



# SSW-555: Agile Methods for Software Development

## DevOps

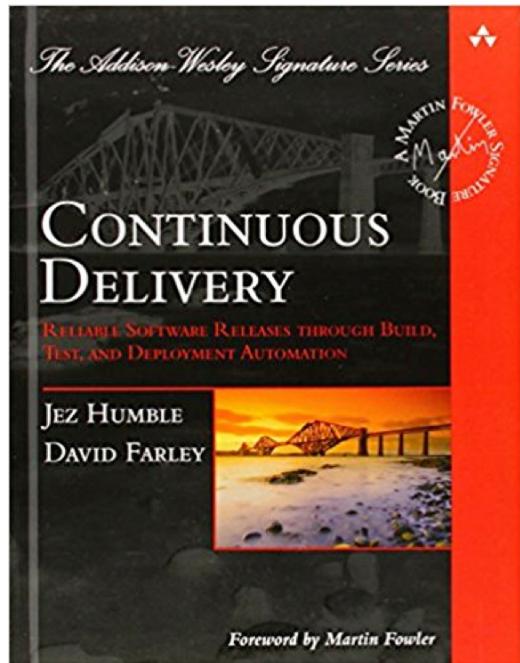
Dr. Richard Ens  
Software Engineering  
School of Systems and Enterprises





# Acknowledgements

Some of the material in these slides is from a talk by David Myers



# Today's topics

What is DevOps?

What problem are we trying to solve?

Evolution from Waterfall to Agile to DevOps

How is DevOps related to Agile?

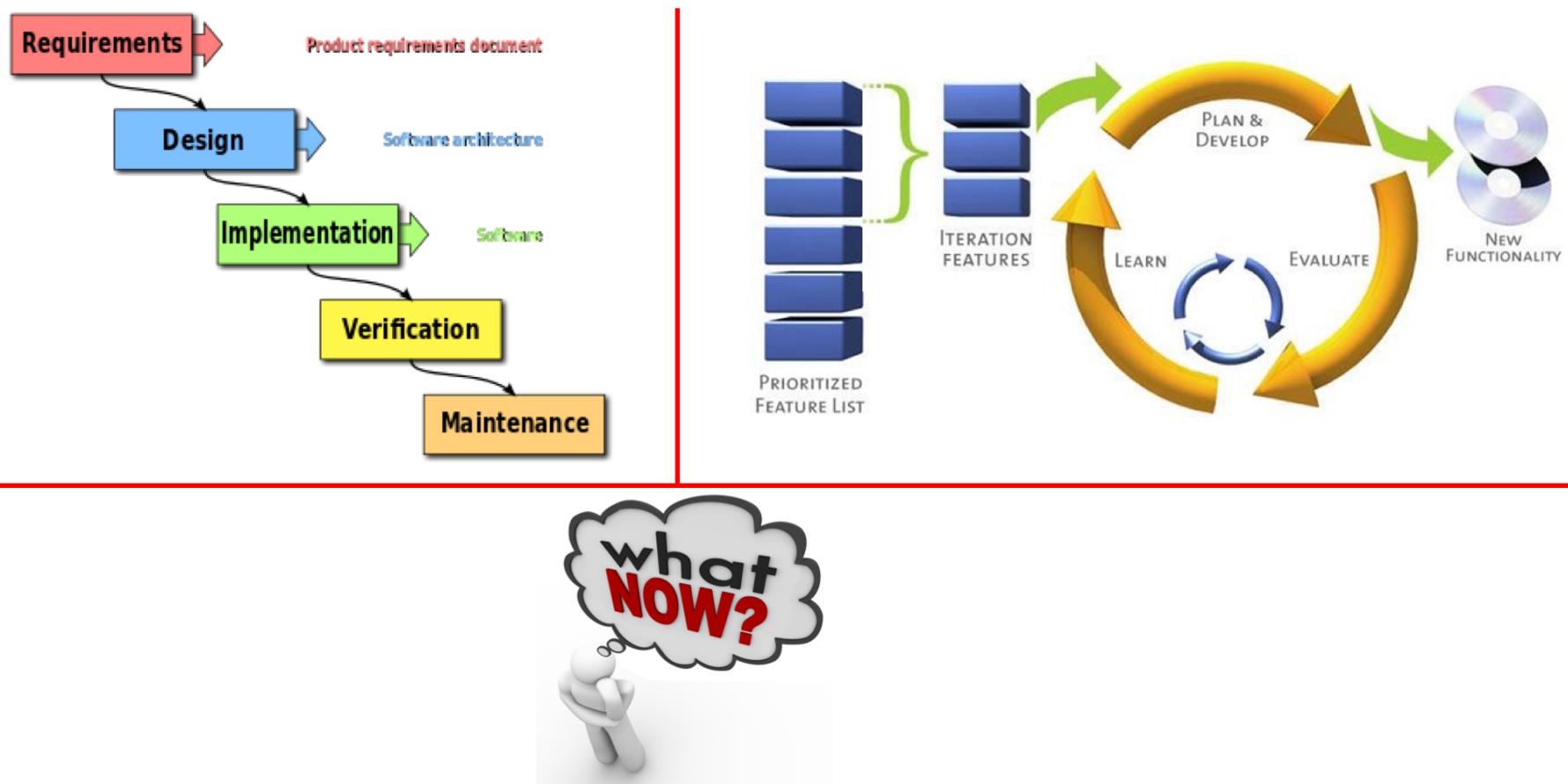
DevOps principles

DevOps tools



# SDLC evolution: What now?

We improved the Waterfall model with Agile Methods



# Delivering value to the customer

Agile methods help to build new features more quickly

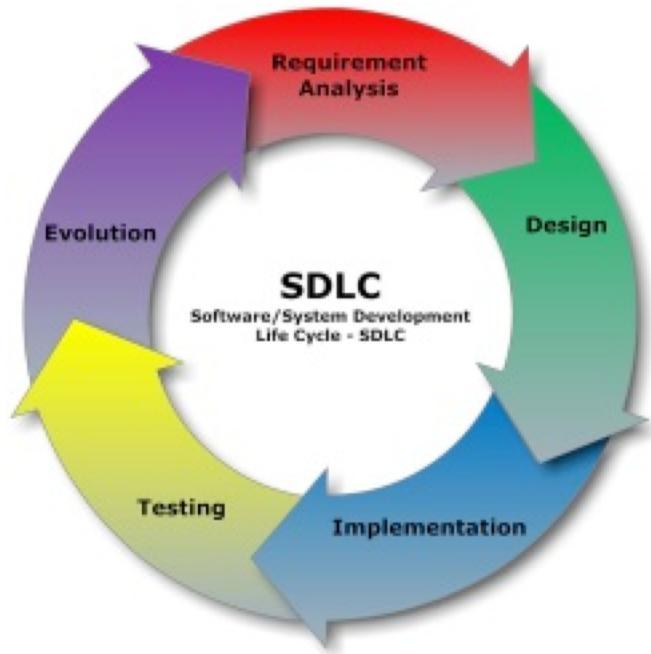
But, we don't achieve value until we deliver to customers

How to deliver those features to customers more quickly?



# Development vs. Deployment/Ops

## Development



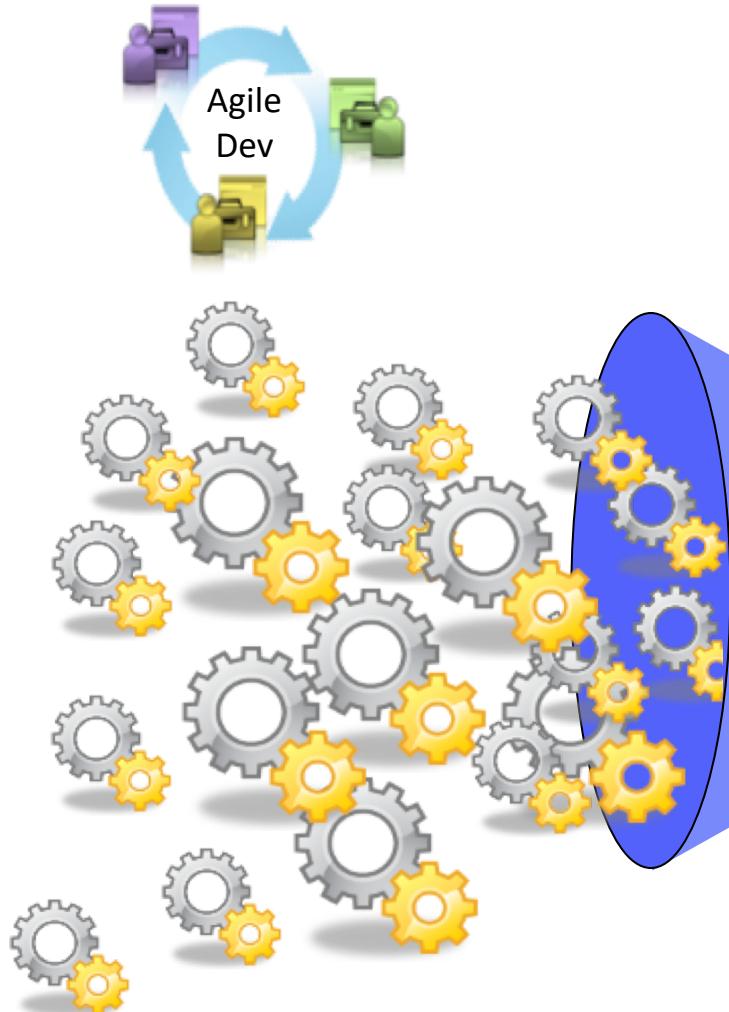
## Deployment/Ops



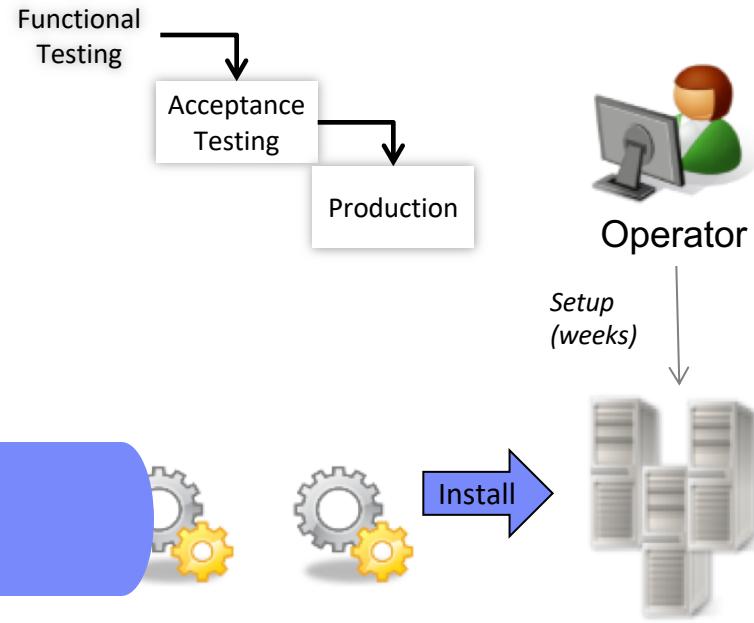
Source: <http://softwaredeploymentplan.blog.com/>

Source: [http://www.matrikasoftware.com/implementation\\_deployment.html](http://www.matrikasoftware.com/implementation_deployment.html)

# Agile is not enough...

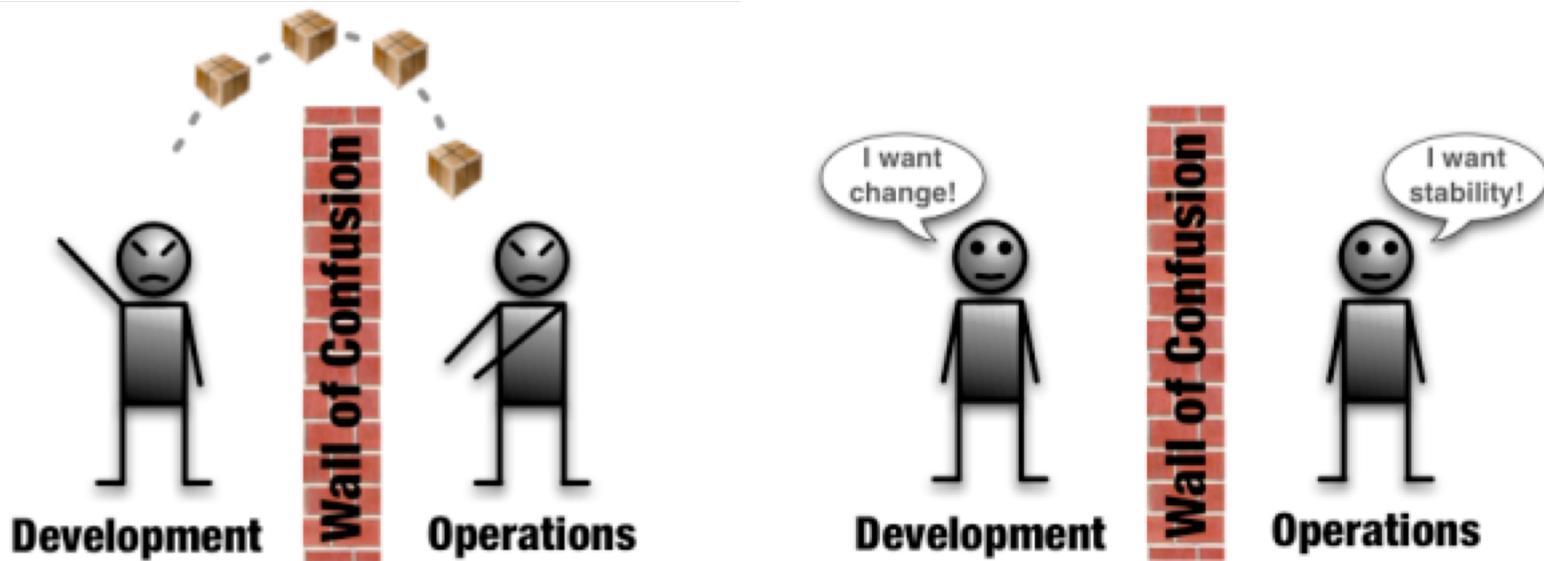


Agile builds are piling up



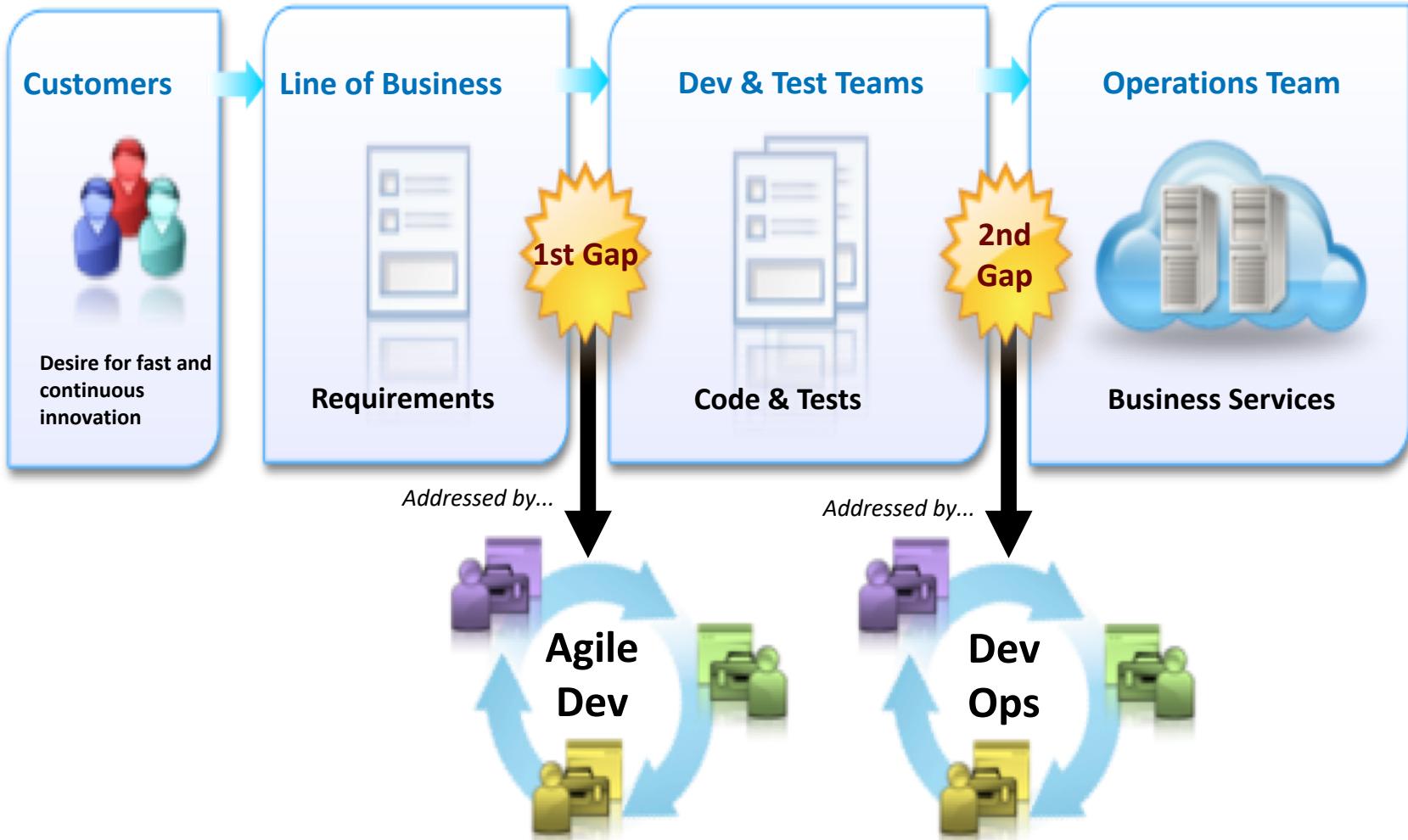
*Test and Ops teams have increased pressures to keep up with increased loads but continue to use waterfall approaches and traditional tools.*

# Application Lifecycle Management

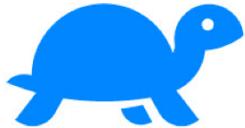


Source: <http://dev2ops.org/2010/02/what-is-devops/>

# Siloed delivery challenges



# How often to release new features?



**slow**

Operating systems,  
mission critical systems

DSDM and SAFe minimize  
customer disruption by  
pacing releases

How frequently can the  
customer absorb new  
releases?



**fast**

Web based services

Spotify: weekly releases

Feature toggles and  
Limited blast radius

Amazon: 50M  
deployments in 2014  
(1.5/second)

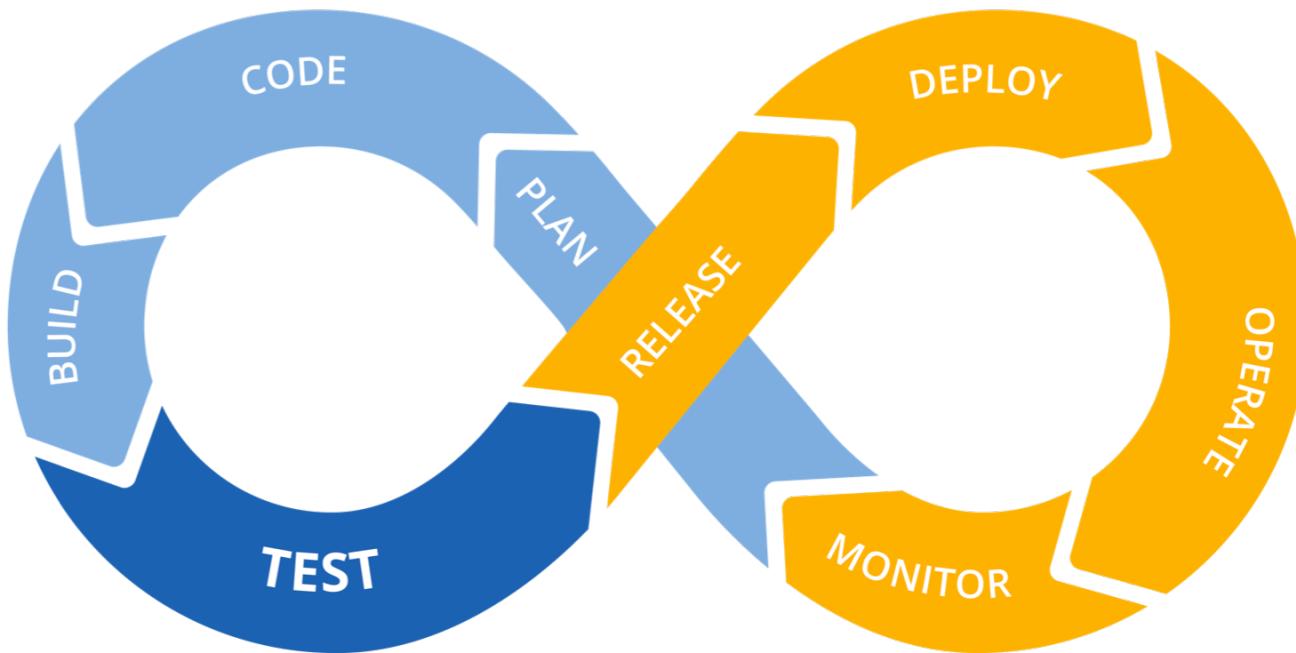
Netflix, Facebook



# DevOps Solution

Tear down the silos!

Re-integrate IT to deliver business value faster and better



# DevOps is...

A set of principles, values, and tools  
that facilitate collaboration across  
disciplines to...

1. Enable rapid evolution of deployed business services
2. Reduce risk, decrease cost, and improve quality

***People***



***Process***



***Tools***



# DevOps principles and values

Collaborate across disciplines

Requirements, development, deployment, operations

Develop and test against a production-like system

Iterative and frequent deployments using repeatable and reliable processes

Continuously monitor and validate operational quality characteristics

Amplify feedback loops

*People*



*Process*

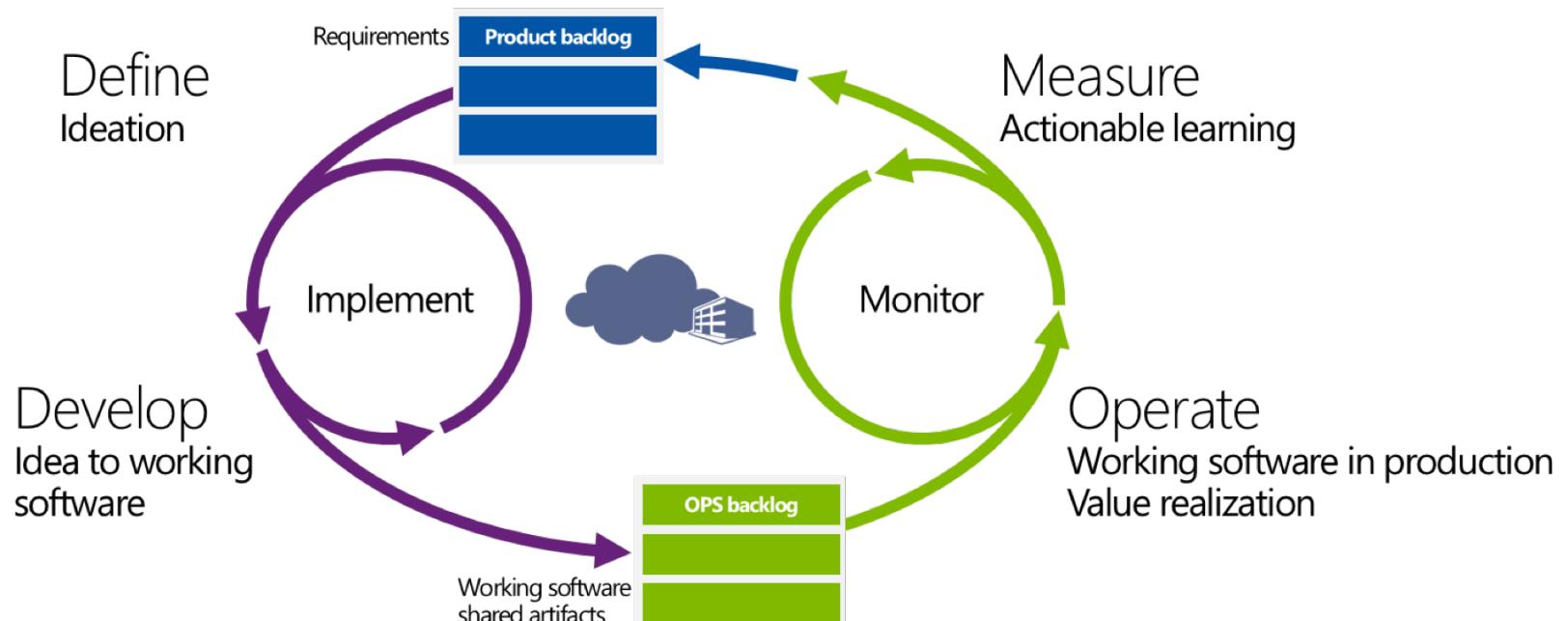


*Tools*



# Application Lifecycle Management

Waste elimination | Cycle time reduction | Integration & visibility



Continuous feedback | Continuous quality | Continuous delivery

Source: <http://incyclesoftware.com/wp-content/uploads/2013/01/modern-release-cycle.png>



# Evolving customer and market expectations

User Experience	Today	Emerging
Delivery Model	Planned	Incremental (DevOps)
Development and Operations Team Sizes	100s and Costly	10s with built-in DevOps automation
Release Frequency	Months to Years	Days to Weeks, based on business opportunity
Integration Frequency	Weeks	Continuous
Infrastructure Deployment	Days	Minutes
Time to Value	Planned	Opportunistic
Operational Model	Systems Management	Built into application, Recovery Oriented Computing, Continuous Availability



# A DevOps Manifesto

We are uncovering better ways of running systems by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes **over tools**

**Reliable and repeatable processes are critical to OPS**

Working **systems** over comprehensive documentation

**Working systems, not just working software**

Customer **and developer** collaboration over contract negotiation

**Customers, developers, and OPS must be partners**

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

# How do we make this happen?

**Automate** everything

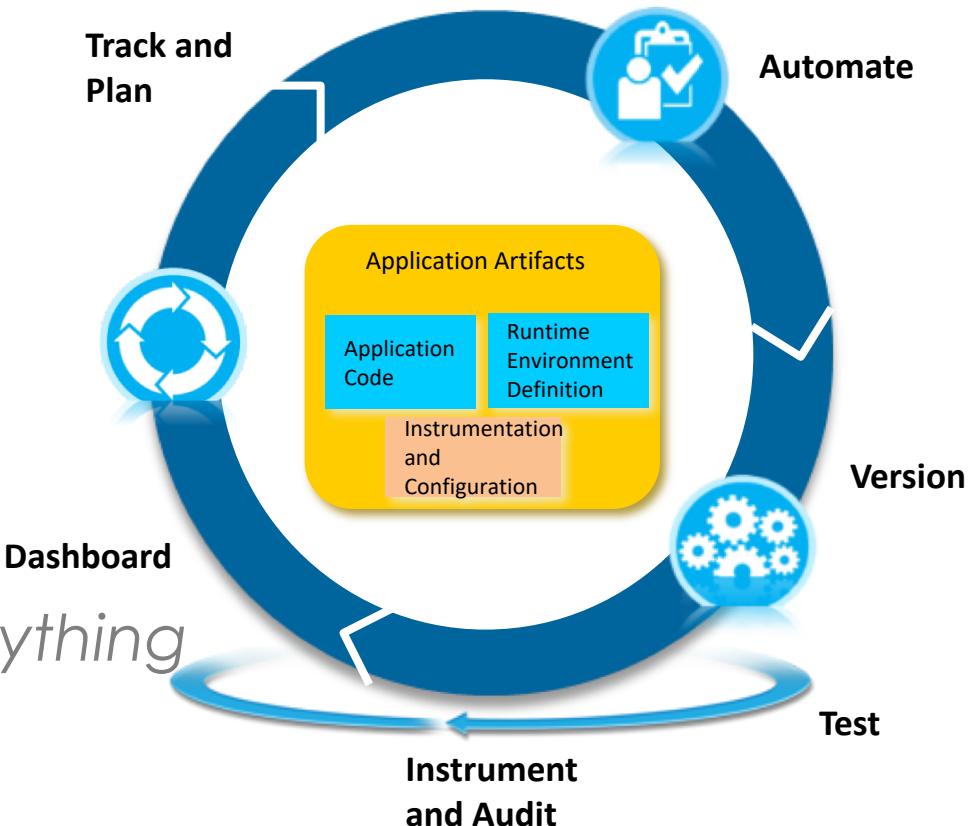
**Version** everything

**Test** everything

**Track and Plan** everything

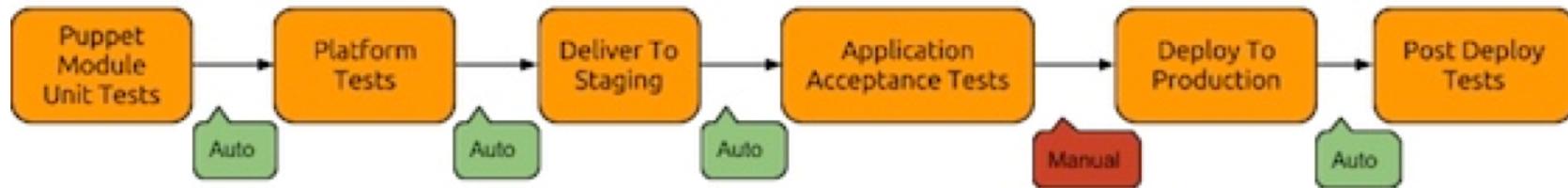
**Instrument and Audit** everything

**Dashboard** everything

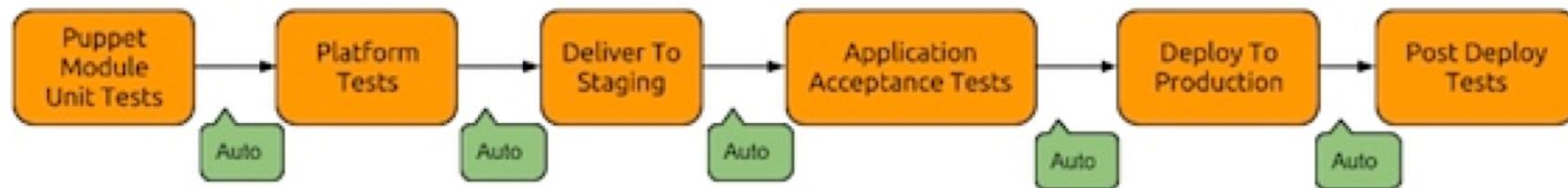


# Continuous Delivery vs Deployment

## Continuous Delivery

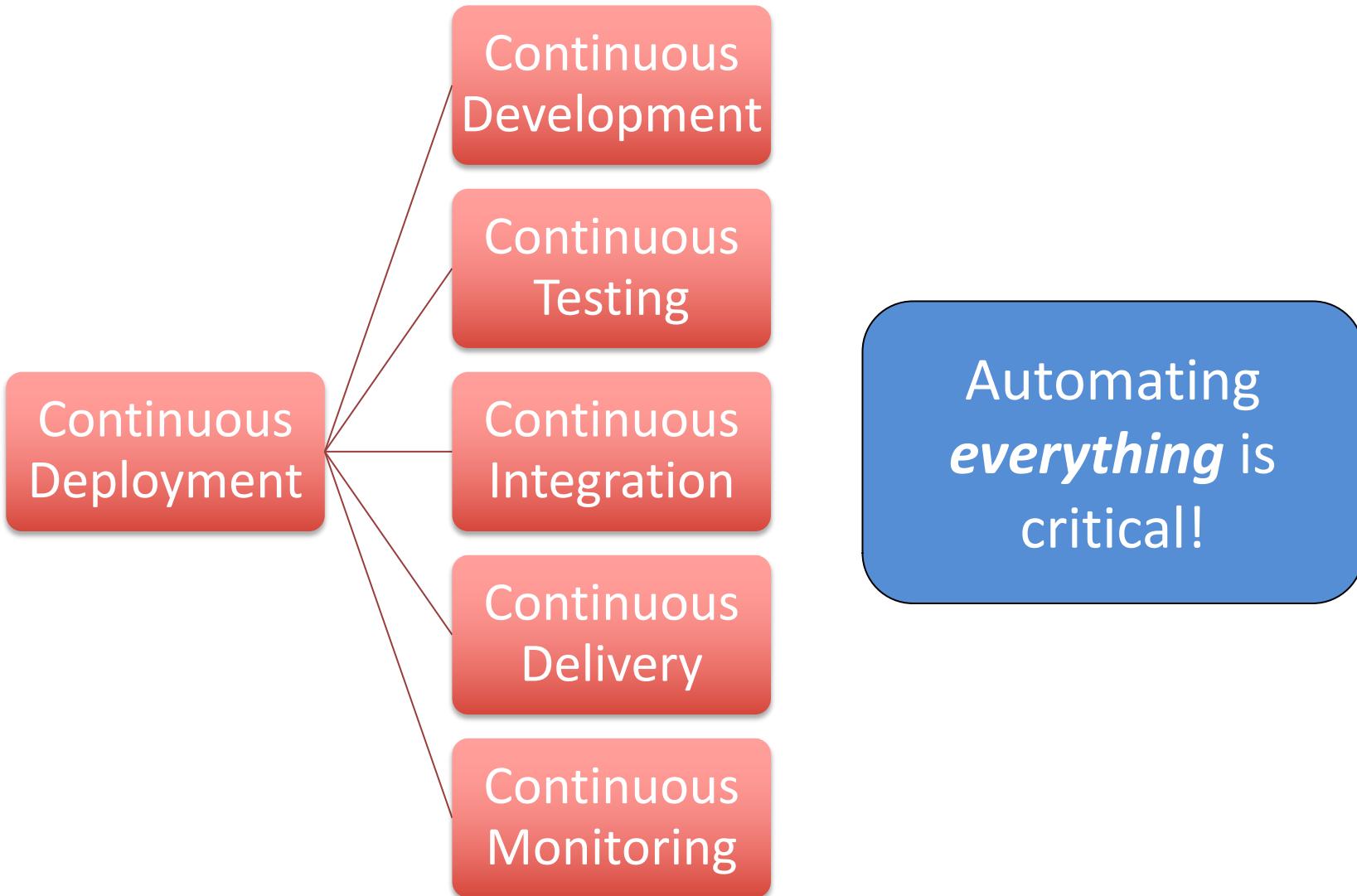


## Continuous Deployment



Source: <http://dev.housetrip.com/2014/03/12/continuous-deployment/>

# Continuous Deployment Requirements





# Continuous Delivery Principles

Create a **repeatable** and **reliable** process for software releases

Releases occur frequently so the right process is critical

**Automate** as much of the process as possible

Manual processes are slow and error prone

Identify and automate the hardest parts

Keep **everything** needed to recreate the build under version control

You may need to recreate any previous build





# Continuous Delivery Principles

## Build quality in (Lean)

Include quality improvements up front, don't just measure after the fact  
Automated testing of all steps is critical

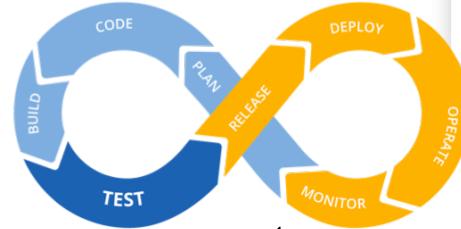
## You're **not “done” until the feature is released**

Everyone **shares responsibility** for delivery, not just Ops

## Reflection and continuous improvement are critical



# DevOps Tools and Practices



## Source Control/Configuration Management

Keep everything needed to deliver a release under source control

## Build tools

Developers' build environment should reflect deployment  
Completely automate the build process including all steps  
Build, stage, and test every time a change is committed to source control

## Automated test

All testing must be automated, e.g. TDD  
Unit, integration, system, performance, security, acceptance

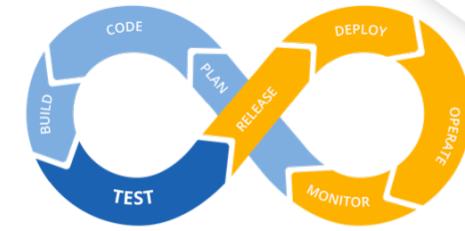
# DevOps Tools and Practices

## Continuous Integration

Integrate and test after every commit

Identify problems quickly

Develop, integrate, and test against a production-like environment



# DevOps Tools and Practices

## Monitoring

Critical to monitor everything relevant

Code quality, complexity, coverage

Mean time to failure

Mean time between failures

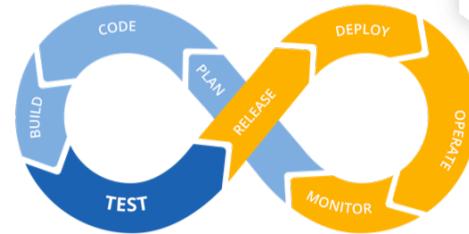
Mean time to resolution

Server capacity: disk, memory, network

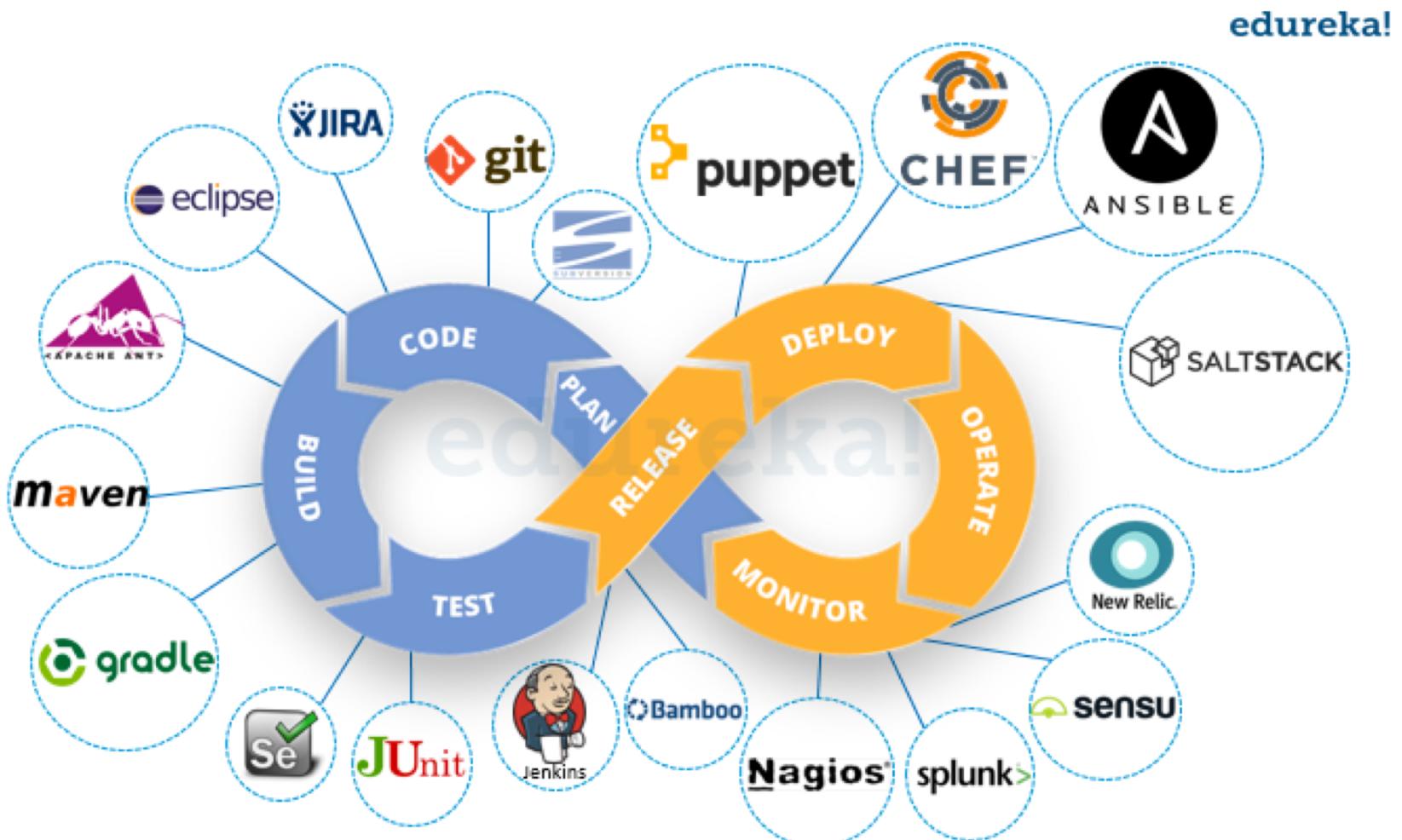
User behavior

Response times

Real-time monitoring



# Tools for continuous deployment



<https://blog.xebialabs.com/wp-content/uploads/2016/03/DevOps-cycle-PPT-COLOURS.png>



# DevOps summary

DevOps extends Agile principles from code to include systems and operations

DevOps is a philosophy

Continuing evolution between developers and operations

# Questions?

