STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

SSW-555: Agile Methods for Software Development

DevOps

Prof. Jim Rowland
Software Engineering
School of Systems and Enterprises

Today's topics

What problem are we trying to solve?

Challenges to success

DevOps solution

DevOps Manifesto

What problem are we trying to solve?

Help our business to succeed
- Build new features quickly
- Optimize 24x7x365 operations

Provide a great ***customer*** experience
- Deliver innovative products quickly
- Respond quickly to customer issues

Provide a great ***employee*** experience
- Attract and retain the best people

Identify and use optimal processes and tools

Sample App: Point Of Sales Solution

You've joined ***Round****, a fictional start up company that offers a Point of Sale e-commerce solution for small and medium businesses

Major Features:

- Accept payment for purchases from many physical and online vendors
- Online customer reports, e.g. purchases
- Online vendor reports, e.g. inventory, sales summary and details
- 24x7x365 availability and customer support

Which processes, procedures, tools, and technologies to succeed?

* Round is inspired by Square, Inc. https://squareup.com

Alternative approaches

Traditional Approach

Infrequent releases
- Monthly, quarterly, annually

Users impatiently awaiting features

Unexpected bugs found
- Unhappy customer
- Emergency release with insufficient testing

Even unhappier customer

Unhappy product team

DevOps Approach

Frequent releases
- Daily, hourly, continuous

Users trying new features

Users providing feedback
- New features
- Ideas for brand new features

Happy customers

Happy product team

What is DevOps?

"DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support."

- theagileadmin.com

How do we get started?

How to organize and manage the company?

How do we get the service off the ground and running?

- Who specifies the required features?
- Who prioritizes the work?
- What is the architecture?

How do we build the code?

What skills do we need to operate the service?

- Where is the service hosted?
- How and when do we deploy new services?

What do we do when things go wrong?

Who's Involved?

Managers

Customers

Operations/Ops

Developers/Dev

Roles and Responsibilities

Customers	• Use service • Provide feedback
Managers	• Strategic direction • Resource management
Dev	• Develop/test front end features • Develop/test back end features
Ops	• Front-end customer support • Back-end product support

Dev vs. Ops

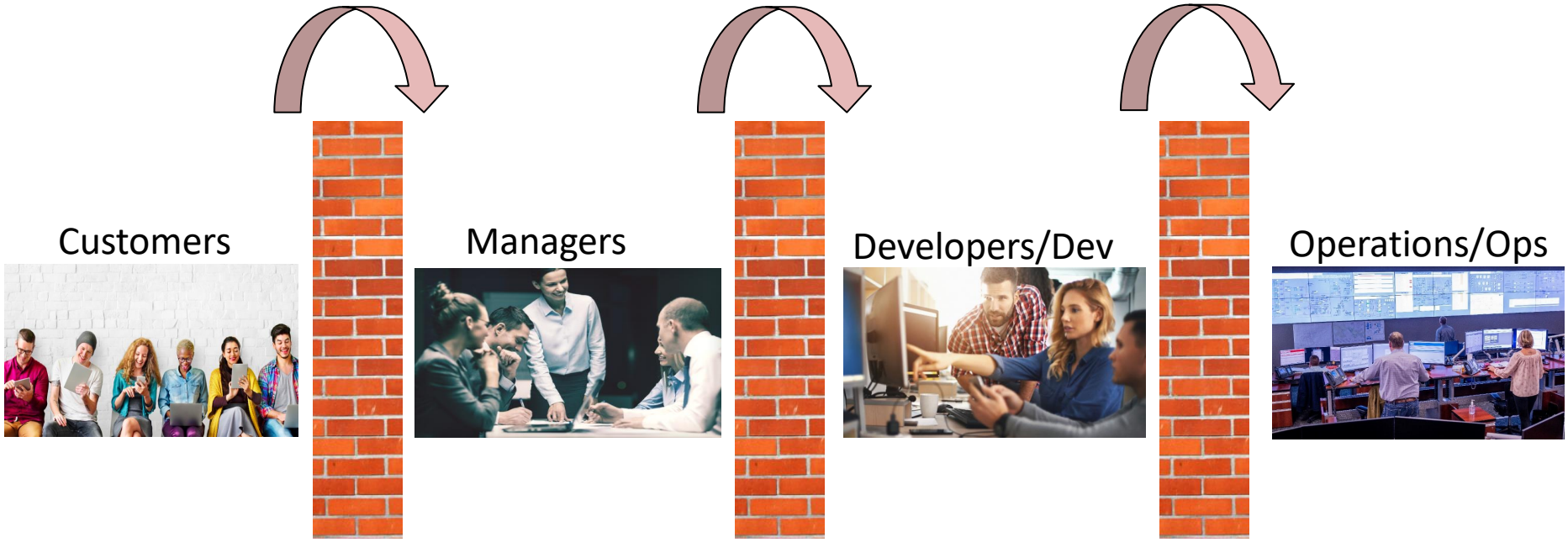
Development

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

Ops

Source: http://www.matrikasoftware.com/implementation_deployment.html

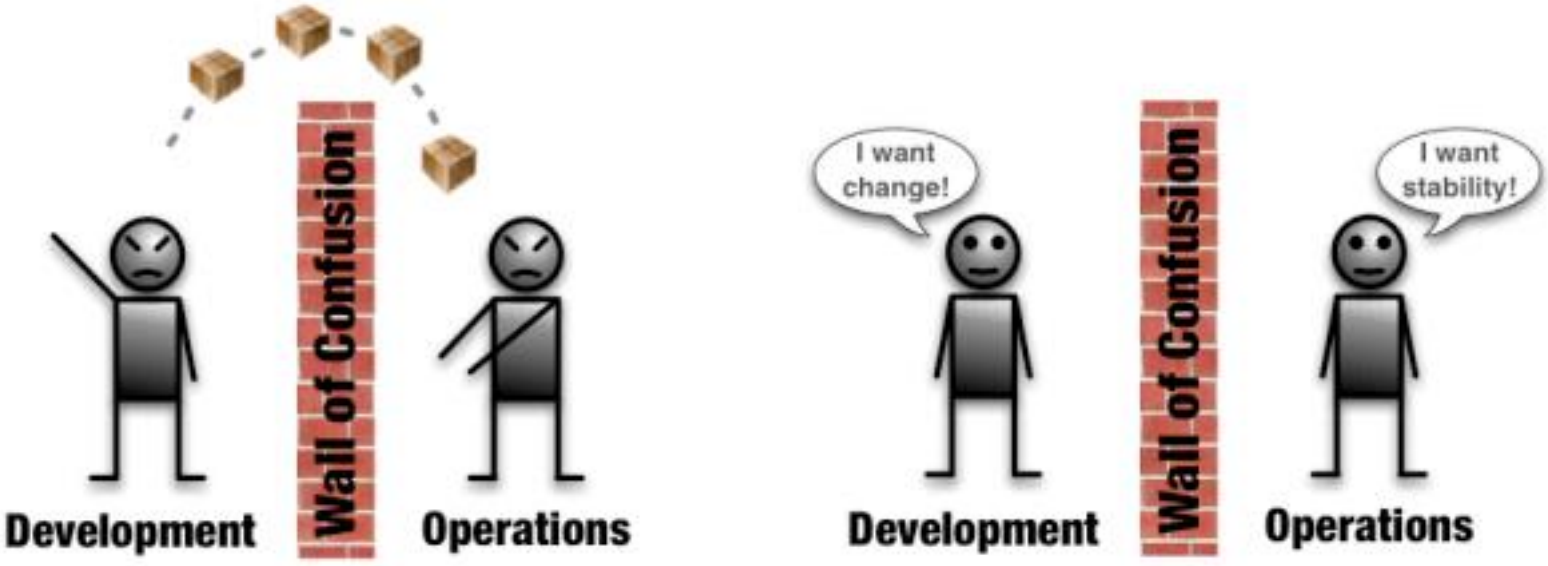
Traditional Organizational Structure



Challenges: Dev vs. Ops

Developers and operations have conflicting goals

- Dev tries to move quickly to deliver new features
- Ops strives to provide a stable environment for customers



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

Challenges: Managers vs Dev & Ops

Managers push Dev to deliver new features

- Dev takes shortcuts to meet unrealistic deadlines
- Ops reacts to problems caused by shortcuts

Challenges: In the end…

The situation deteriorates:

- Customers report more problems requiring more changes
- Process requires more communication and more process
- Everything slows to a crawl with lower productivity, frustrated customers, disgruntled employees

Imagine a (DevOps) world…

All parts of the company work together to focus on the customer's experience

Small, cross functional teams experiment to deploy new, high quality features quickly

Teams focus on the entire value stream rather than on their individual goals

A better way with DevOps …

Cross-functional teams work together to optimize customer experience

New features released frequently with higher quality

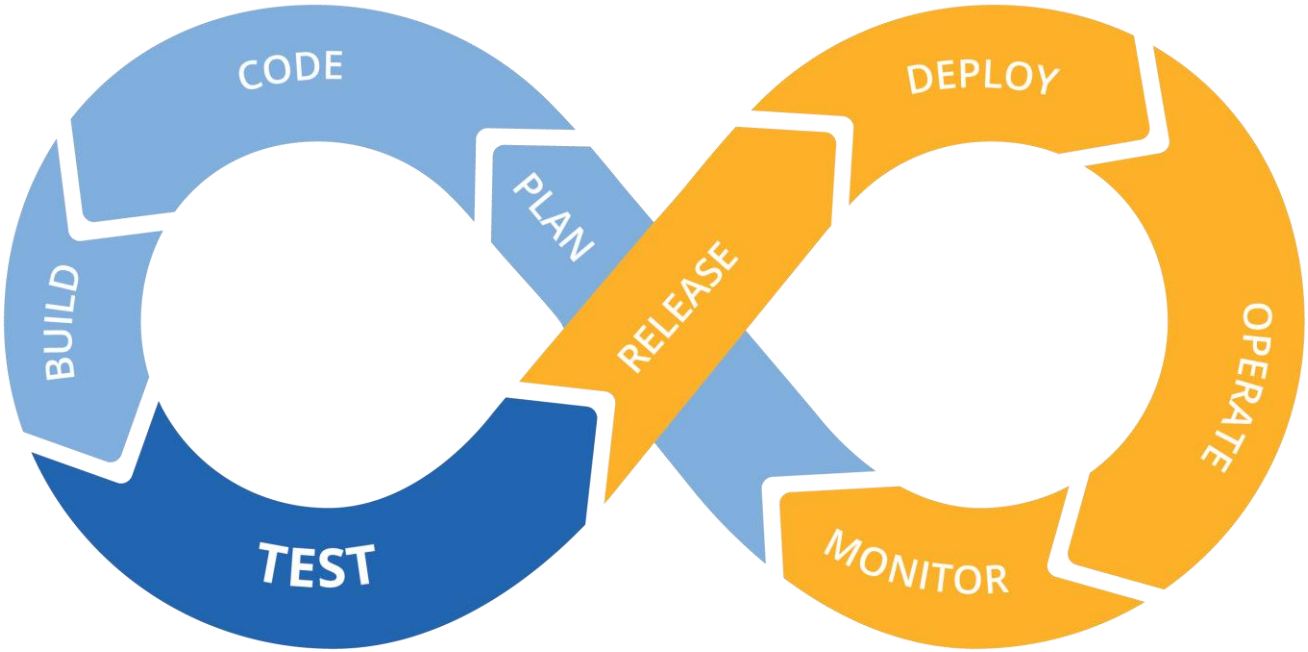
Higher job satisfaction for team

Focus on end to end flow

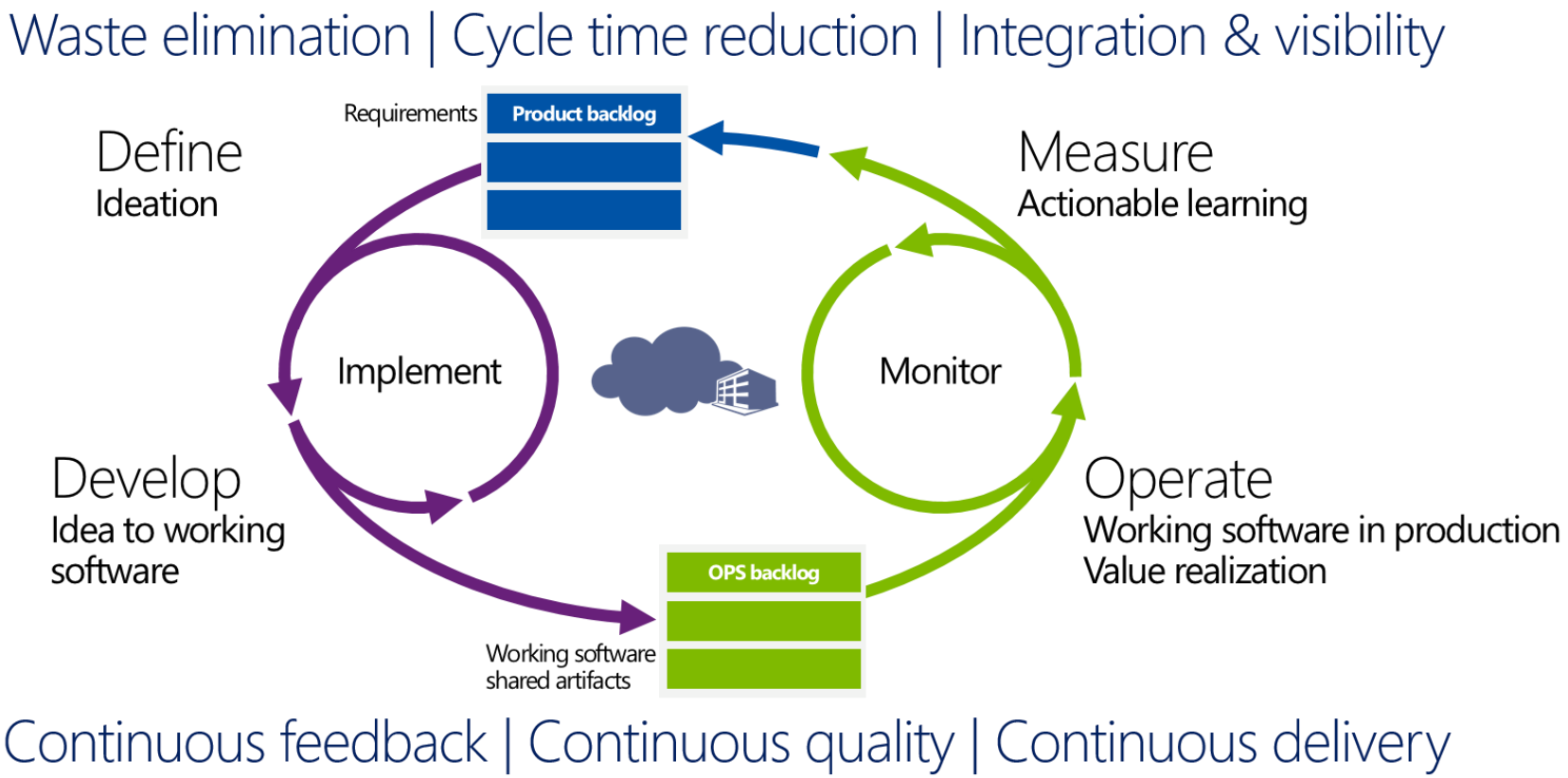
- Not on whose fault

DevOps Solution

Tear down the walls!
Integrate Development and Operations to deliver business value faster and better



Application Lifecycle Management



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

Achieving DevOps: CALMS

Culture
- Empowered teams
- Shared responsibilities

Automation
- Automate everything

Lean
- Continuous improvement
- Eliminate waste

Monitoring
- Measure and report everything

Sharing
- Share learnings across the organization

DevOps Solutions: Culture

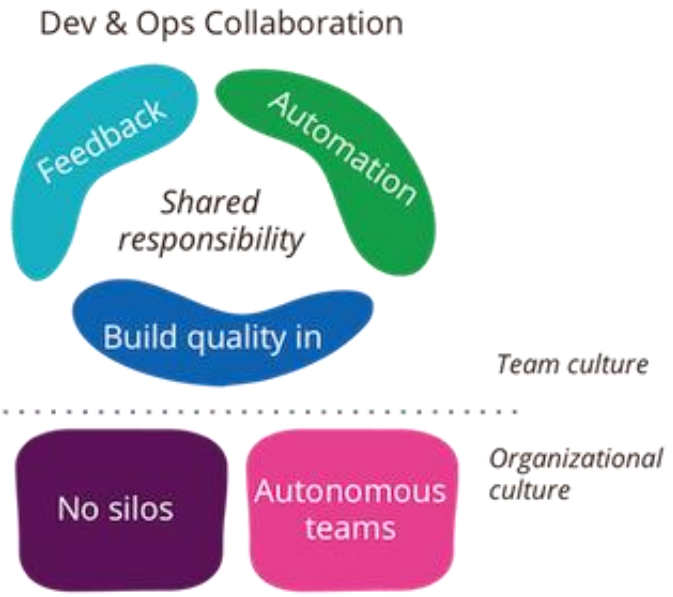
Hypothesis driven culture
- Take nothing for granted – measure everything
- Treat development and process improvement as experiments

Value everyone's time
- Don't build features that customers don't want
- Don't deploy code that doesn't work

Replace culture of fear with collaborative culture where people are rewarded for taking risks

Peer reviews, blameless-post mortems

Encourage learning across the organization



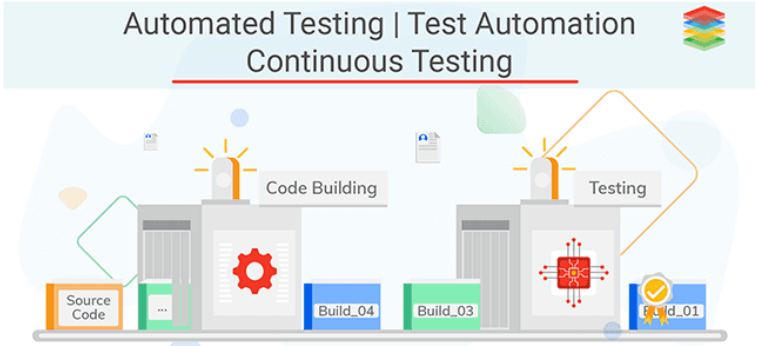
https://martinfowler.com/bliki/DevOpsCulture.html

DevOps Solutions: Development

All new features include automated tests

All code is integrated, and automated tests are run every time new code is checked in

- Bugs are discovered when the feature is built rather than waiting until later to integrate and test
- Bugs are easier to find and fix right after development



https://www.xenonstack.com/blog/automated-testing-devops/

DevOps Solutions: Deployment

Small, independent, decoupled teams develop, validate, and deploy new features quickly and safely

New features are created and validated in a production-like environment to minimize risk

Code deployments occur frequently, during normal business hours

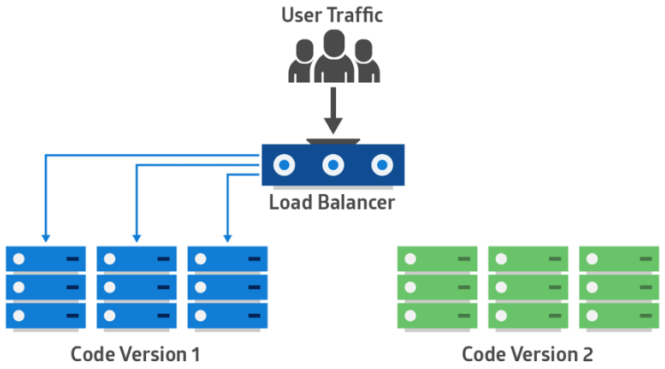
- Frequent deployments make deploying common place
- Traditional deployments happen infrequently and are scheduled for off-hours

DevOps Solutions: Deployment

Dark launches

- Deploy only to a small subset of users until the new feature is validated
- Features can be enabled and disabled easily through configuration, e.g. Spotify feature toggles

Roll back new features if a problem is detected



https://dev.to/mostlyjason/intro-to-deployment-strategies-blue-green-canary-and-more-3a3

DevOps Solutions: Sharing

Fast feedback loops

Entire development/deployment pipeline includes telemetry logs to track changes, performance, and usage

Blameless post-mortems

Continually share learnings across the organization to improve and avoid previous mistakes

Blogs, internal conferences,

DevOps Benefits

Puppet Labs' State of DevOps Report (2013-2016)

30x improvement in code and change deployments

200x faster code and change deployment lead time

60x better success rate for production deployments

168x faster time to restore service

50% higher market cap

Higher employee satisfaction

50% less time spent fixing security issues

Who's using DevOps?

Many others…

DevOps sounds great! How/where do we start?

DevOps requires both Dev & Ops

Development

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

Ops

Source: http://www.matrikasoftware.com/implementation_deployment.html

Consider some alternatives…

Software Development Life Cycle (SDLC) choices

Organizational choices

How will these choices impact our company's success?

How to deliver features more quickly?

Minimize the barriers between conceptualization and deployment

- Everyone must work together:
  Customer, developers, testers, operations

Minimize waste in the process

Focus on customer needs

Automate ***everything***

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*

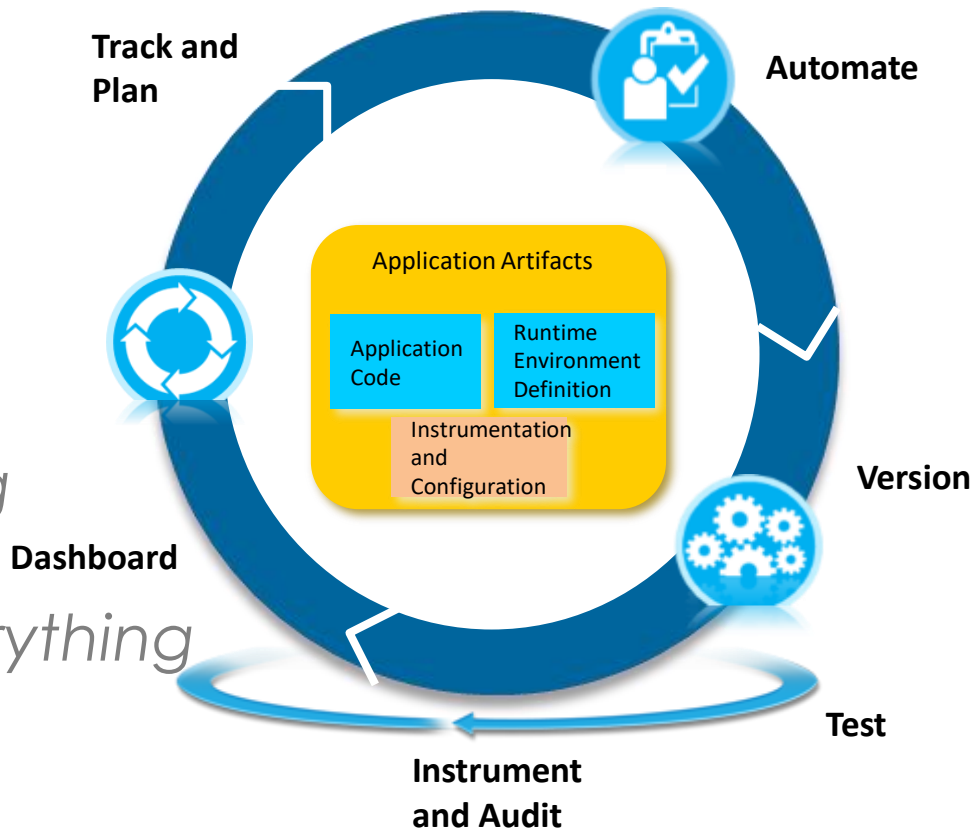
Version *everything*

Test *everything*

Track and Plan *everything*

Instrument and Audit *everything*

Dashboard *everything*



Questions?