



IS4301 Agile IT with DevOps – Lecture 9

Adjunct Professor Foong Sew Bun

Department of Information Systems and Analytics

National University of Singapore

Learning Objectives

At the end of this lecture, you will understand:

- Introduction of DevOps
- Overview of Continuous Deployment, Continuous Delivery, Continuous Integration
- What is Continuous Delivery?
- Value of continuous delivery
- Principles of Continuous Delivery



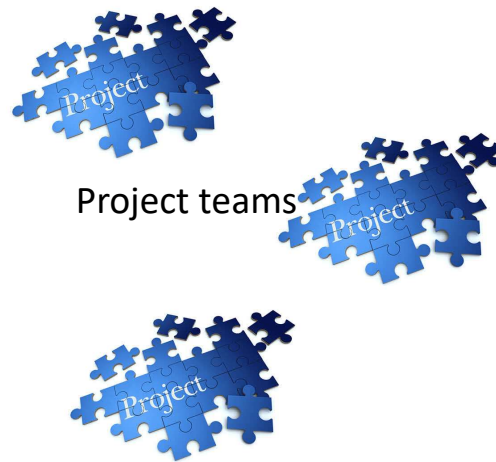
State of Enterprise Project Development and Operations



Business

Engineering

IT Operations



Infrastructure team

QA

DBAs

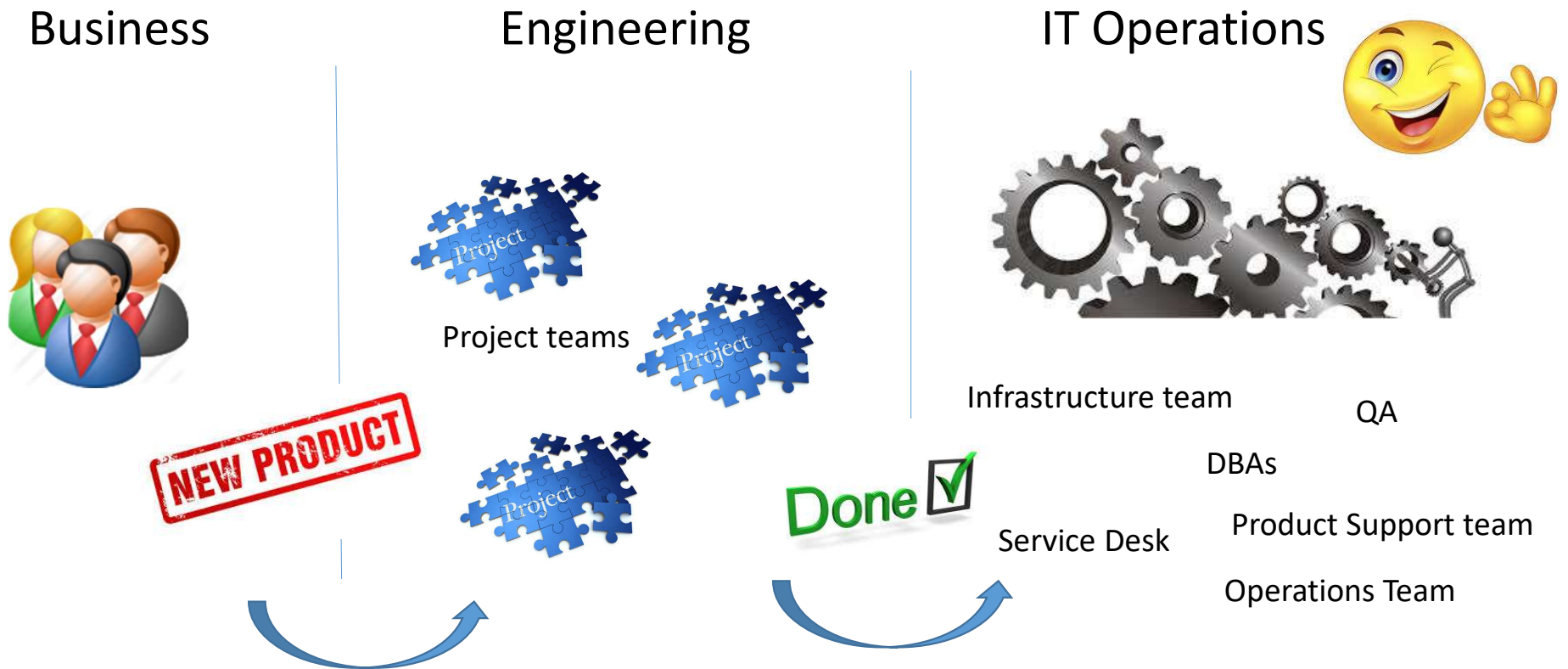
Service Desk

Product Support team

Operations Team



State of Enterprise Project Development and Operations



State of Enterprise Project Development and Operations

Business



Engineering



IT Operations



Infrastructure team

QA

DBAs

Service Desk

Product Support team

Operations Team

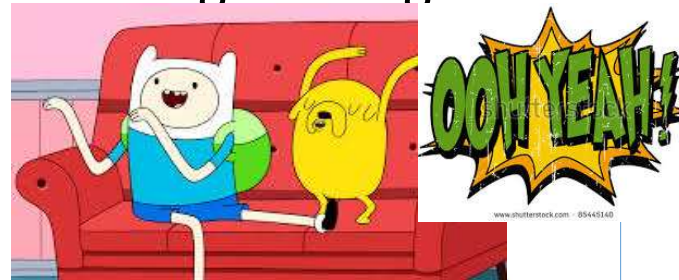
What Happens: Product Creation Taking too Long – Go Agile



Business



Engineering



Project teams

->

Agile Sprint
Teams



IT Operations



Infrastructure team

QA

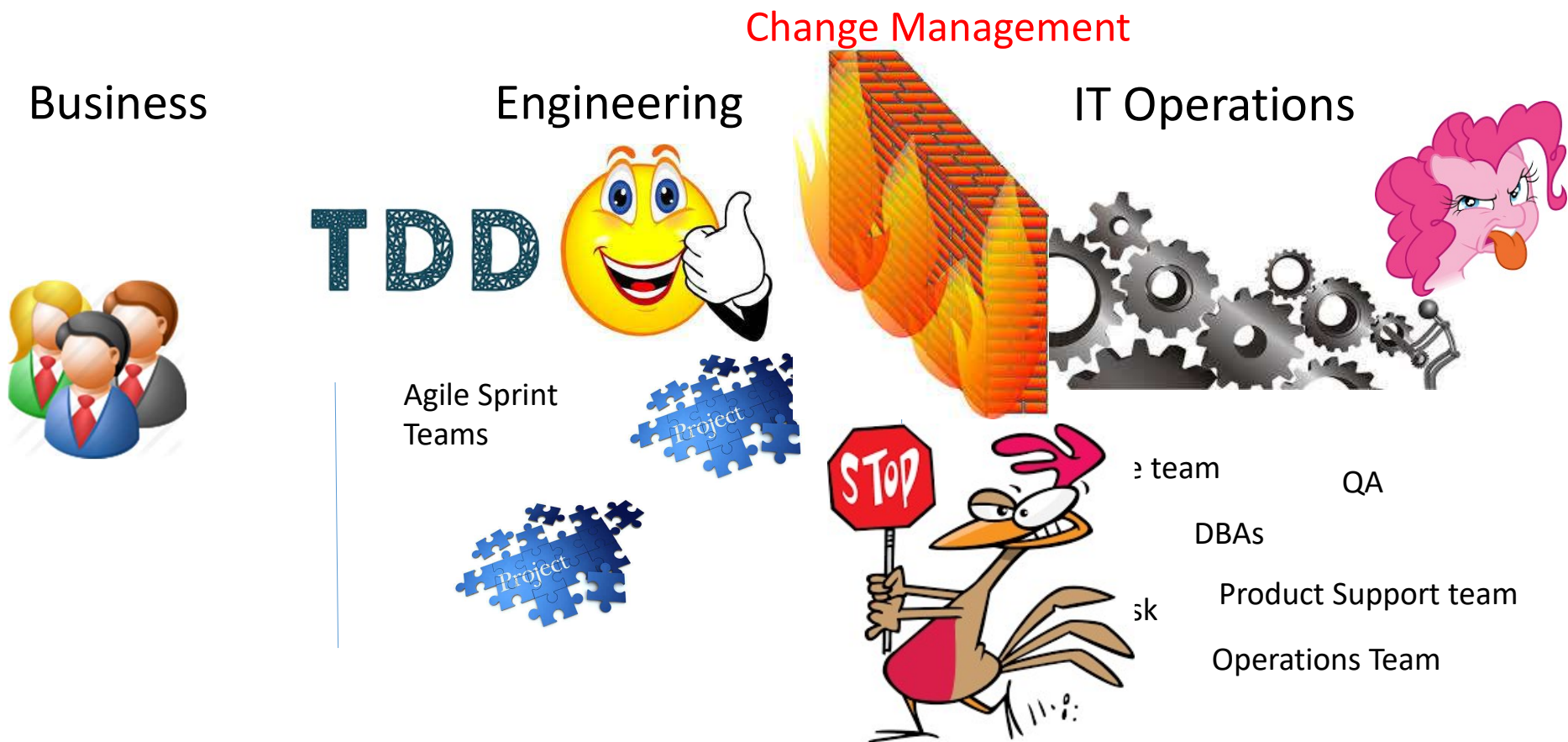
DBAs

Service Desk

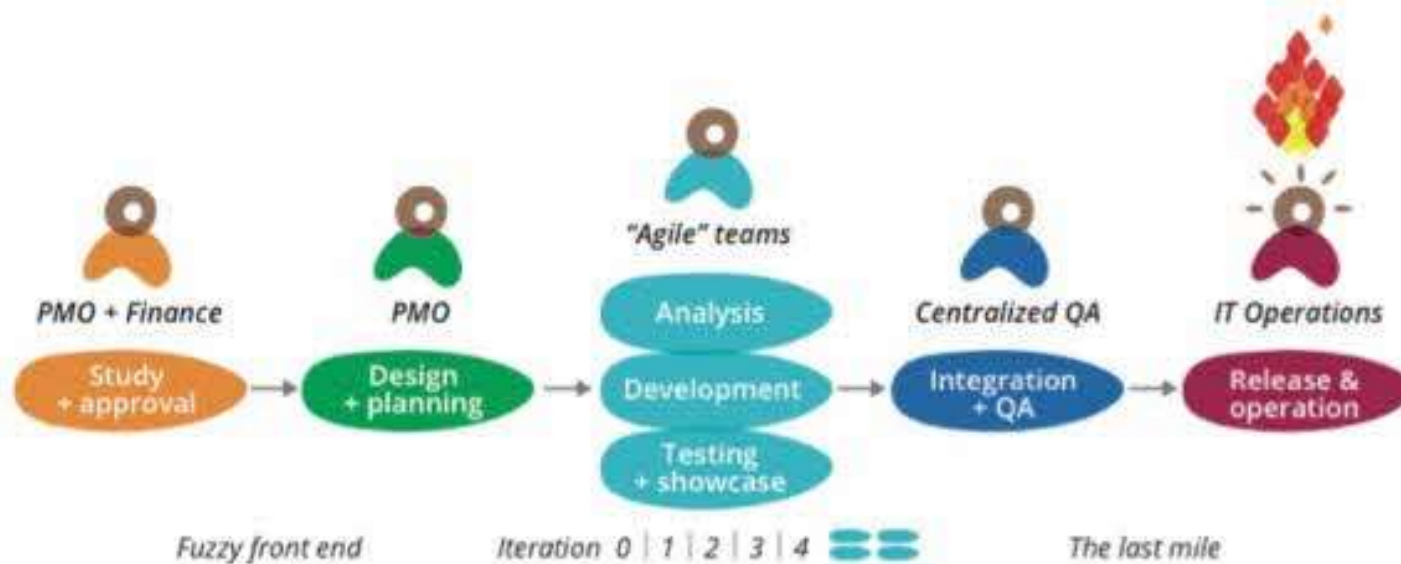
Product Support team

Operations Team

What Happens: Product Creation Taking too Long – Go Agile



WATER - SCRUM - FALL



Water-

Scrum-

Fall

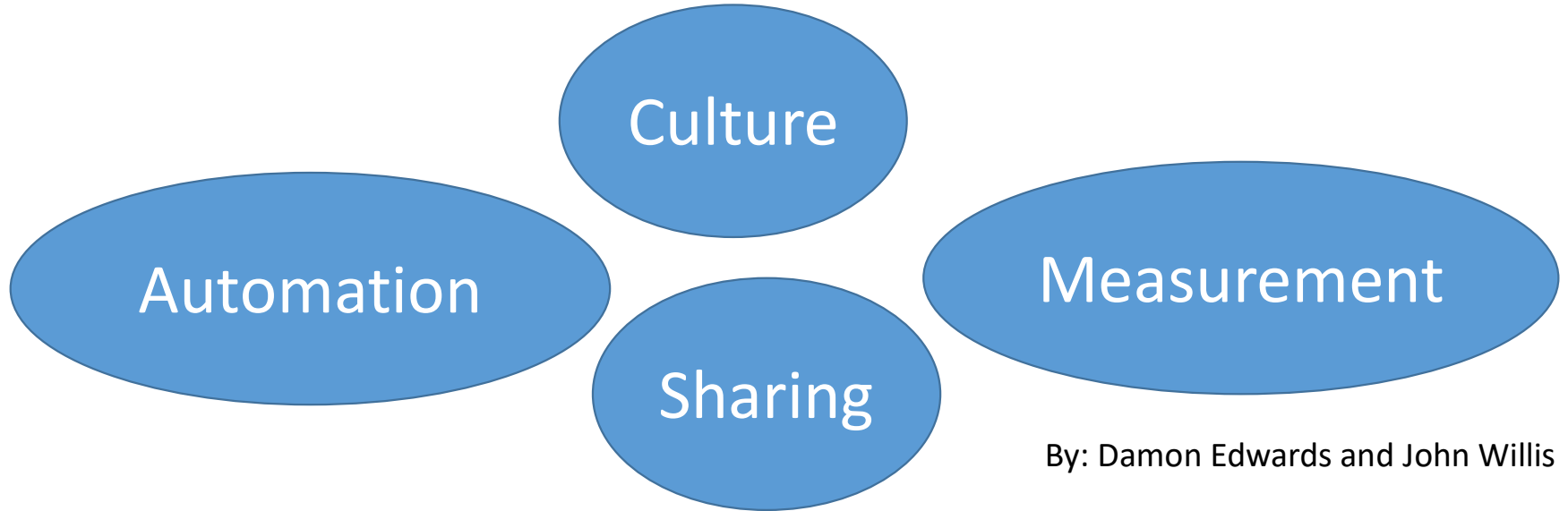
Challenges Faced by IT Operations

- Maintaining Legacy applications, heterogeneous platforms
- Prefers evolutionary architectural changes
- New software and services hard to accommodate
- Spend time fire fighting and troubleshooting
- Conservative, process heavy and driven
- Typically 80% of IT budget spent on operations



DevOps Movement – still evolving

- Not a fixed methodology or process
- Cross functional Community of Practice and set of principles
- What is DevOps? [CAMS](#)



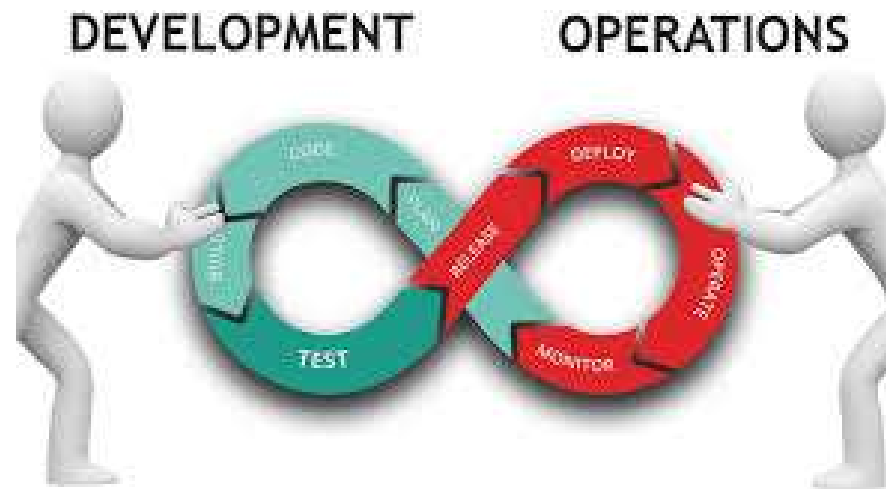
By: Damon Edwards and John Willis

Top Predictions of IT Performance

- Peer-reviewed change approval process
- Version control everything
- Proactive monitoring
- High trust organizational culture
- Win-win relationship between dev and ops



Dev and Ops Collaboration



Supports changes
Alerts broadcasted if software failed
Creates deployable software

Access to requirements tool
Part of sprint team
Participates at sprint kickoffs, retrospectives, demos

Measuring IT Performance

- Throughput of code
 - Deployment frequency – how frequently a team is able to deploy code
 - Change lead time – how fast it can move from committing code to deploying it
- Stability of system
 - Mean time to recover (MTTR) – how quickly can a system recover from downtime
 - Change failure rate – how many changes succeed versus how many fail



Lead Time

“How long would it take your organization to deploy a change that involves just a single line of code? Do you do this on a repeatable, reliable basis?”

-- *Mary and Tom Poppendieck*

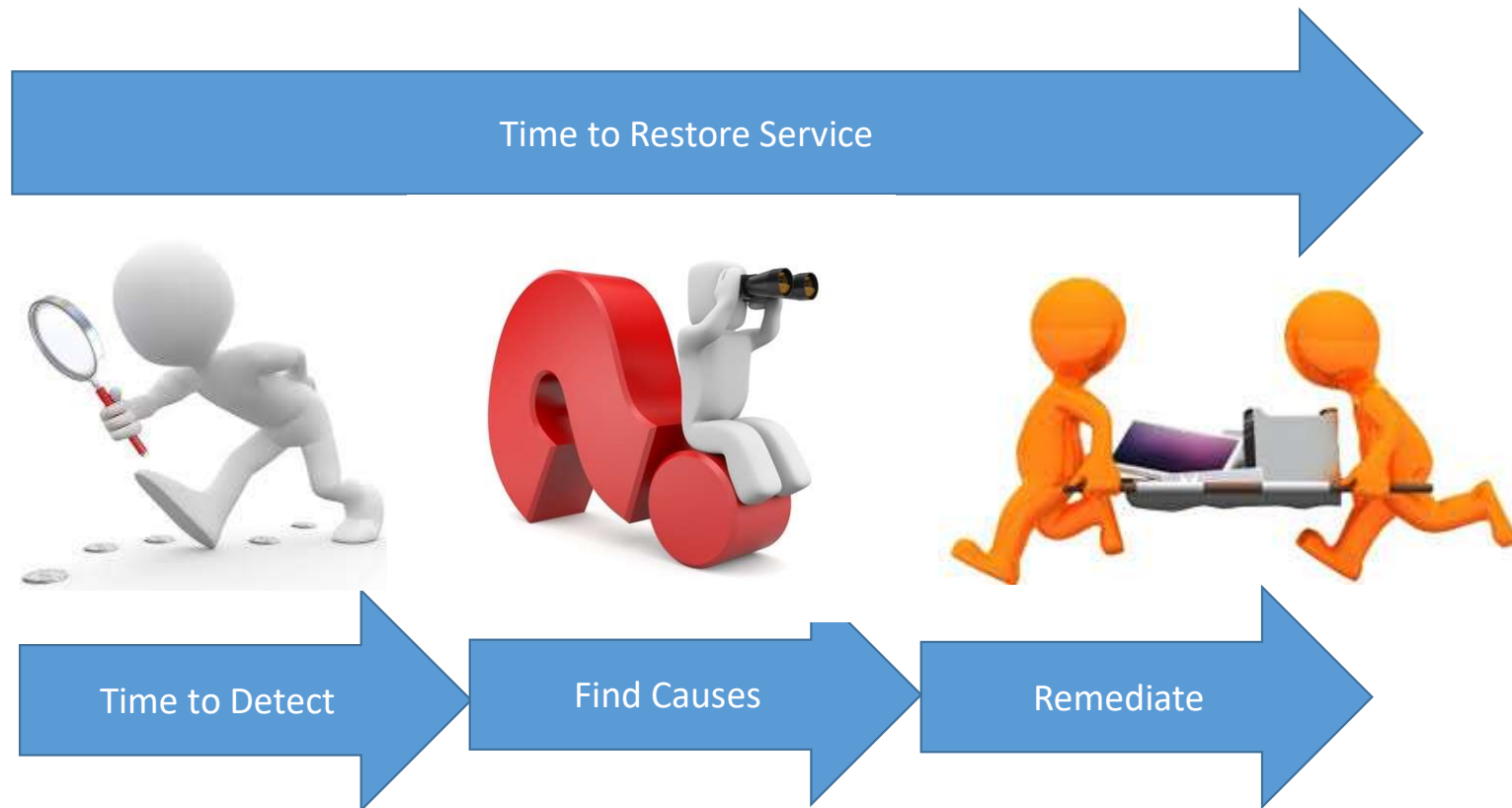


Lead Time

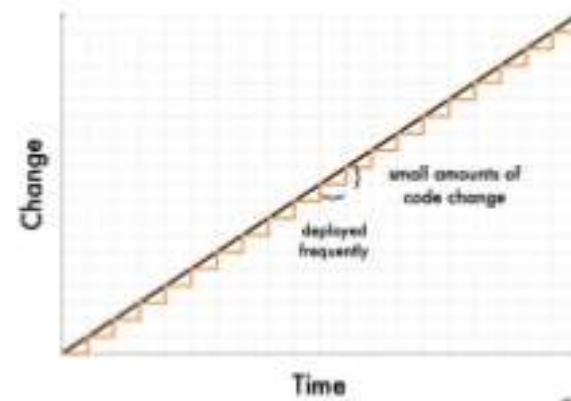
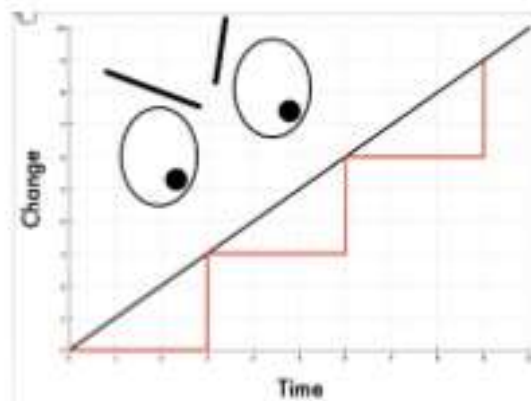
- How quickly can I restore my services?
- How quickly can I get a critical fix to users?
- How quickly can I validate whether a feature is valuable?



Cycle for Restoring Services



working in small batches



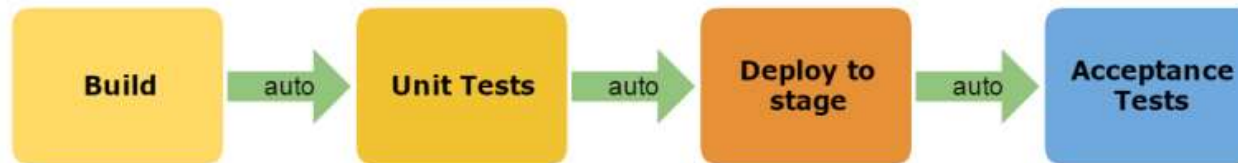
John Allspaw: "Ops Metametrics" | <http://slidesha.re/dsSZlr>

@jeshumble | chief software

Continuous Integration and Continuous Delivery (CI/CD)

Continuous Integration

Integrate early and frequently



Continuous Delivery

Software is always deployable



Continuous Deployment

Deploy as the final stage of CI





12 Principles Behind Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Amazon May 2011 Deployment Stats

11.6 seconds

Mean time between deployments (weekday)

1,079

Max# of deployments in a single hour



<http://assets.en.oreilly.com/1/event/60/Velocity%20Culture%20Presentation.pdf>

Amazon May 2011 Deployment Stats

10,000

Mean# of hosts simultaneously receiving a deployment

30,000

Max# of hosts simultaneously receiving a deployment



<http://assets.en.oreilly.com/1/event/60/Velocity%20Culture%20Presentation.pdf>

Why Microservices?

75,000

production deploys
per year

Services

900

developers

300

services

4k

deploys per week



Necessary Ingredients for Continuous Delivery

- Comprehensive configuration management
- Continuous integration
- Automated testing



Practices in Continuous Delivery

- “Our code, app configurations and system configurations are in version control system.”
- “We get failure alerts from logging and monitoring systems”
- “Developers merge their code into trunk daily”
- “Developers break up large features into small, incremental changes.”
- “Development and operations interact, outcome is generally win/win”



Tutorial: Gitflow as an simple example of a CI/CD tool

- What is Git?
 - Distributed version control system
 - Merge changes among branches and repositories
- What is Gitflow?
 - Git level extensions to provide high level repository operations for Vincent Dresissen's branching model

