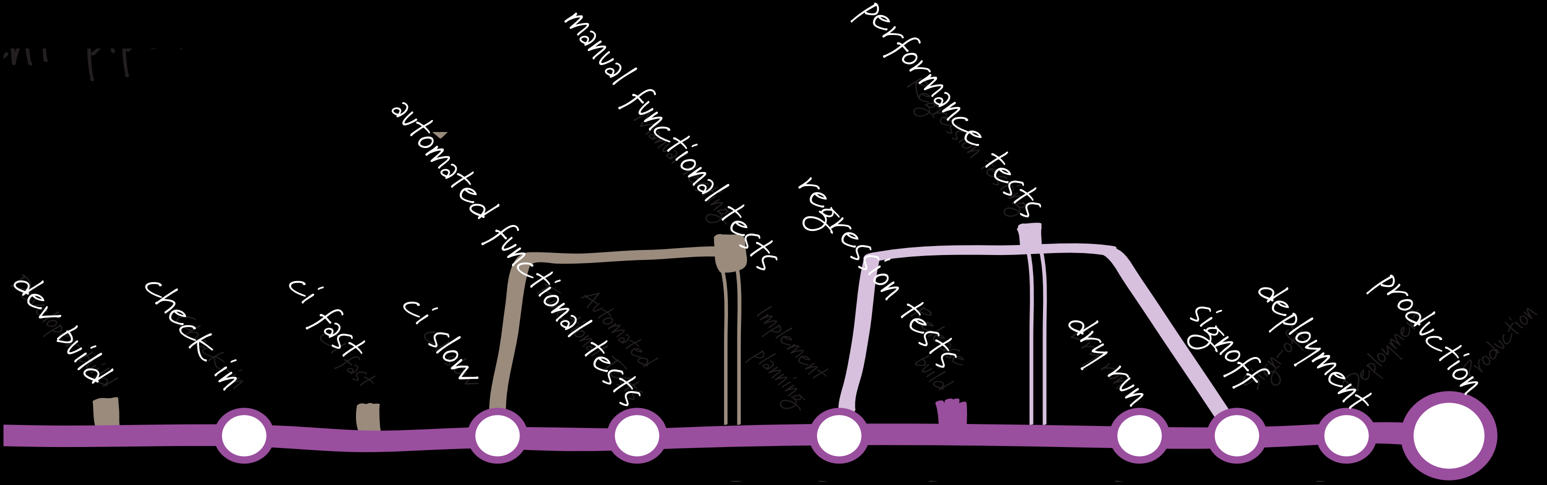
lead-time	4 wks	4 wks	6 wks	1 wk	4 wks	1 wk	1wk
value-add-time	2 days	1 wk	4 wks	2 days	1 wk	1 day	4 hrs
complete & accurate	60%	50%	25%	40%	80%	90%	80%

```
while (true) {
      if (change checked into vcs) then build & test
      sleep 60
}
```

step 1 - continuous integration



full production pipeline



faster feedback

full production pipeline

automated implementation of your system's
build, deploy, test, approval processes

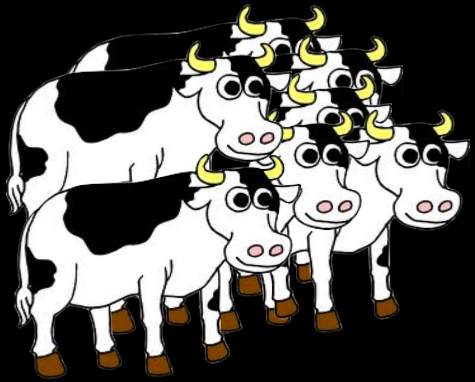
- visibility
- traceability
- compliance

treat everything like code

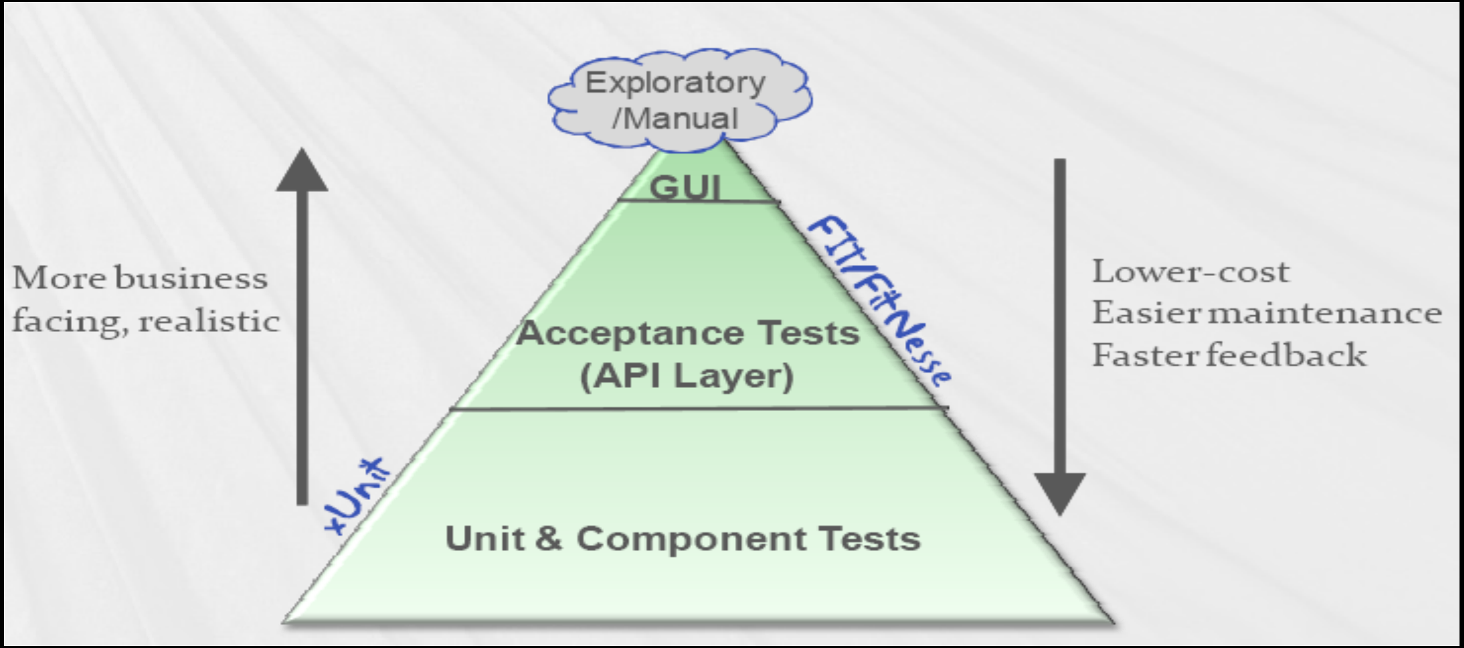
check in, automate, test in CI, promote in deployment pipeline

- database: DDL & static data
- deployment automation
- infrastructure/configuration mgmt
- monitoring configuration

treat servers like cattle, not pets

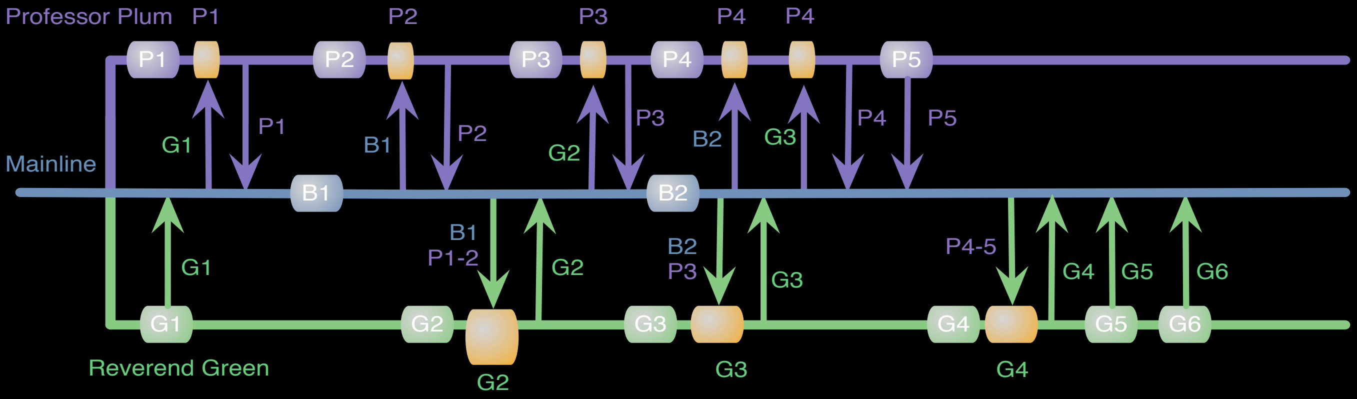


adhere to the test pyramid



Adapted from Mike Cohn (Automated Test Pyramid)
and Lisa Crispin & Janet Gregory (Agile Testing)

trunk based development



branches discourage refactoring

branches delay integration and hide risk

merging wastes time and is tedious

trunk based development

feature toggles let you deploy incomplete features turned off

branch-by-abstraction lets you make architectural changes

consistency from development to production

accidental inconsistency >> necessary inconsistency

deployment process

environment configuration

testing tools

pull itops onto the delivery team

sit together: biz, dev, qa & sysadmin

share KPIs for stability and change

same story wall and iterations

“in production”? “live”?

what does “in production” mean today?

what does “live” mean? is it a binary state?

how can we take advantage of shades of grey in “live”?

concerns

reliability & stability

compliance & traceability

releasing 10 times/day

- don’t need to, just keep your code always production releasable

complexity of my systems

- its about continuous improvement. start with low hanging fruit

it will take investment

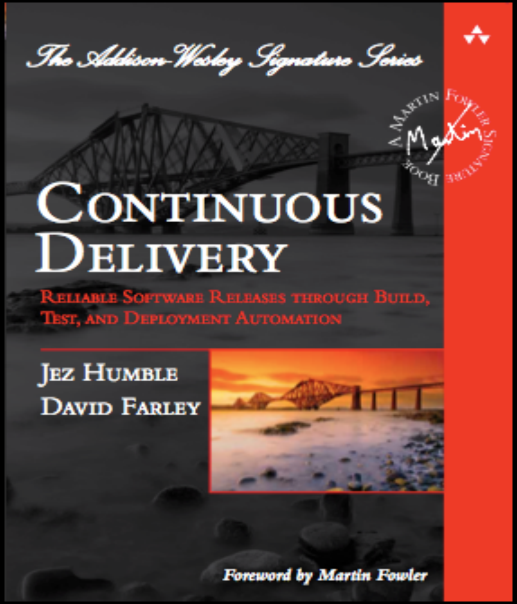
- yes it will, but it will also start paying dividends quickly

homework

how long would it take your organization to get a one line code change into production using the normal process?

Q&A

Thank you!


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

The Lean Startup
How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses
Eric Ries

ThoughtWorks®

extra credit: lean startup movement

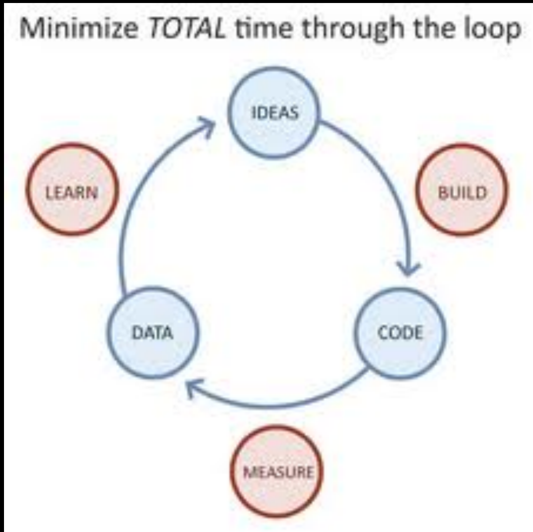
approach to managing disruptive innovation

goal of startups is to learn

- true for everyone early in the innovation cycle

learning should not be adhoc

- be rigorous & use the scientific method
- hypothesis -> experiment -> analysis


Minimize *TOTAL* time through the loop
IDEAS
LEARN
BUILD
DATA
CODE
MEASURE