DevOps

Continuously Improving Culture

Alicia M. Sepanik
Joel J. Parks
Robert T. Bauer

Background

DevOps is a hot topic in IT. Developers see it as a way for them to "push" their code faster; operations thinks it's a way to make the code better; and management isn't quite sure what DevOps means.

In this talk, we define DevOps, identify/discuss some challenges that face (will face) organizations adopting DevOps, overview adopting & sustaining DevOps and cover the top 12 practices within DevOps.

Material for this presentation was gathered from the 2013 & 2014 DevOps Survey and various software engineering sources.

Outline

- Define DevOps
- Identify and discuss challenges
- Introducing, adopting & sustaining DevOps
- Top 12 practices

What is DevOps?

(what you think it is)

Responses to “What is DevOps?”

- It’s a software development process that brings developers and IT operations closer together
- It’s a cultural movement where developers and operations work closer together to create business agility
- It’s another term for agile development
- It’s automating the production environment
- It’s another industry buzzword
- It’s a way for both sides to avoid being accountable

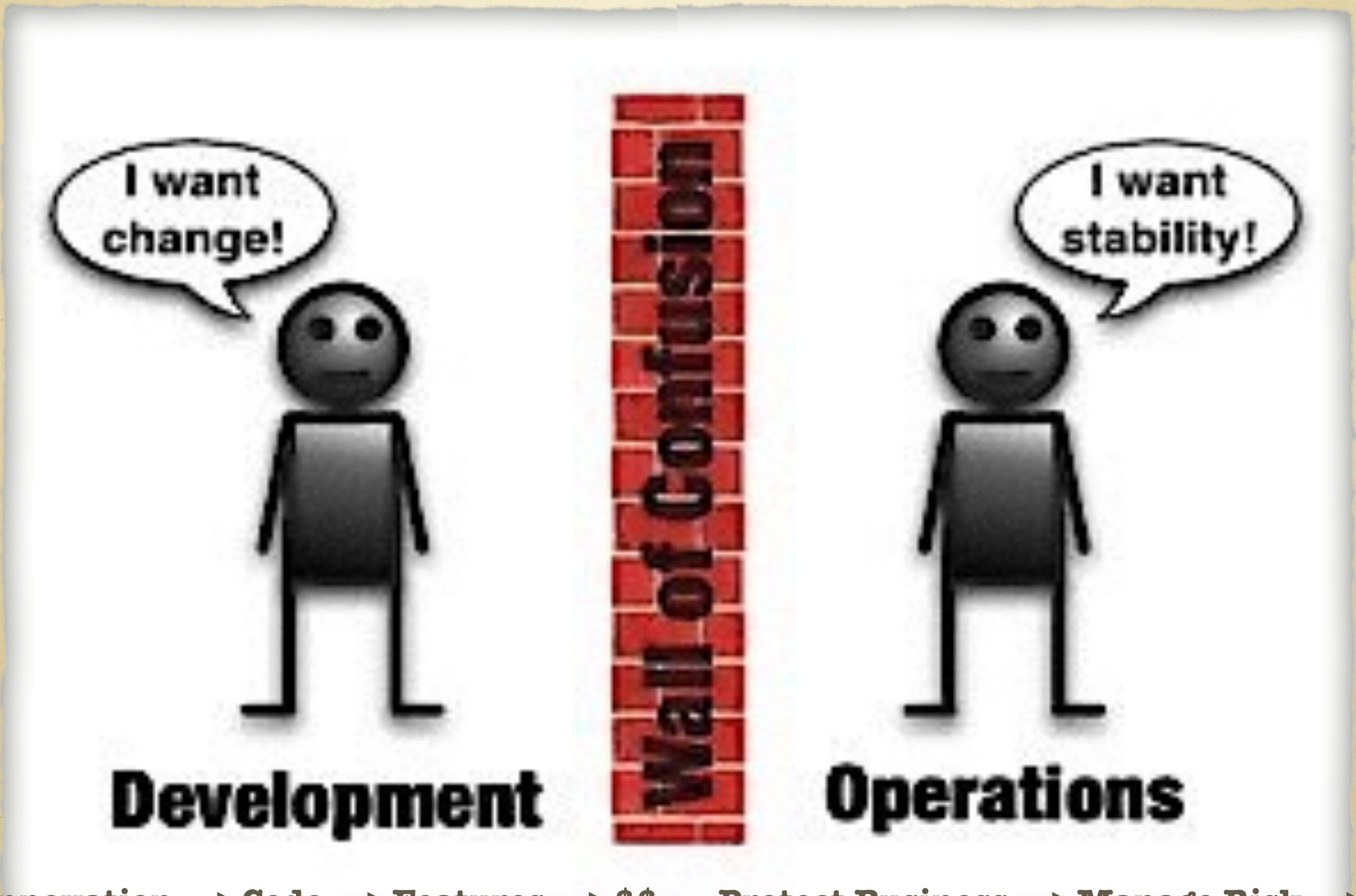
DevOps Is

- Set of principles to drive greater collaboration between different groups responsible for taking product or service to market; building on strong principles of automation and agile development to create a more end-to-end view of shipping value to customers
- Many (most?) IT organizations build services (not products); this drives short cycles with continuous incremental change; this drives continuous delivery; this drives customer demand which drives cultural changes from slow (55 on 65N) to fast (taking curves at 92 on the Autobahn)
- DevOps is making the heavily trafficked road from Dev to Prod safe at high-speeds

DevOps Can

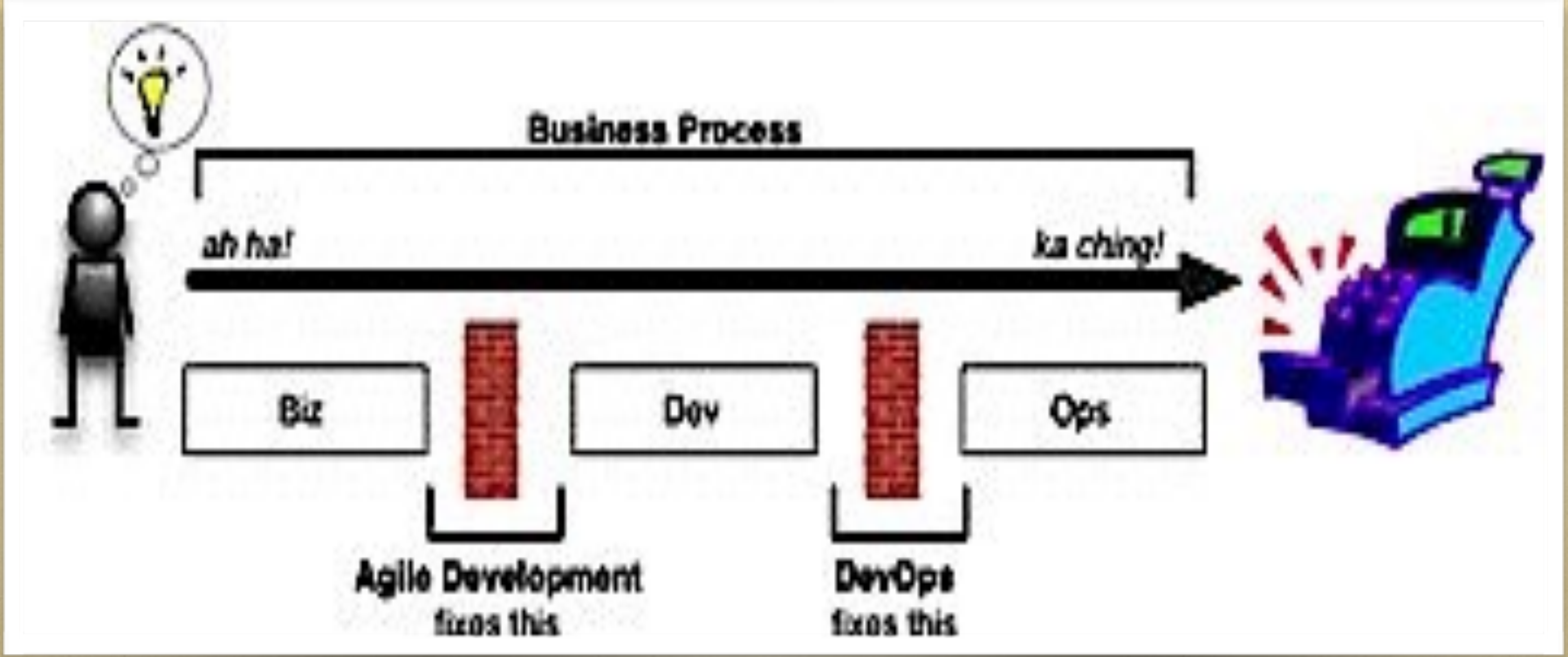
- Improve productivity by taking on deployment and configuration tasks that developers usually handle
- Reduce cycle time between code releases by making deployments faster
- Minimize infrastructure costs by scaling infrastructure with workload
- Improve infrastructure agility by implementing IaaS & PaaS and by automating configuration changes
- Improve quality and reliability by automating testing, implementing continuous integration and improving application monitoring

What About Culture?



Innovation ==> Code ==> Features ==> $$ **Protect Business ==> Manage Risk ==> Fixable**

DevOps Cultural Values



- The success of both individuals and groups needs to be measured within the context of the success of the entire development-to-production life-cycle
- The important theme of DevOps is that the entire development-to-production life-cycle must be viewed as one end-to-end process
- Infrastructure as code, model driven automation and continuous deployment are all concepts that fall under the DevOps banner

DevOps Principles

- Collaborative learning
- Agile methods and automation accelerate innovation
- Rapid feedback loops
- Whole system shapes the goal

DevOps is Continuous Change

- What types of people are comfortable in a structured environment that is always changing?
- What sorts of processes lend themselves to continuous change?
- Technology and technology adoption isn't changing as fast as we think but disruptions are like tidal waves
- What behavior does a culture of change drive? How to regulate without stifling innovation?

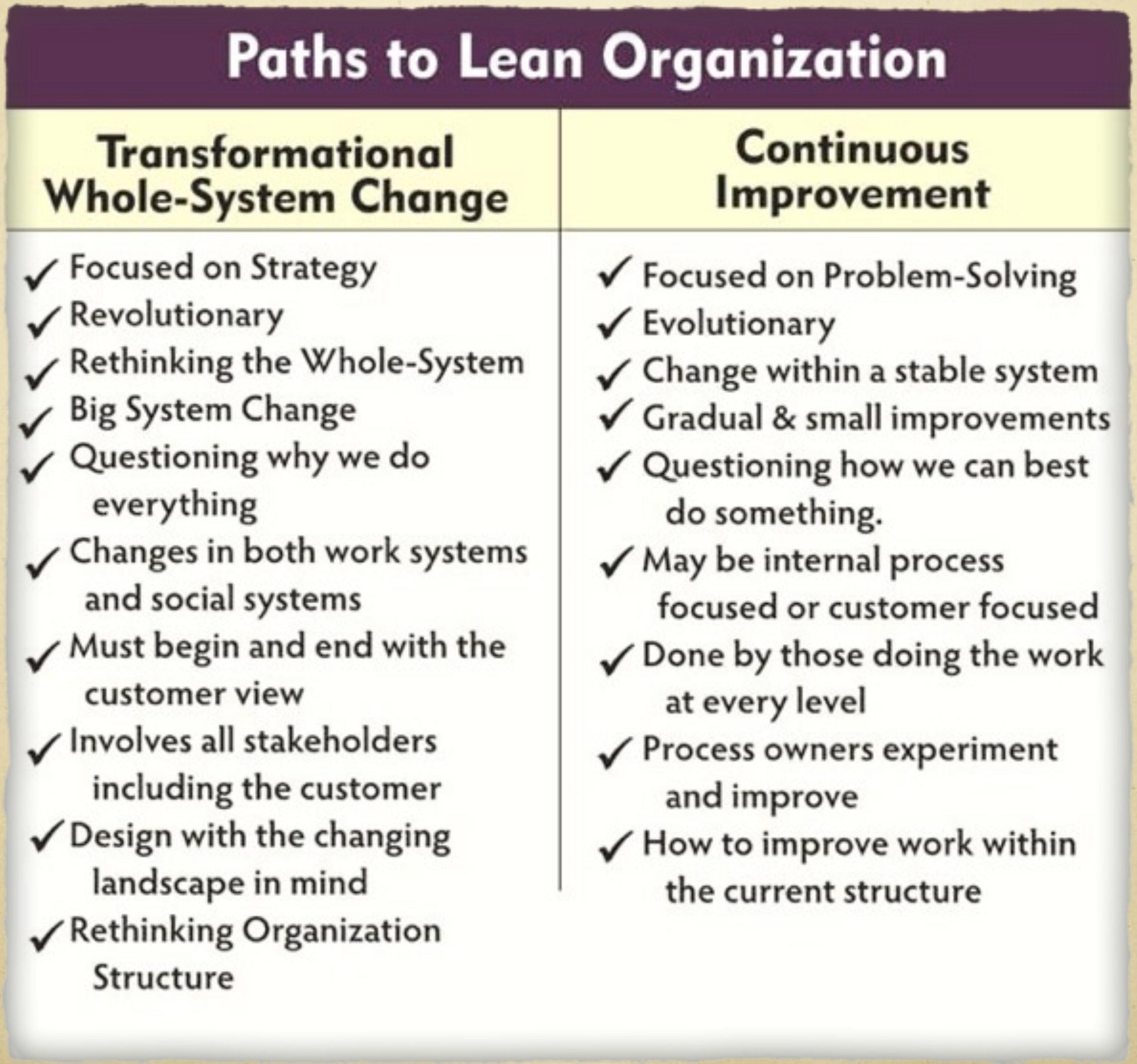
Introducing DevOps

(where are you)
(what’s important to introduce)

Introducing DevOps

- Start with ‘WHY’
  - What business problem? What IT issue?
- Look at culture
  - Eliminate development and operations silos
  - Promote communication among everyone delivering applications
  - Emphasize constant process & product improvement
- Evangelize and lead DevOps effort
- Select first application
- Set goals
  - Not just faster deployment or greater stability
  - Increase productivity
  - Reduce Cost from Dev-to-Prod
  - Increase customer engagement

Adopting DevOps



Paths to Lean Organization

Transformational Whole-System Change	Continuous Improvement
✓ Focused on Strategy	✓ Focused on Problem-Solving
✓ Revolutionary	✓ Evolutionary
✓ Rethinking the Whole-System	✓ Change within a stable system
✓ Big System Change	✓ Gradual & small improvements
✓ Questioning why we do everything	✓ Questioning how we can best do something.
✓ Changes in both work systems and social systems	✓ May be internal process focused or customer focused
✓ Must begin and end with the customer view	✓ Done by those doing the work at every level
✓ Involves all stakeholders including the customer	✓ Process owners experiment and improve
✓ Design with the changing landscape in mind	✓ How to improve work within the current structure
✓ Rethinking Organization Structure	

Sustaining DevOps

(staying fast and going faster)

Sustaining DevOps

- Specify and design to capture existing knowledge & building in tests to reveal problems
  - expected outcomes, process verification and validation, methods used to accomplish each piece of work
- Test components in isolation & build in resilience
  - inject failures, inject failures during intermediate states
- Automate testing
  - Simulate users, mock out backends, inject faults
- Blameless post-mortem for every customer-impacting outage
  - Fix problem! Fix process! Build in more tests!
  - Written post-mortems — and socialize!!!!!
- Share solutions & process that lead to solution
  - Contain problems, propagate solution and process
- Managers help the teams become self-diagnosing, self-improving, skilled at detecting problems, solving them and multiplying the effect by making solutions available throughout the organization

Top 12 DevOps Practices

- Integrate development with operations
- Align DevOps goals with business goals
- Continuous integration
- Automated testing
- Application monitoring
- Production support
- Integrated (Dev & Ops) deployment planning
- Dev/Ops war room to 'test' DevOps process
- Continuous deployment
- Automated dashboard
- Infrastructure automation
- Log aggregation

Expectations & Comparisons

- 11.6 seconds MTBD (Mean-Time Between Deploy)
- 0.001% deployments cause an outage (airlines beat this)
- 10,000 deployments/hour (maximum)
- Value of DevOps not known outside of DevOps
  - Difficult to get managers to support
- DevOps Skill Set
  - Coding, Process Re-engineering, People, General Tools
  - Automation is key success factor
- Metrics
  - Test cycle time, deployment time, prod defect rate
- Common tool chain
- Disaster day

Take Away

- DevOps is really about culture
  - Focused on delighting customer
  - Making the road from dev-to-prod safe at high-speeds
- DevOps is developer centric
  - Automation means (a lot of) code
- DevOps is continuous learning, continuous improvement and continuous change
  - Blame teams for crappy code
  - Identify and fix, replace or eliminate crappy processes

Q&A