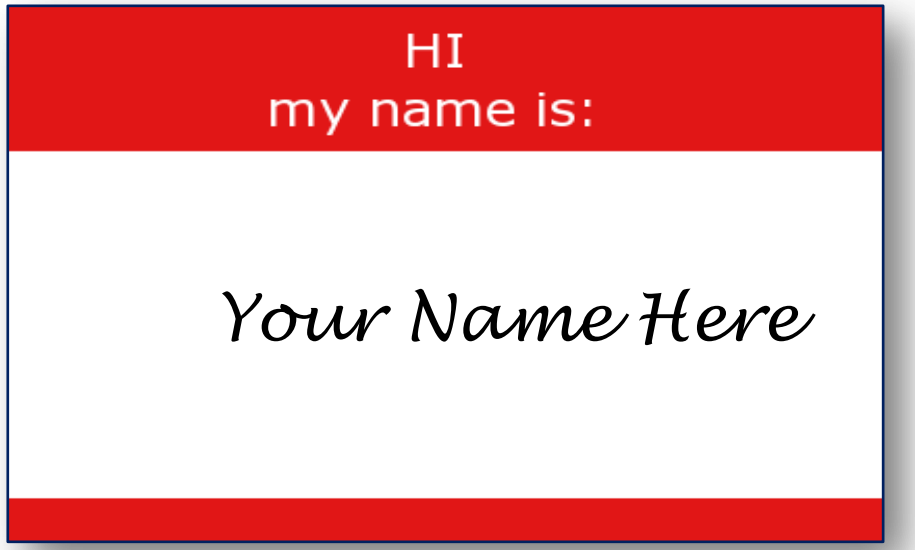
DEVOPS FOUNDATION

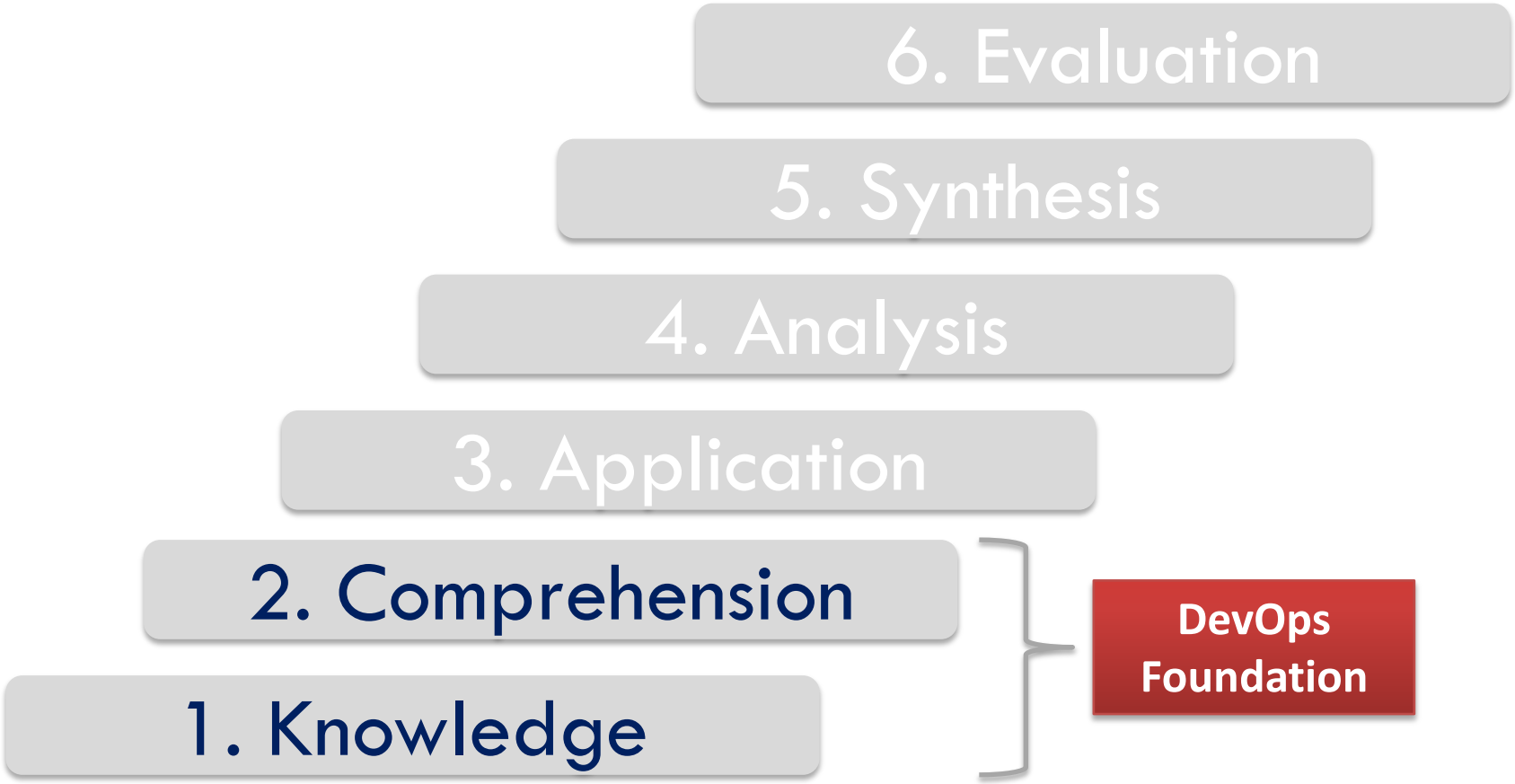
Tell Us a Little About Yourself

- Tell us a little about yourself
  - Name, organization and role
  - DevOps/Agile/Lean/ITSM experience
  - Why you are attending this course
  - What you expect to learn



What is your definition or perception of DevOps?

About Bloom's Taxonomy



Bloom's Taxonomy is used to categorize learning objectives and, from there, assess learning achievements.

DevOps Foundation Course Agenda

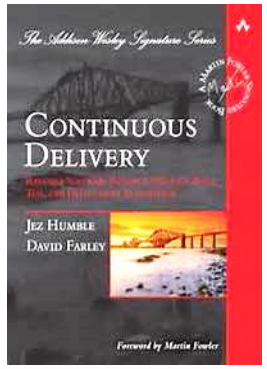
Day 1	Day 2
• Class and course Introductions	• Morning Review

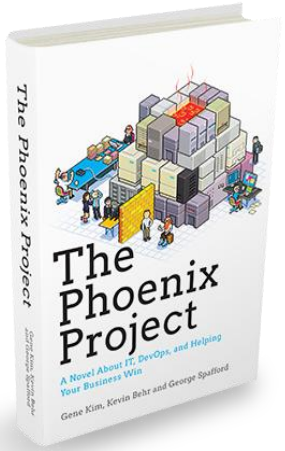
• What is DevOps? Why Now?
• The Business and IT Perspectives
• DevOps Principles
  • The Three Ways (Phoenix Project)
• DevOps Practices
  • Continuous Integration
  • Continuous Delivery and Deployment
  • Rugged DevOps/DevSecOps
  • ChatOps
  • Kanban
• DevOps and Other Frameworks
  • Agile
  • ITSM
  • Lean
	• DevOps Values
	  • Culture
	  • Automation
	  • Lean
	  • Measurement
	  • Sharing
	• DevOps In the Enterprise
	  • Roles
	  • Organizational Considerations
	• Getting Started
	  • Challenges Risks and Critical Success Factors
	• Exam Review

4

The DevOps Collective Body of Knowledge

- True to its core values, DevOps is emerging through a shared and collective body of knowledge (CBok) including
  - Publications
  - Conferences
  - Videos and webinars
  - Blogs and articles
  - Case studies
  - Subject matter expertise





The DevOps Community actively researches and influences emerging DevOps practices so as to create meaningful training and certification.

WHAT IS DEVOPS?

What is DevOps? (1)

While there are many interpretations of DevOps, the most commonly agreed definition is

A cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals while automating the process of software delivery and infrastructure changes.

It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably."

Wikipedia

Improving the ability of IT to produce software faster
improves the ability of the business to deliver value to customers faster.

What DevOps is NOT

- A title
- A separate team
- A tool
- Only culture
- Only automation
- Anarchy
- A one size fits all strategy

DevOps is coming to life through emerging practices that are delivering real value in real organizations.

http://devops.com/2016/03/17/what-devops-is-not/

DevOps Goals

- Smaller, more frequent releases
- Reduced effort and risks
- Reduced cost of product iterations and delays
- A culture of communication and collaboration
- Consistency and speed through automation

Improvements in

- Time to market
- Integration with the business
- Responsiveness
- Code and deployment quality
- Productivity
- Visibility
- Agility

NoOps or NewOps?

Does migration to the cloud signal the end of Ops? No!
The role of Ops is evolving much like the role of Dev.

While automating operations may seem to be the goal of NoOps, there will always be a need for human competencies and roles including

- Designing, implementing and managing internal and external infrastructures and enterprise architectures
- Providing customer support, managing alerts, triage and escalation
- Serving as an interface between cloud providers and customers
- Managing the operational automation

Many IT organizations are taking a hybrid approach towards local and cloud infrastructure based on governance, risk and compliance requirements.

DevOps Values

Automation

Lean

Measurement

Sharing

More than anything else, DevOps is a cultural movement based on human and technical interactions to improve relationships and results.

Automation is an Essential Element

Automation enables agility, consistency, speed and reliability.

and many more…

Shared decision-making, access to and an understanding of toolchains and other automation streamlines software delivery and prepares Ops for the long run.

DevOps Stakeholders

- Dev includes all the people involved in developing software products and services including
  - Architects, business representatives, customers, product managers, project managers, quality assurance (QA) testers and analysts, suppliers, etc.

- Ops includes all the people involved in delivering and managing software products and services including
  - Information security professionals, systems engineers, system administrators, IT operations engineers, release engineers, database administrators (DBAs), network engineers, support professionals, suppliers, etc.

DevOps extends beyond software developers and IT operations.

"You never change things by fighting the existing reality. To change something, build a new model which makes the existing model obsolete."

Buckminster Fuller

WHY DEVOPS? WHY NOW?

Our Cadence is Off

Historically…

The Business

Innovation

Dev

Waterfall Projects

Ops

Rigorous Processes

Cadence – the flow or rhythm of events.

DevOps Improves IT's Cadence and Velocity

…agile, lean and ITSM practices are also needed.

The Business

Winning through Innovation

Agile/Lean DevOps

Continuous Delivery

Why the Business is Driving DevOps

- Every business has become a tech business
- IoT is rapidly increasing
- Consumers have developed “app” mentalities
- Customers value outcomes, not products
- Time to value is replacing time to market
- Intelligent data must shape direction quickly
- Customer delight is more important than customer satisfaction

Your biggest competitor may be a start-up.

Why IT is Driving DevOps

- Every business has become a tech business
- IoT is rapidly increasing
- Consumers have developed “app” mentalities
- Customers value outcomes, not products
- Time to value is replacing time to market
- Intelligent data must shape direction quickly
- Customer delight is more important than customer satisfaction

Notice the similarities?

Who is Doing DevOps?

- **Web Pioneers**
  - Amazon
  - Netflix
  - Etsy
  - Facebook
- **Financial**
  - Bank of America
  - Barclays
  - Commonwealth Bank of Australia
  - ING Bank
- **Entertainment**
  - Disney
  - Sony Pictures
- **Academia**
  - Portland State
  - Seton Hill
  - Kansas State
- **Insurance**
  - Nationwide
  - Travelers
  - Hiscox
- **Manufacturing**
  - Apple
  - LEGO
  - Unilever
- **Retailers**
  - Target
  - Nordstrom
  - Sherwin Williams
  - REI
  - Macy's

and many more…

By 2016, DevOps will evolve from a niche to a mainstream strategy employed by 25 percent of Global 2000 organizations.

Technology that supports the DevOps toolchain is predicted to grow 21 percent worldwide.

Gartner, 2015

http://www.gartner.com/newsroom/id/2999017

Recent Findings

- According to the 2016 State of DevOps Report, high-performing organizations
  - Are outperforming in terms of throughput
  - Have better employee loyalty
  - Spend 50% less time remediating security issues
  - Understand that improving quality is everyone's job
  - See that taking an experimental approach to product development improves IT and organizational performance

Read the full report:
https://puppet.com/resources/white-paper/2016-state-of-devops-report

Undertaking a technology transformation initiative can produce sizeable cost savings for any organization.

DevOps is Increasing Agility and Stability

DevOps practices predict IT performance and IT performance predicts organizational performance.

- High-performing organizations are more agile
  - Code is deployed 200 times more frequently
  - Deployments are completed 2555 times faster
- Services are more stable
  - There are 3 times fewer deployment failures
  - Mean time to recover (MTTR) is 24 times faster

Source: *2016 State of DevOps Report*

"The (completely achievable) goal aligns IT goals with business goals by removing all of the bottlenecks, inefficiencies, and risks between a business idea (the 'ah-ha!') and a measurable customer outcome (the 'ka-ching!')."

Damon Edwards

THE BUSINESS PERSPECTIVE

The Outcome Economy

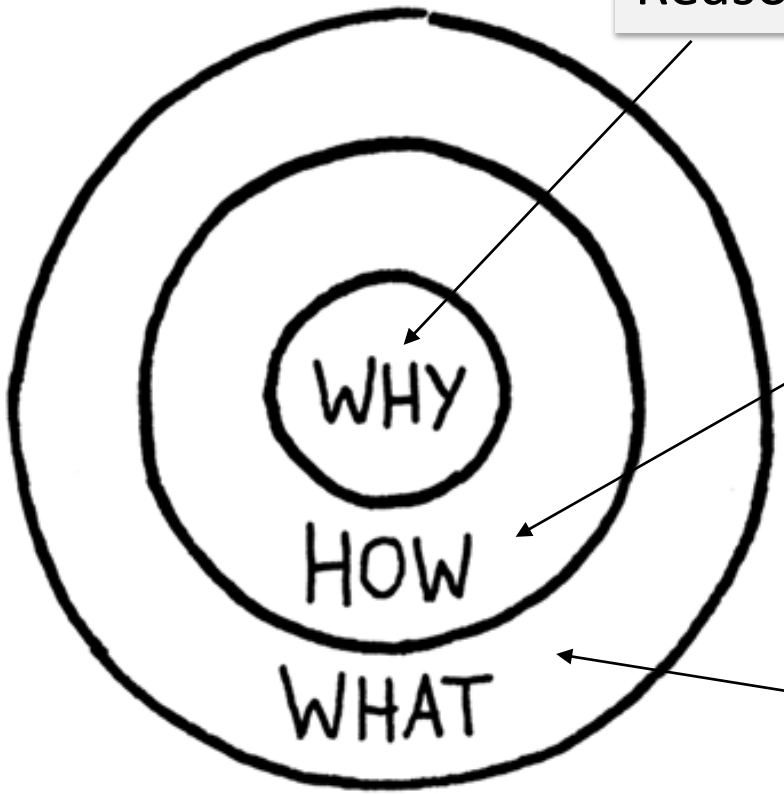
The outcome economy is a shift from competing by selling products and services to competing by selling measurable results important to the customer.

It's an economy where buyers are looking to buy an outcome and the sellers are selling a promise of an outcome.

Alan Alter, Accenture

http://searchcio.techtarget.com/feature/The-outcome-economy-is-upon-us-is-your-business-ready

Start with the “Why” - The Golden Circle



Purpose, cause, belief

Reason organization exists

What sets you apart

Products and services

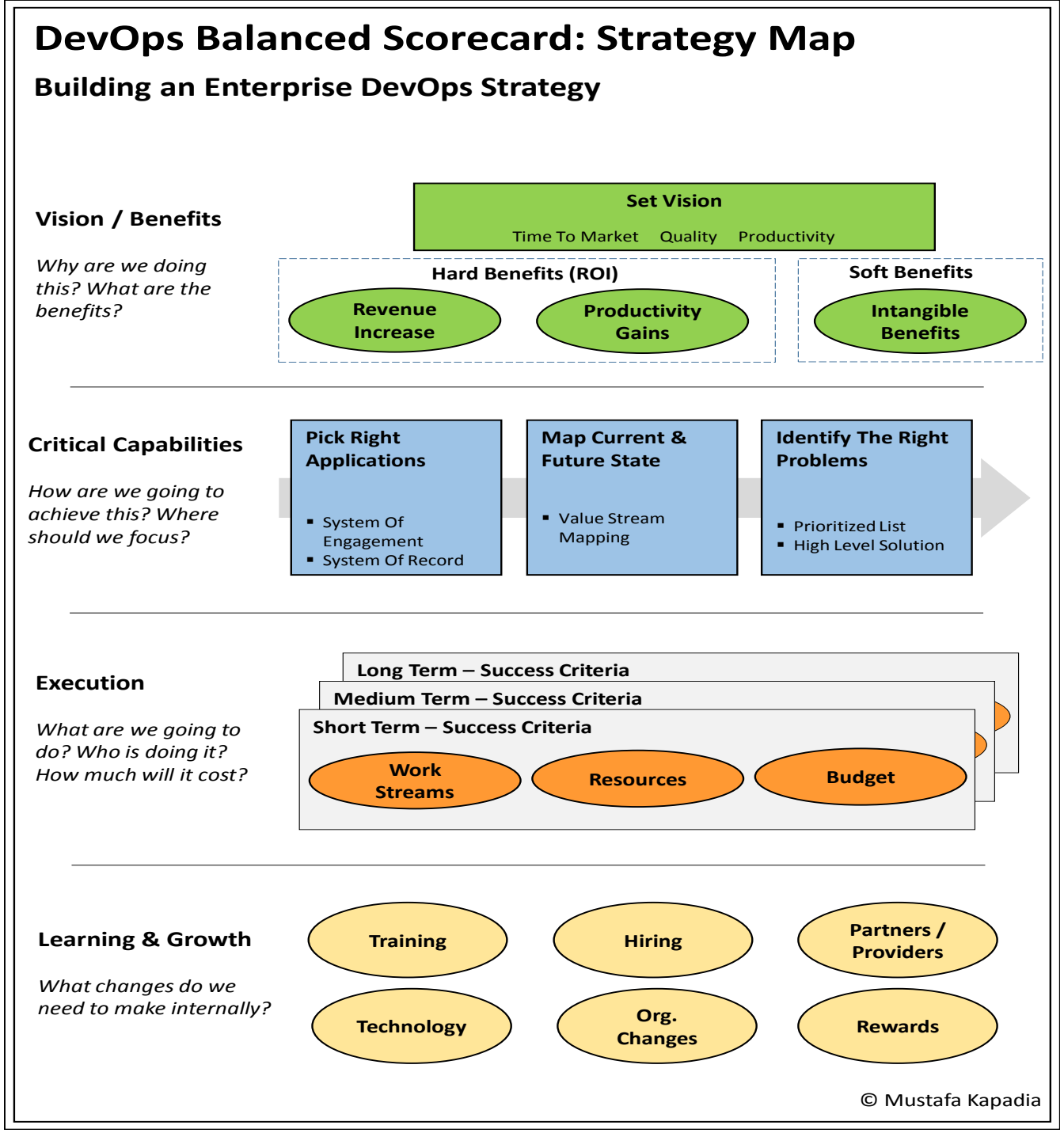
What is your organization’s “why”?

The WHY

The HOW

The WHAT

Supported by LEARNING & GROWTH



Source:
Mustafa Kapadia, IBM

The Business Value of DevOps

DevOps practices improve IT performance.

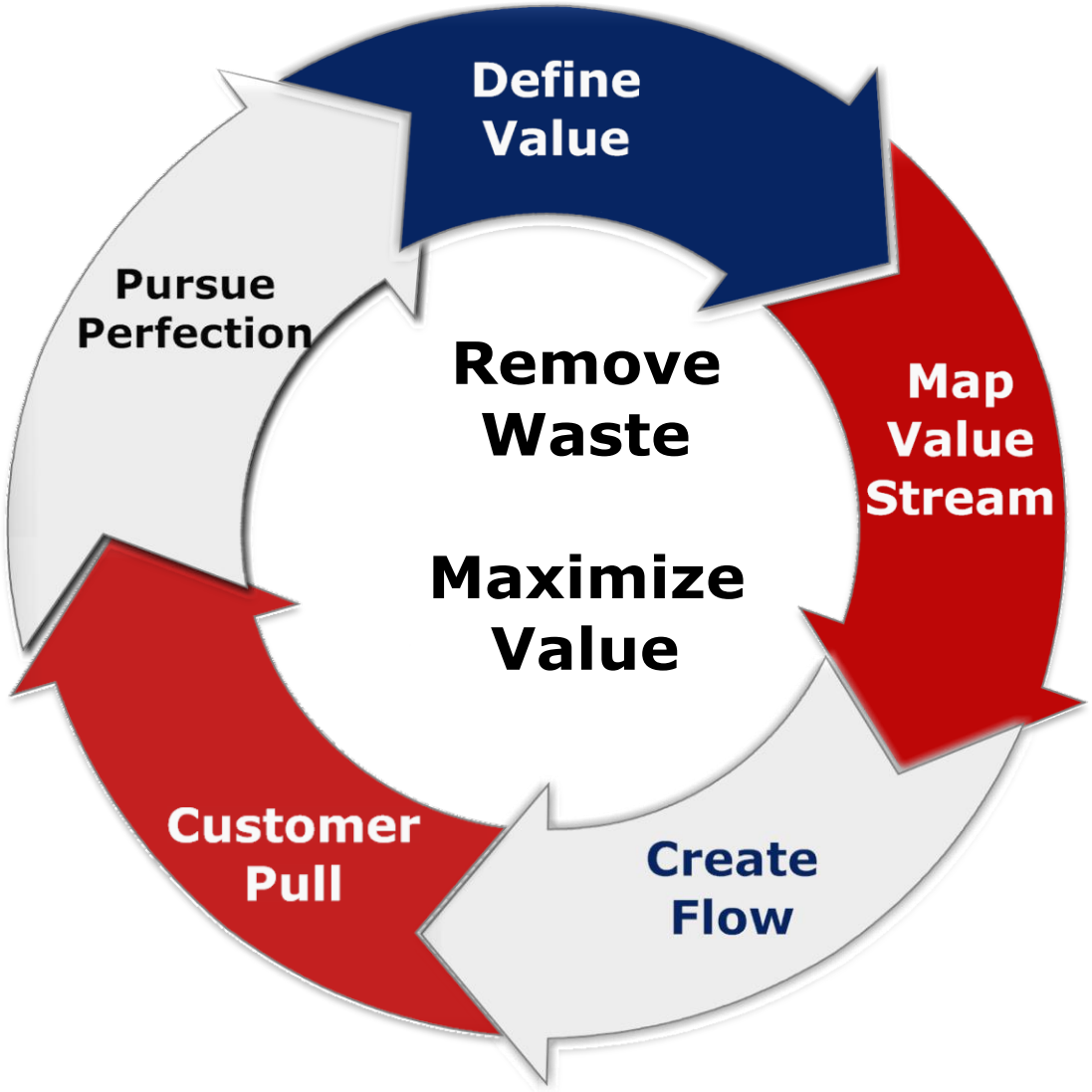
- Companies with high IT performance are twice as likely to exceed their goals
  - Profitability
  - Market share
  - Productivity

Strong IT performance is a competitive advantage.

Source: *2015 State of DevOps Report*

IT Must Contribute to the Business Value Stream

Value is defined by the customer.



"Agile was instrumental in Development regaining the trust in the business, but it unintentionally left IT Operations behind. DevOps is a way for the business to regain trust in the entire IT organization as a whole."

Clyde Logue
Founder of StreamStep

THE IT PERSPECTIVE

The IT Challenge

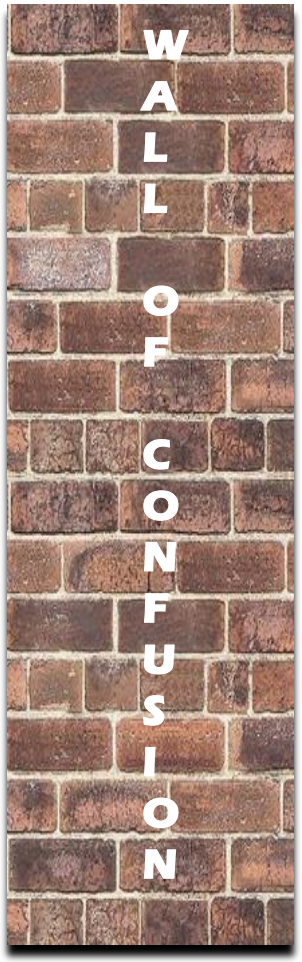
DevOps must continuously deliver outcomes by bridging and improving almost every aspect of IT.

- Internal IT challenges
  - IT must go faster, faster, faster without risking quality
  - Prior investments aren't delivering end to end value
    - Agile SW development is good but isn't delivering full value
    - ITSM processes are good but aren't delivering full value
    - New automation is good but isn't delivering full value
  - IT's silo culture is constraining the value stream

IT no longer needs to align or integrate with the business, IT <u>is</u> the business

The Wall of Confusion (1)

Dev wants change



Ops wants stability

What about Security, Governance, Risk Management and Compliance? What do they want?

The Wall of Confusion (2)

What does the business want?

All of the above

IT's Silo Culture

- **Security**
- **QA**

Development

Service Desk

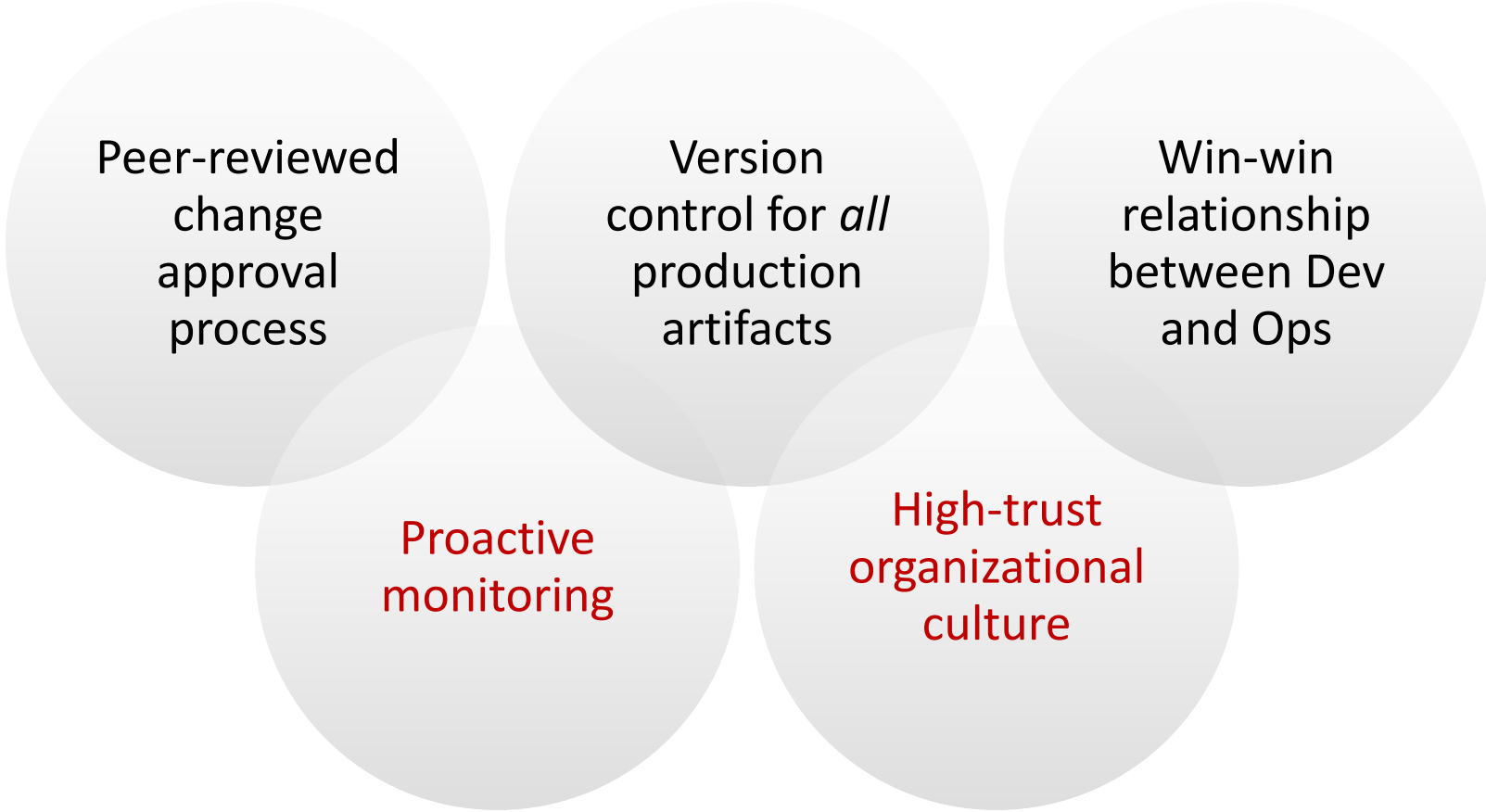
Operations

Isolated IT silos can foster stereotypes and misconceptions between Dev, Ops and other IT teams.

It can also affect the flow of work and IT's ability to deliver innovation continuously.

Top 5 Predictors of IT Performance

DevOps helps IT organizations become high-performing.



Source: *2014 State of DevOps Report* – Puppet Labs, IT Revolution Press and ThoughtWorks

Can IT be Bi-Modal?

Bi-modal IT is a controversial concept where two distinct IT methodologies exist in the same company, sometimes in two separate teams.

Mode 1 (Traditional IT)

- Focuses on predictability
- Optimized for what is known
- Renovates legacy environments to digital
- Develops long-term plans and goals
- Manages technology budgets
- Takes a disciplined approach to deployments

Mode 2 (Agile IT/DevOps)

- Focuses on innovation and exploration
- Optimized for uncertainty
- Experiments to solve new problems
- Tests and adapts hypotheses during short iterations (sprints)
- Adopts a minimum viable product (MVP) approach

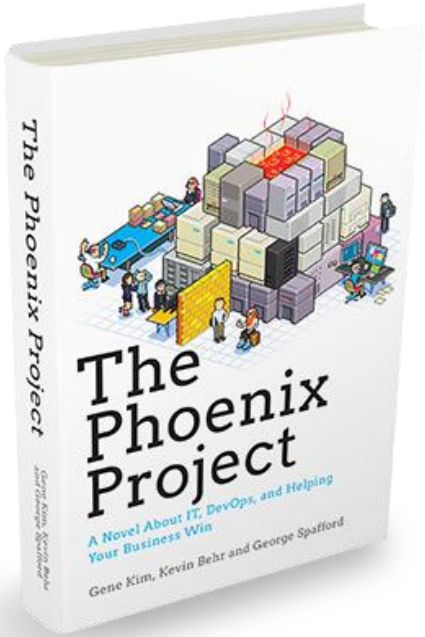
There is also a concept known as "variable speed IT" where traditional and digital processes co-exist and are allowed to operate at their own speed.

Isn’t This How It’s Supposed to Be?

Speed – Quality – Stability

DEVOPS PRINCIPLES

THE THREE WAYS



The Three Ways

- **The First Way** – Flow
  - Understand and increase the flow of work (left to right)
- **The Second Way** – Feedback
  - Create short feedback loops that enable continuous improvement (right to left)
- **The Third Way** – Continuous experimentation and learning
  - Create a culture that fosters
    - Experimentation, taking risks and learning from failure
    - Understanding that repetition and practice is the prerequisite to mastery

The First Way: Flow

- Understanding the flow of work
- Increasing flow by understanding and removing constraints
- Never passing a known defect downstream
- Never allowing local optimization to cause global degradation
- Achieving a profound understanding of the entire system

A goal of The First Way is to have work flow quickly from left to right.

Theory of Constraints

A methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.

The Theory of Constraints recognizes that

- Every process has at least one constraint or bottleneck that affects its ability to consistently meet its goal
- The process will only meet the capacity of its constraints and will be only as successful as its weakest link
- Improving constraints is the fastest and most efficient way to improve the entire process or system

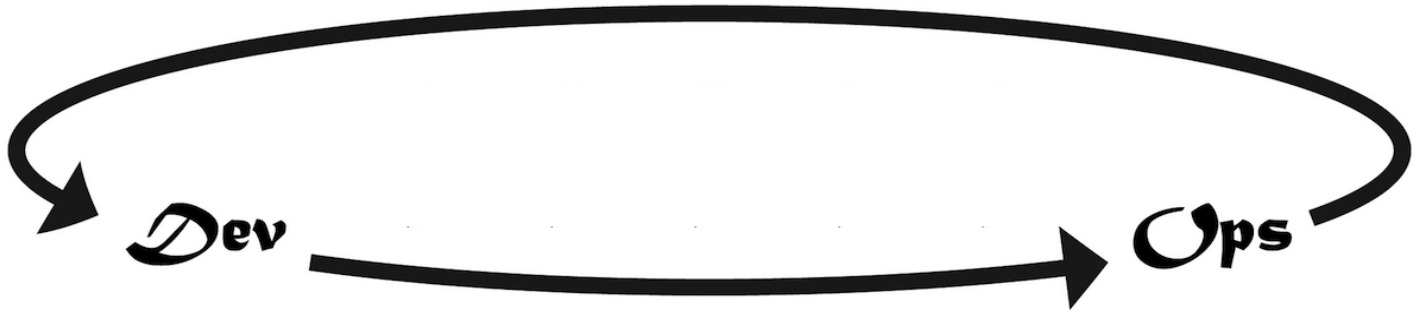
The Theory of Constraints was introduced in the book The Goal by Eliyahu M. Goldratt.

Common Constraints

- Development delays
- Environment creation (test, staging, production, etc.)
- Code deployment
- Test setup and run
- Security or QA assessments
- Overly tight architecture
- Product management
- Complex or bureaucratic processes

What are your organization's main constraints affecting flow?

The Second Way: Feedback

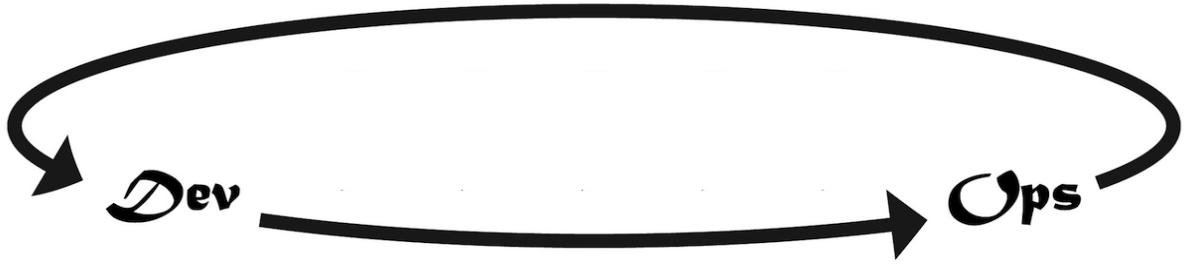


- Understand and respond to the needs of all customers – both internal and external
- Shorten and amplify all feedback loops
- Create and embed knowledge where needed

A goal of The Second Way is to shorten and amplify right to left feedback loops so necessary corrections can be continually made.

Examples of Feedback Loops

- Automated testing
- Peer review of production changes
- Monitoring/Event Management data
- Dashboards
- Production logs
- Process measurements
- Post-mortems
- Shared on-call rotation
- Change, Incident, Problem and Knowledge Management data



The Third Way: Continual Experimentation and Learning

The Third Way encourages a culture that fosters two things: continual experimentation, taking risks and learning from failure; and understanding that repetition and practice is the prerequisite to mastery.

- Allocate time for the improvement of daily work
- Create rituals that reward the team for taking risks
- Introduce faults into the system to increase resilience
- Plan time for safe experimentation and innovation (hackathons)

"Antifragility is beyond resilience or robustness. The resilient resists shocks and stays the same; the antifragile gets better."

Nassim Nicholas Taleb

Practicing for Failure Response

- The ‘Simian Army’ concept was first adopted by Netflix as a service that randomly terminates a production instance
- Response to attacks helps to build competencies to recover the production environment from inevitable failures

Getting stronger through failure is the basis of anti-fragility.

Tools like the Chaos Monkey

- Apply monitoring, diagnostics, randomization and disruption to the infrastructure
- Ensure engineers use automation to limit the user impact when big problems do occur

49

In Netflix' Own Words

"Chaos Monkey is a tool that randomly disables our production instances to make sure we can survive this common type of failure without any customer impact.

The name comes from the idea of unleashing a wild monkey with a weapon in your data center (or cloud region) to randomly shoot down instances and chew through cables -- all the while we continue serving our customers without interruption.

By running Chaos Monkey in the middle of a business day, in a carefully monitored environment with engineers standing by to address any problems, we can still learn the lessons about the weaknesses of our system, and build automatic recovery mechanisms to deal with them.

So next time an instance fails at 3 am on a Sunday, we won't even notice."

Source: http://techblog.netflix.com/2011/07/netflix-simian-army.html

Encourage a Learning Culture

- Encourage daily learning and knowledge sharing
- Create training and skills-based education plans
- Incorporate learning into processes
- Use technology to accelerate learning
- Make work educational through experimentation, problem solving and demonstrations
- Allow and use mistakes as sources of learning
- Make the results of learning visible

"An organization's ability to learn, and translate that learning into action rapidly, is the greatest competitive advantage."

Jack Welsh

DEVOPS PRACTICES

"If agile was the opening act,
continuous delivery is the headliner."

Kurt Bittner

DevOps Practices

Practices and Principles

- Continuous Integration
- Continuous delivery and deployment
- Continuous testing
- Rugged DevOps
- DevSecOps
- ChatOps
- Kanban

DevOps and Existing Frameworks

- Agile
- Scrum
- ITSM/ITIL®
- Agile Service Management®
- Lean

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY

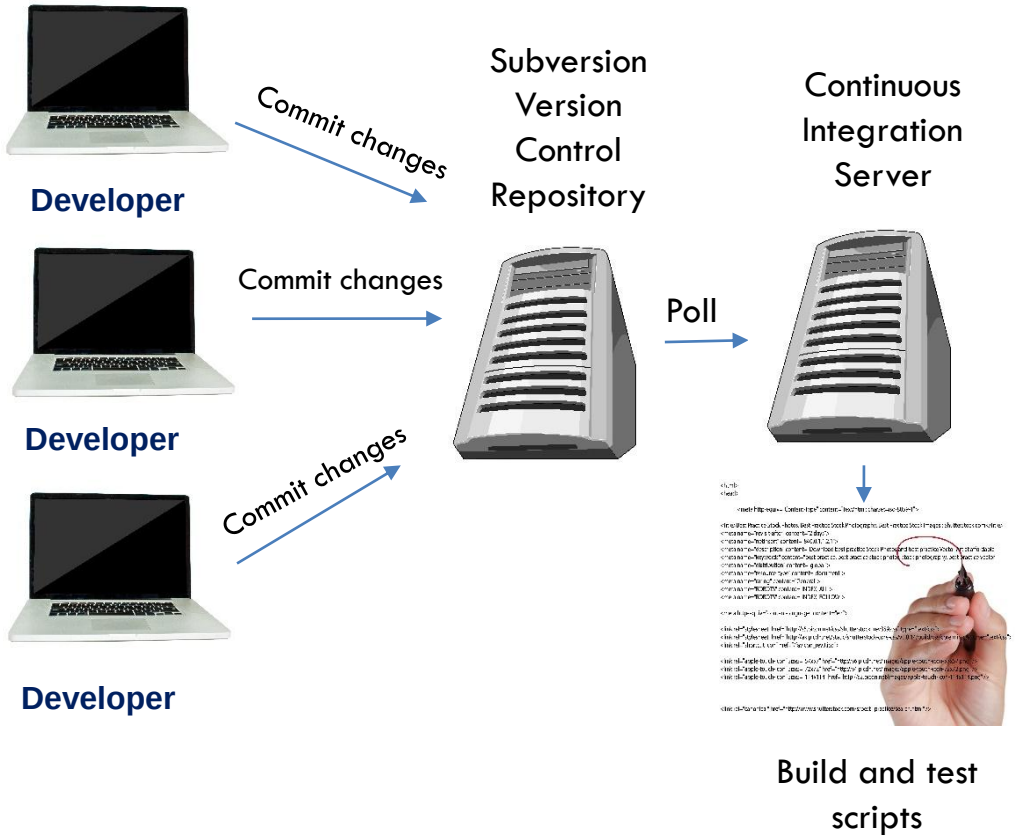
Continuous Integration (1)

Continuous integration (CI) is a development practice that requires developers to commit code into a shared repository at least daily.

- Each check-in is validated by
  - An automated build
  - Automated unit, integration and acceptance tests
- Is dependent on consistent coding standards
- Requires subversion version control repositories and CI servers to collect, build and test committed code together

Continuous Integration (2)

- Runs on production-like environments
- Integrates multiple code branches into a trunk (also known as a master)
- Must pass unit, acceptance and integration tests
- Allows for early detection and quick remediation of errors from code changes before moving to production



While mostly associated with agile software development, waterfall approaches can also take advantage of continuous integration and test-driven development practices.

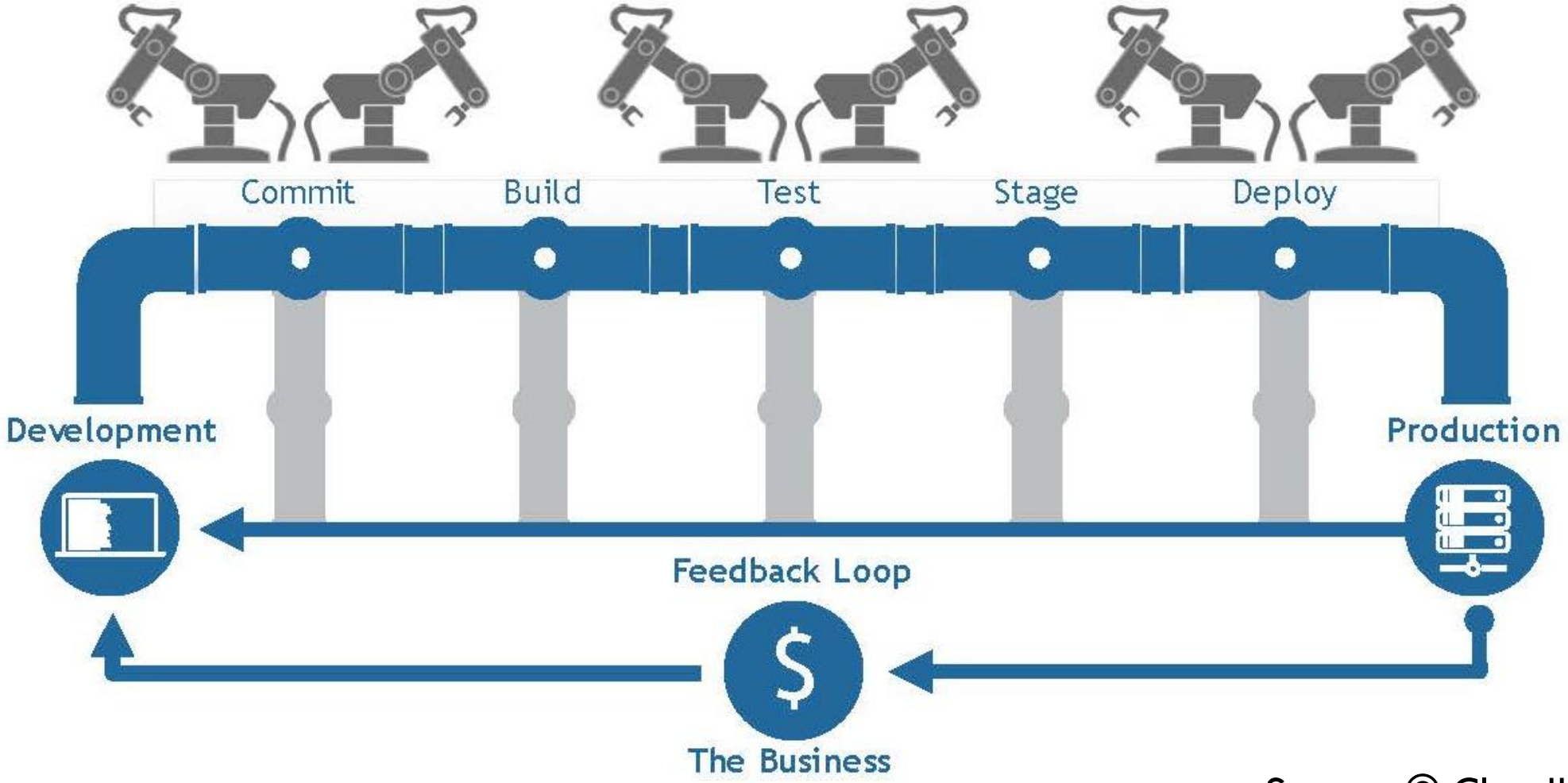
Continuous Delivery

Continuous delivery is a methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.

- Takes continuous integration to the next level
- Provides fast, automated feedback on a system's production-readiness
- Prioritizes keeping software deployable over working on new features
- Relies on a deployment pipeline that enables push-button deployments on demand
- Reduces the cost, time, and risk of delivering incremental changes

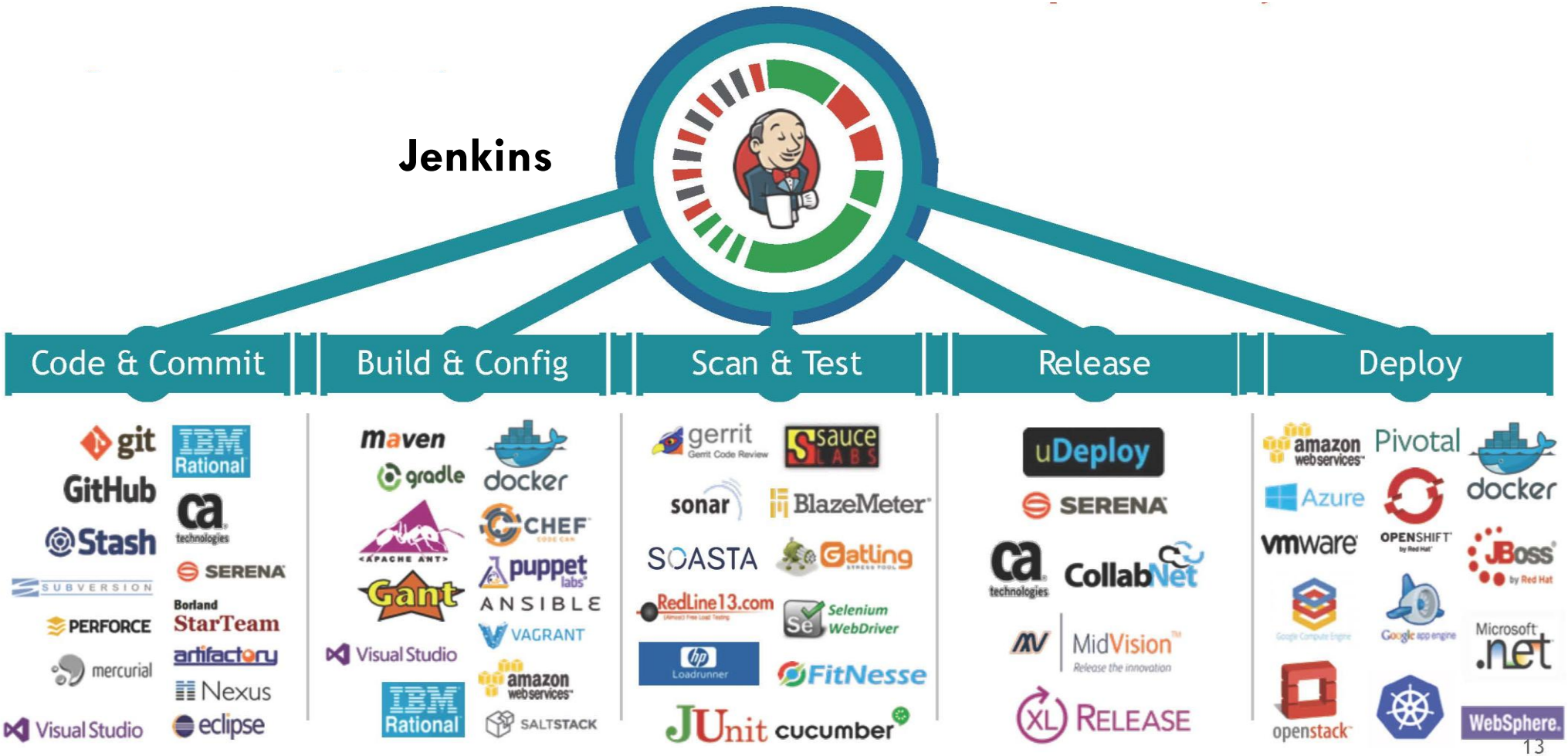
Continuous delivery is not the same as continuous deployment.

The Stages of the Deployment Pipeline



Source: © Cloudbees

CI and CD Are Built on Platforms and Ecosystems



13

While many of these tools are open source, how they are adapted and integrated into a DevOps toolchain will determine their value.

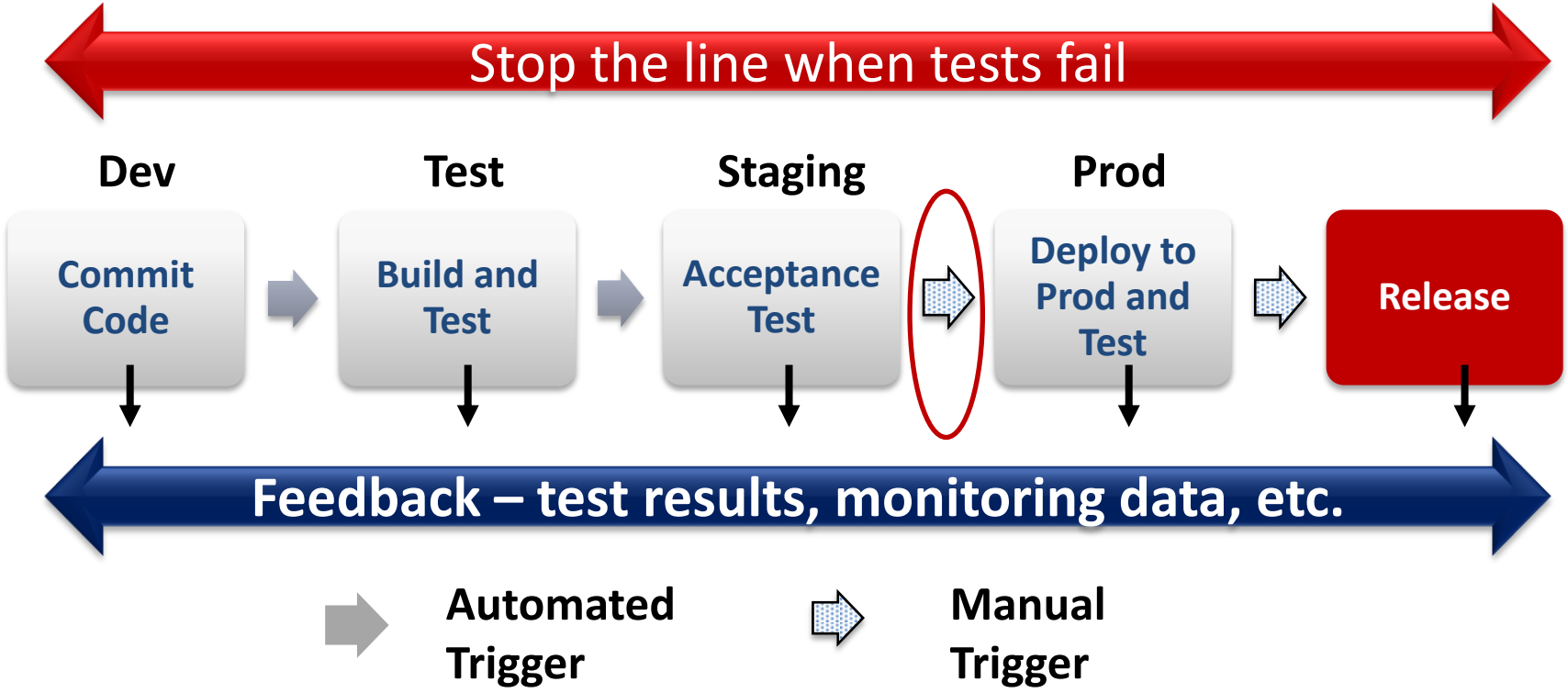
and many more…

Source: © Cloudbees

60

Continuous Delivery and Testing

Automated tests in production-like environments assure the code and environment operate as designed and are always in a deployable state.



<u>Deployment</u> is the installation of a specified version of software to a given environment (e.g., promoting a new build into production).

Continuous Testing

Continuous testing is the process of executing automated tests as part of the deployment pipeline to obtain immediate feedback on the business risks associated with a software release candidate.

- Functional
  - Unit tests, API, integration, system testing
- Non-functional
  - Performance, security, compliance, capacity

Automated testing is a key aspect of Continuous Integration and Continuous Delivery and encourages a "shift left" testing strategy.

Testing and Deployment Strategies

- Test driven development (TDD)
- Performance testing
- End-to-end use-case testing
- Security testing
- Manual testing
- Testing in production

"Shifting left" is about building quality into the software development process. When you shift left, fewer things break in production, because any issues are detected and resolved earlier.

2016 State of DevOps Report

Continuous Deployment (1)

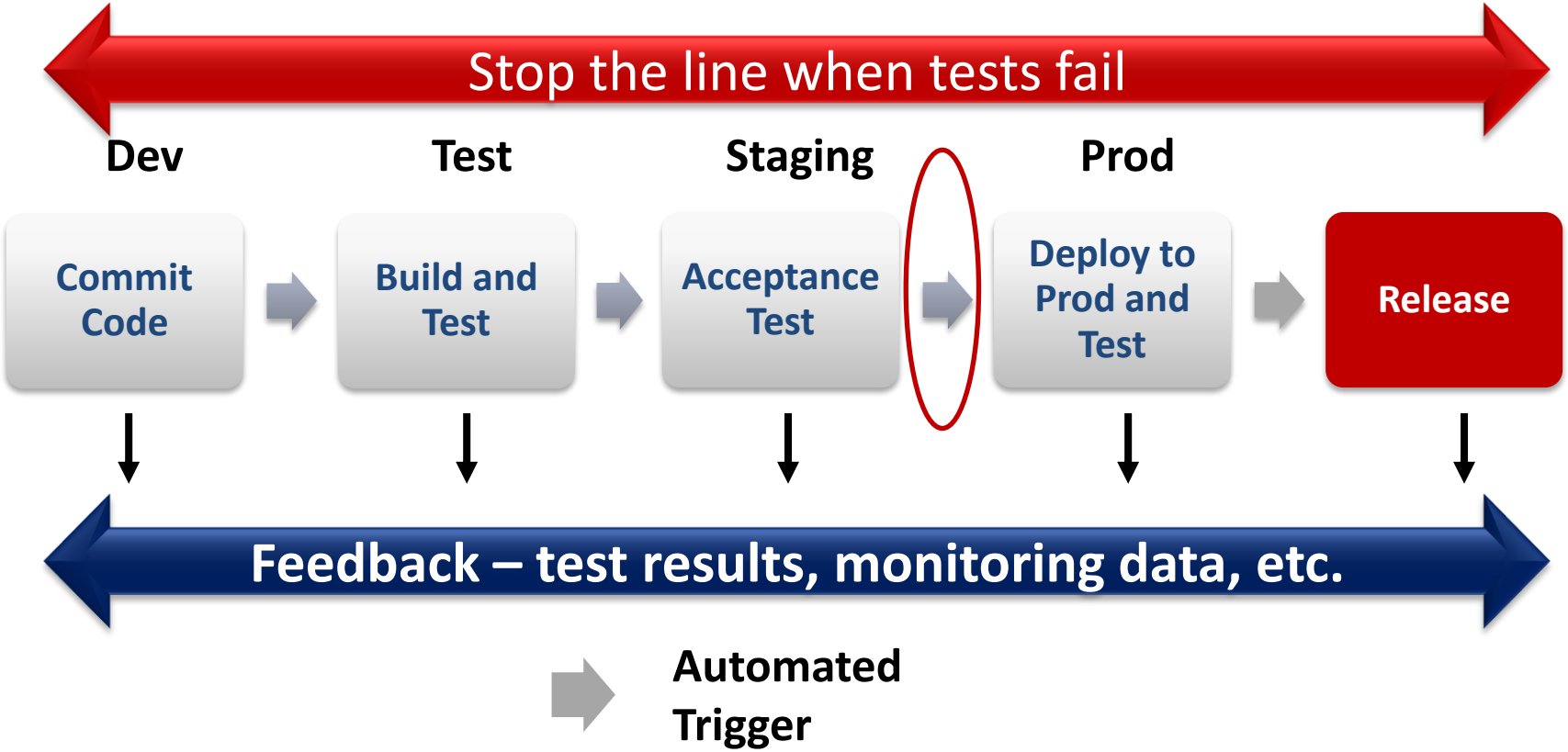
Continuous deployment is a set of practices that enable every change that passes automated tests to be automatically deployed to production.

- Removes the manual step in the deployment pipeline
- Results in multiple deployments per day

Continuous deployment may not be practical or possible for companies constrained by regulatory or other requirements.

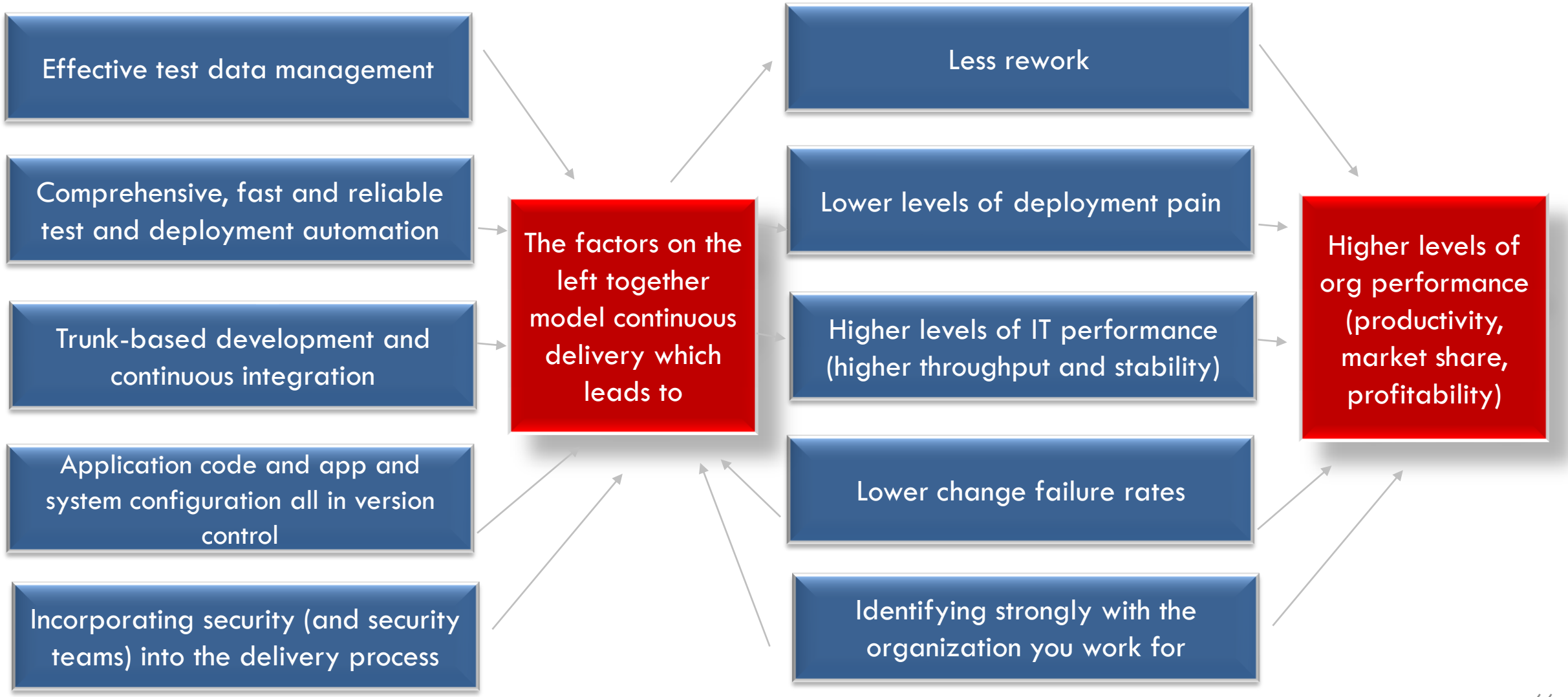
Continuous Deployment (2)

Code deployed into production may be invisible to customers, but features can be run and tested by internal staff.



<u>Release</u> is the process or event of making a feature (or set of features) available to a segment of customers.

Continuous Delivery Can Lead to Higher IT and Business Performance



66

Source: 2016 State of DevOps Report

OTHER DEVOPS PRACTICES

Rugged DevOps

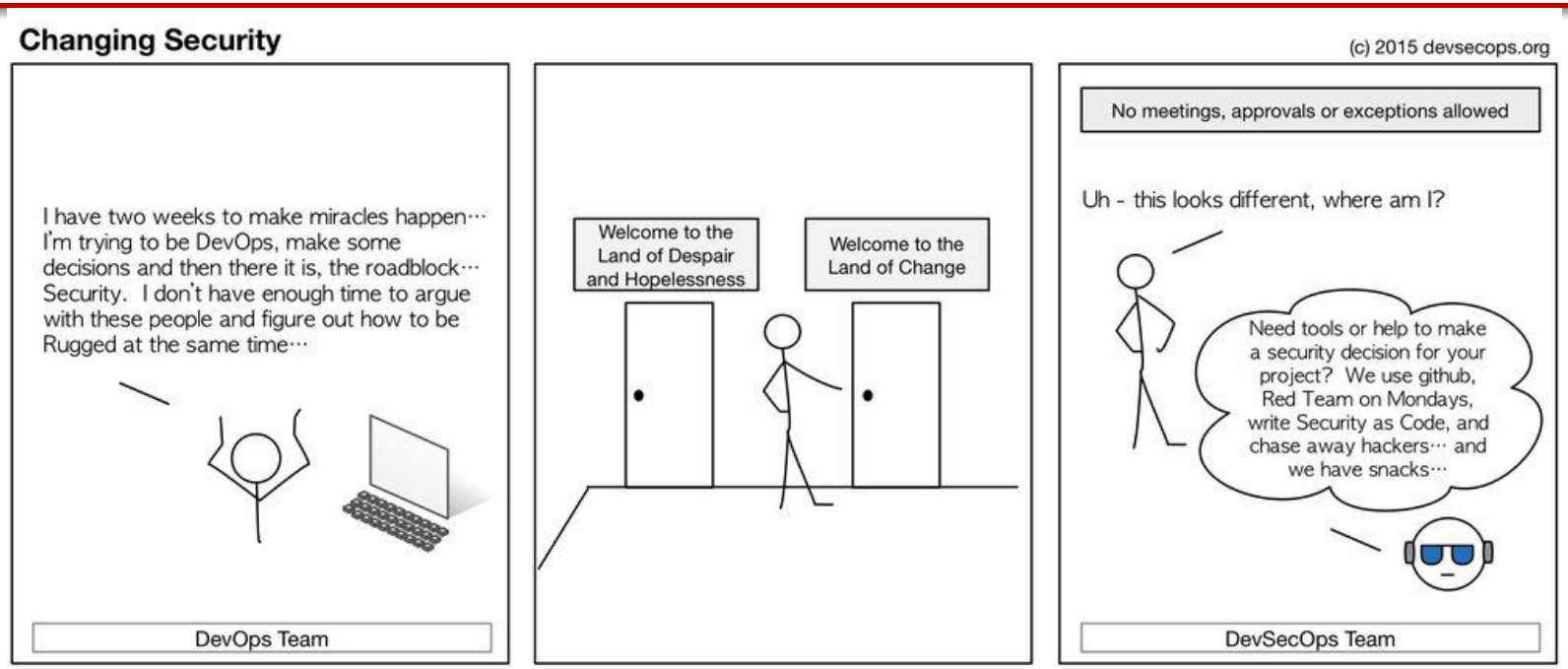
Rugged DevOps is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone. (Forrester)

- Rugged software development is a cultural approach to creating available, survivable, defensible, secure, and resilient software
- Rugged organizations are comfortable with instrumentation, experimentation and experience
- It is not the same as DevSecOps but related

http://devops.com/2015/04/06/rugged-devops-idea-automation-impact/

DevSecOps

The purpose and intent of DevSecOps is to build on the mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.



www.devsecops.org

- Introduces security as code
- Embraces the "shift left" testing strategy
- Leverages automation for resilience, testing, detection and audit

69

ChatOps

Chat client + chat bots = conversation-driven development, delivery and support.

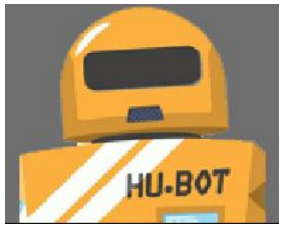
slack

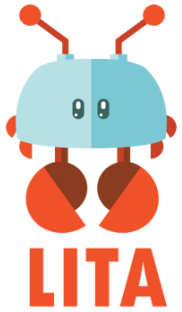
VictorOps

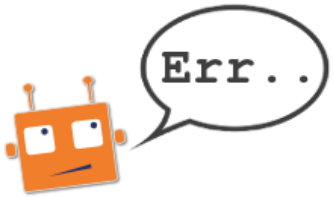
Atlassian HipChat

Campfire™

CA FLOWDOCK







Eric Sigler	I think something's up with the latest changes to Apollo.
Eric Sigler	!lock apollo production
Officer URL	✓ apollo production locked
Eric Sigler	Here's a log snippet from the web server:
Eric Sigler	!enhance "10.0.0.1", "-", "02/Dec/2014:06:34:32 +0000", "GET /someurl HTTP/1.0", "-", "-", "200", "3704", "-", "a-totally-fake-user-agent", "127.0.0.1"
Officer URL	"*a-fake-production-webserver-dns-name* (US-WEST-2B, environment: production)", "-", "02/Dec/2014:06:34:32 +0000", "GET /someurl HTTP/1.0", "-", "-", "200", "3704", "-", "a-totally-fake-user-agent", "127.0.0.1"
Eric Sigler	!status
Officer URL	Status: NORMAL
Eric Sigler	!define me apollo
Officer URL	apollo is a highly available service that creates and delivers On-call Handoff Notifications to Mercury.
Eric Sigler	All clear
Eric Sigler	!unlock apollo production
Officer URL	✓ apollo production unlocked

The transparency of ChatOps shortens feedback loops, improves information sharing, enhances team collaboration and enables cross-training. It can also be used to decrease MTTR .

Kanban

Kanban is a method of work that pulls the flow of work through a process at a manageable pace.

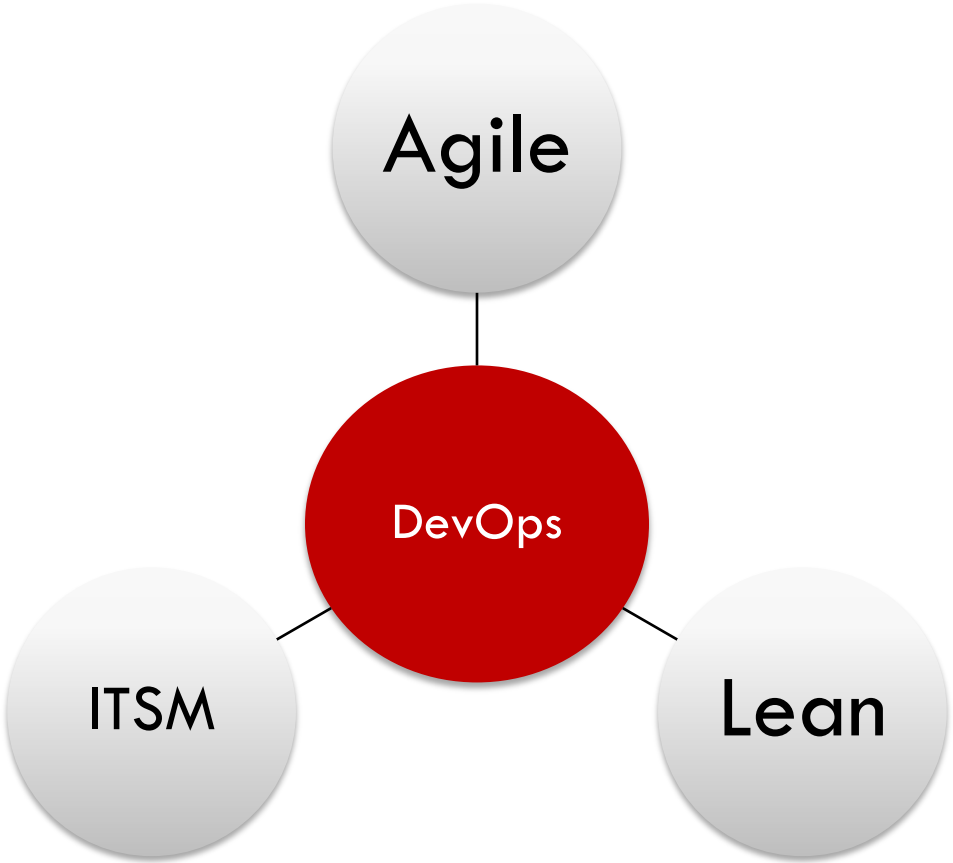
- Visualizes and manage workflow
- Pulls work for teams when they are ready for it
- Enables people to work collaboratively to improve flow
- Measures team velocity (quantity of work done in an iteration)
- Reduces idle time and waste in a process

- Makes work visible
- Make policies explicit
- Limits work in progress (WIP) to capacity

Why is Work In Progress bad?

DEVOPS AND OTHER IT FRAMEWORKS

DevOps Cannot Stand Alone

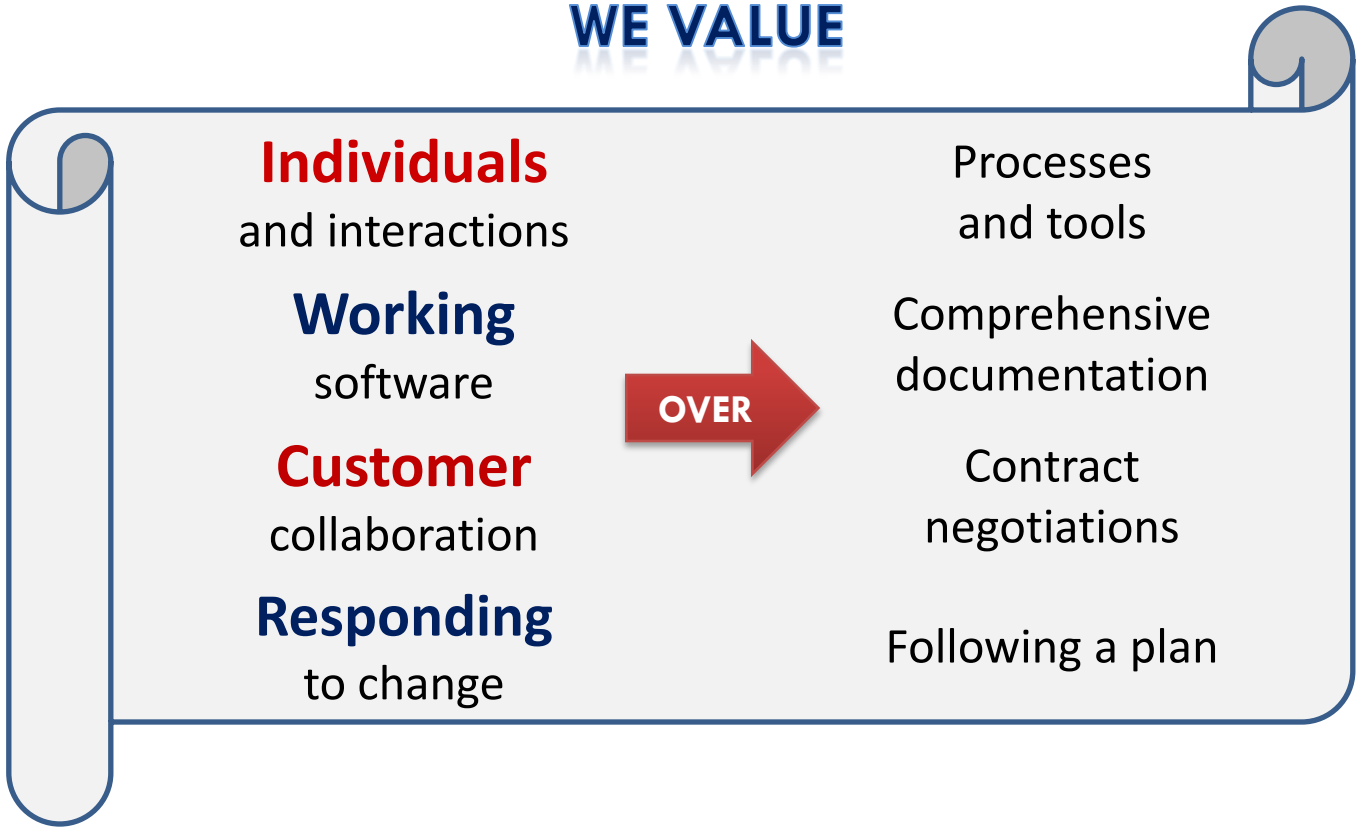


Successful DevOps relies on the adoption and integration of multiple frameworks and methodologies.


agile

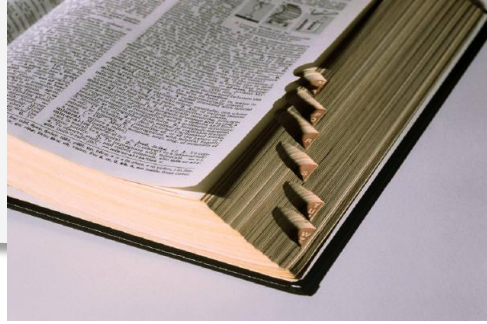
The Agile Manifesto

The underlying concepts of agile software development were first laid out in the Agile Manifesto.



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

Important Terms



- **Agile** (adjective)
  - Able to move quickly and easily; well-coordinated
- **Agile enterprise**
  - A fast moving, flexible and robust company capable of rapid response to unexpected challenges, events and opportunities
- **Agile software development**
  - A group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams

Agile software development methods deliver working software in smaller and more frequent increments.

What Does it Take to "Be Agile"?

- Be customer-centric
- Be lean
- Be collaborative
- Be communicative
- Be adaptive
- Be measurable
- Be consistent
- Be results-oriented
- Be reflective

It is more important to "be agile" than to "do agile."

Scrum

Scrum is a simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "just enough" structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.

Scrum.org

- Scrum is
  - The most commonly applied Agile software development practice
  - Deceptively simple yet difficult to master
  - Not a process or a technique for building *products*

Scrum increases the ability to release more frequently.

http://www.scrumguides.org/

Scrum Basics

- **Roles**
  - Product Owner
  - ScrumMaster
  - Team

- **Artifacts**
  - Product Backlog
  - Sprint Backlog
  - Burndown Charts
  - Potentially Shippable Increment

- **Meetings**
  - Release Planning
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

Scrum In a Nutshell

Scrum = 3 Roles + 4 Artifacts + 5 Events

Team

Product Owner

Scrum Master

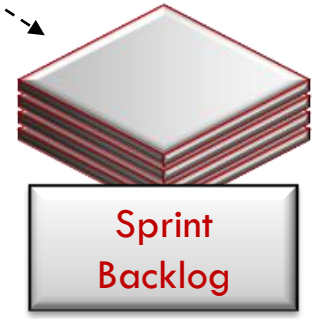
Scrum is based on timeboxed iterations.

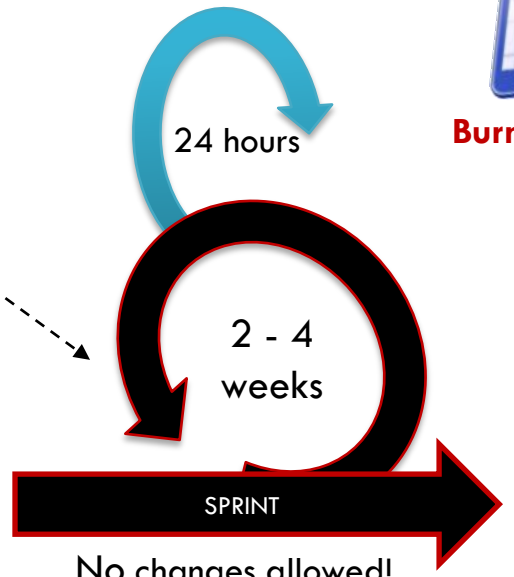
Release Planning Meeting (Optional)

Sprint Planning Meeting (4-8 hours)

Daily Scrum (15 minutes)

Burndown Chart





No changes allowed!

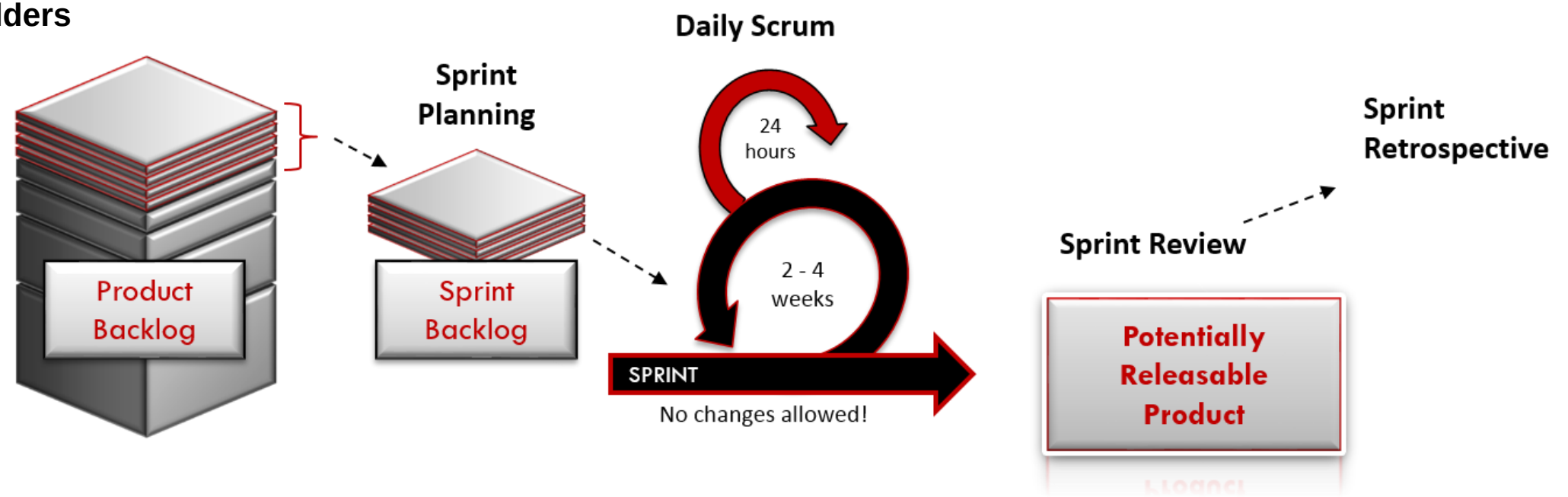
Sprint Review (2-4 hours)

Potentially Releasable Product Increment

Sprint Retrospective (1.5-3 hours)

Ops Involvement in Agile SW Development

Inputs from executives, customers, users, team, stakeholders



- Non-functional requirements
- Define Ops acceptance criteria
- Infrastructure provisioning

Operations

- Design, build and test ITSM and IT Operations automation capabilities
- Embed Ops knowledge in Dev

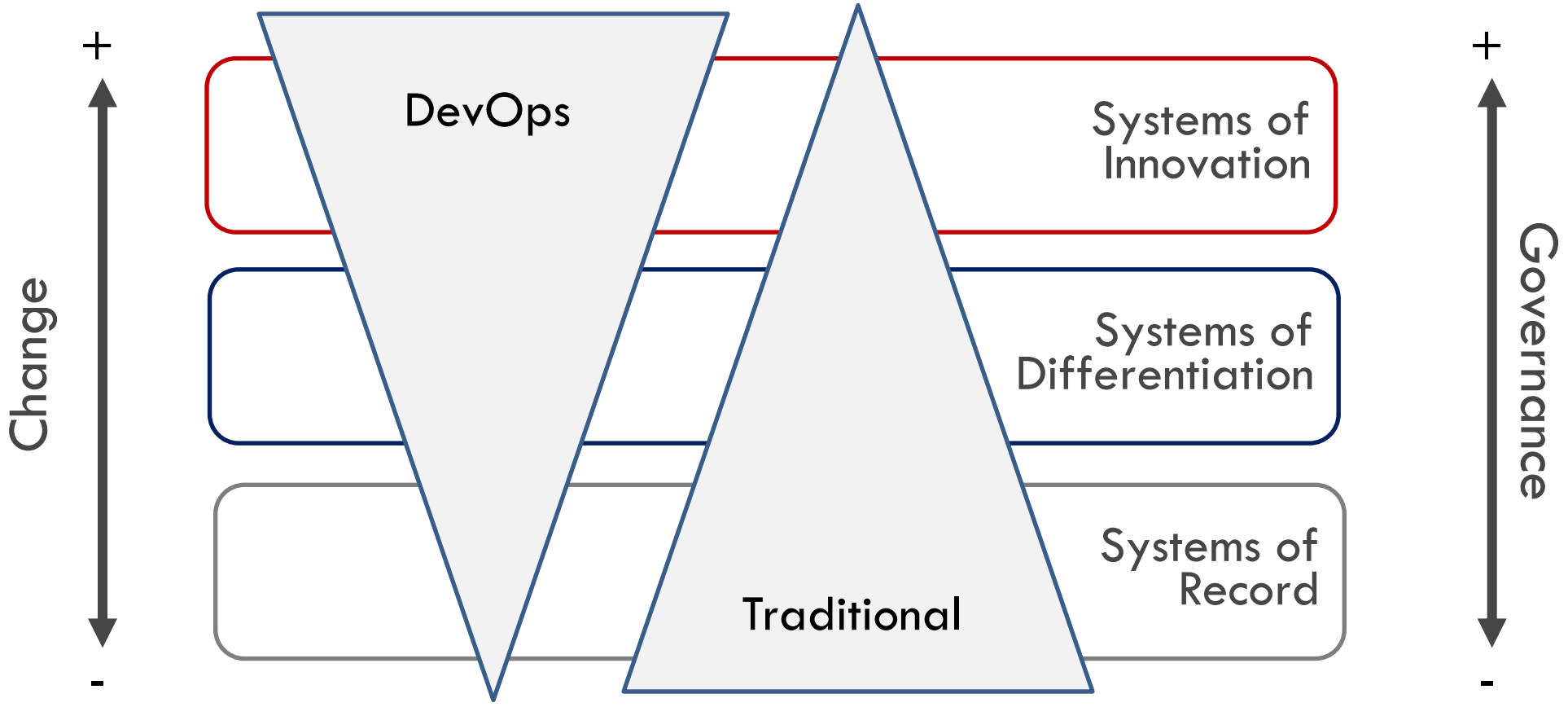
Scaled Agile Framework® (SAFE ™)

The Scaled Agile Framework (SAFE) is a proven, publicly available framework for applying Lean-Agile principles and practices at enterprise scale.

- Integrates Lean and Agile thinking into software development
- Focuses on iterative and incremental development, agile SW development, product development flow, lean thinking and field experience at enterprise scale
- Can be applied to organizations with a large number of practitioners and teams

http://www.scaledagileframework.com/

Pace-Layered Application Strategy - One Approach Doesn't Fit All



This is also where "variable speed IT" can be applied.

Source: Gartner - Pace-Layered Application Strategy

Increasing Agility

DevOps increases agility by

- Breaking down silos
- Improving constraints
- Taking a unified approach to systems engineering
- Applying agile principles to both Dev and Ops
- Sharing knowledge, skills, experience and data
- Recognizing the criticality of automation
- Deploying faster with fewer errors

DevOps extends agile principles beyond the boundaries of the software to the entire delivered service.

IT SERVICE MANAGEMENT

"It is my firm belief that ITSM and the DevOps movement are not at odds.

Quite to the contrary, they're a perfect cultural match."

Gene Kim

IT Service Management

IT service management (ITSM) is the implementation and management of quality IT services that meet the needs of the business.

- Provides guidance and structure to processes such as Change, Configuration, Release, Incident and Problem Management
- ITSM processes underpin the entire service lifecycle from strategy, design, transition, operations, continual improvement and value creation
- DevOps needs ITSM practices to meet the goal of deploying faster changes without causing disruption

Repeatable service management processes – adapted to an organization's current business needs – can lead the way to stable continuous delivery and increased flow.

IT Infrastructure Library® (ITIL®)

ITIL is a set of best practice publications for IT service management.

- ITIL Core consists of five books that provide best practice guidance for an integrated service management process approach

- ITIL® Service Strategy
- ITIL® Service Design
- ITIL® Service Transition
- ITIL® Service Operation
- ITIL® Continual Service Improvement

ITIL process improvement supports DevOps practices.

Important Terms

- **Service**
  - A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks
- **Service Management**
  - A set of specialized organizational capabilities for providing value to customers in the form of services
- **IT service**
  - A service provided by an IT service provider – made up of a combination of information technology, people and processes

Source: *ITIL® Glossary and Abbreviations*

ITSM Process Models Support DevOps and Continuous Delivery

- Predefined procedures
  - Steps to be taken
  - Chronological order and dependencies
  - Responsibilities
  - Timescales and thresholds
  - Escalation procedures
- Define steps for handling specific types of transaction
- Ensure a defined path or timeline is followed
- Can be automated

Examples

- Change models
- Release models
- Test models
- Incident models
- Problem models
- Request models

Based on ITIL Text - ST 4.2.4.5

ITSM Processes Enable DevOps

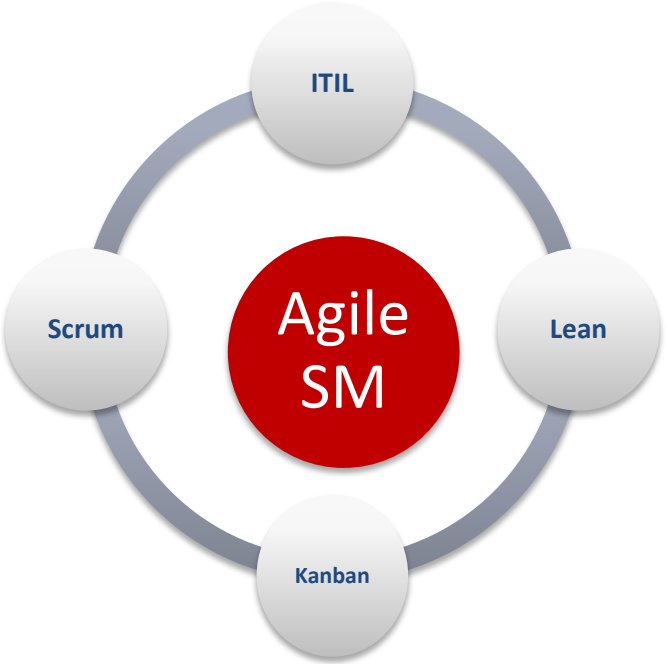
- ITSM processes that underpin DevOps include
  - Change management
  - Release management
  - Service asset and configuration management (SACM)
  - Knowledge management
  - Event management
  - Incident management
  - Problem management
  - Capacity management

and all others…

All ITSM processes are necessary to DevOps. How they are designed, implemented and managed may need to be adapted to make them more agile.

Agile Service Management

Agile Service Management (Agile SM) ensures that ITSM processes reflect Agile values and are designed with "just enough" control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.



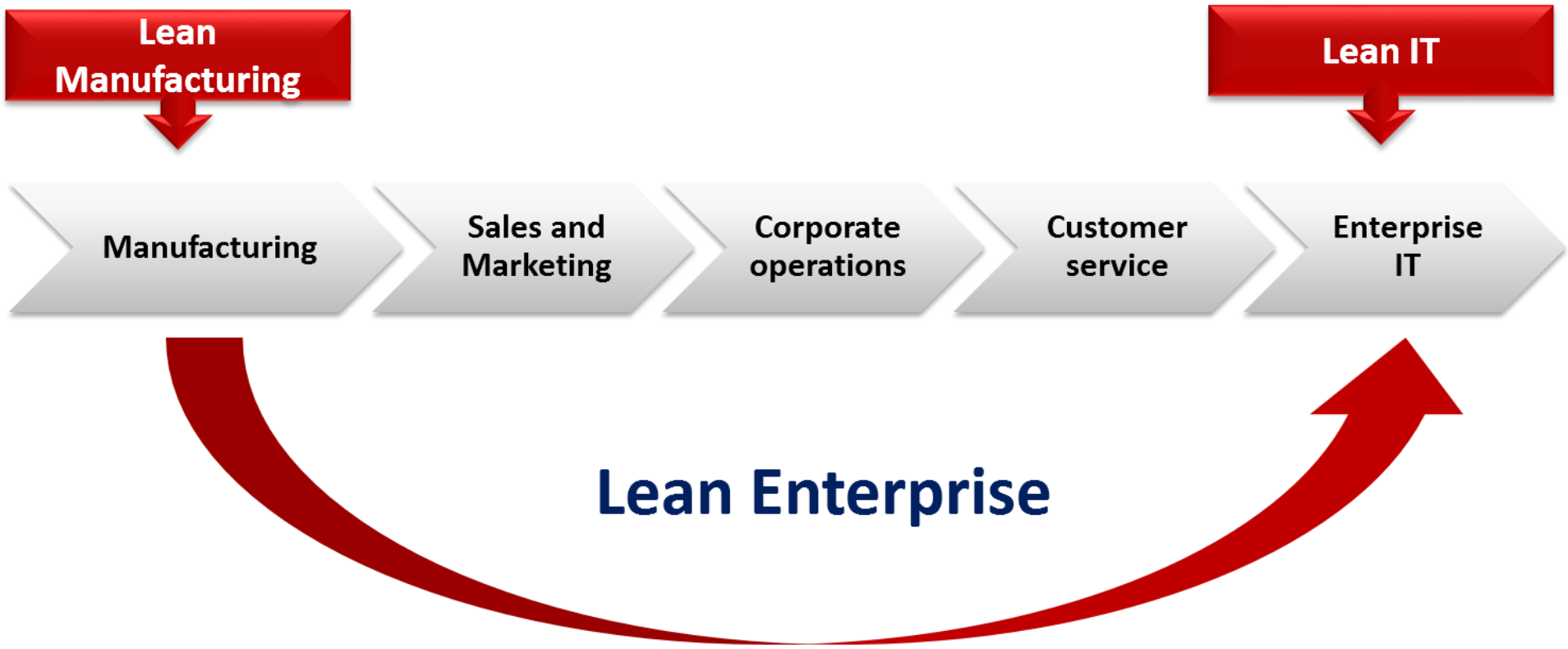
- Adapts Agile practices to ITSM process design
- Implements service management in small, integrated increments
- Ensures ITSM processes reflect Agile values from initial design through CSI
- Encourages "minimum viable" and "just enough" processes to increase speed and conformance

Agile Service Management does not reinvent ITSM – it modernizes the approach.

Source: Agile Service Management Guide


lean
enterprise
service
target
growth
process
approach
accounting
work
quality
design
relationships
feedback

Lean Perspectives



Lean IT applies the key ideas behind lean production to the development and management of IT products and services.

Important Terms

DevOps has its roots in the lean manufacturing world, which addresses the problem of engineers designing products that factories can't afford to build.

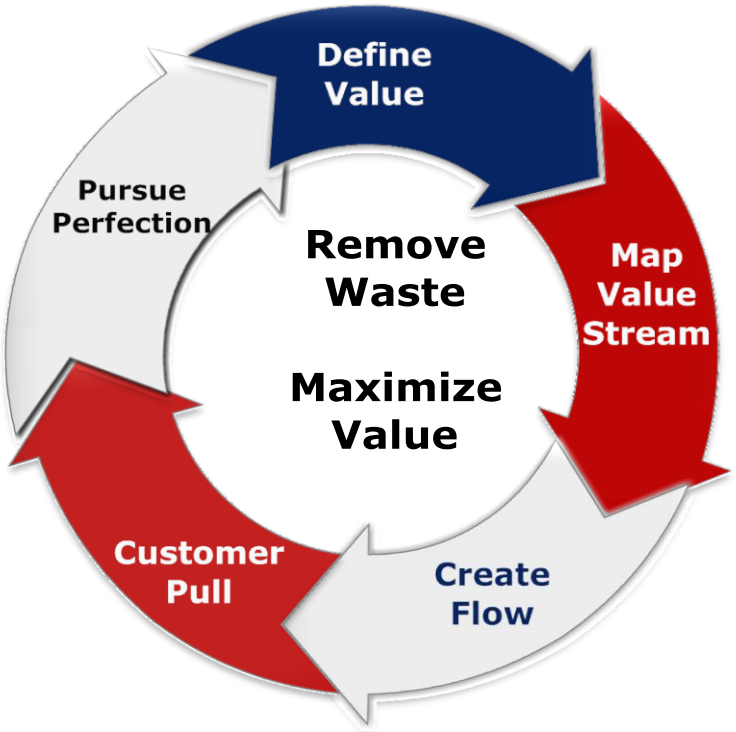
- **Lean** (adjective)
  - Spare, economical, lacking richness or abundance
- **Lean Production**
  - A production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value
- **Flow** (also known as single-piece flow)
  - Describes how people or products move through a process
- **Continuous flow**
  - Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps
- **Value Stream Mapping**
  - Lean tool that depicts the flow of information, materials and work across functional silos with an emphasis on quantifying waste, including time and quality.

Lean Tools

Tool	Purpose
A3 thinking	Problem solving
Continuous flow	Eliminating waste
Kaizen	Continuous improvement
Kanban	Pull system
KPI	Key Performance Indicator
Plan-do-check-act	Deming continuous improvement cycle
Process mapping	Visualize process to identify areas of improvement
Root cause analysis	Identify source of errors and waste
SMART goals	Specific, measurable, achievable, relevant, time-bound
Value stream mapping	Depict flow

Five Principles of Lean Thinking

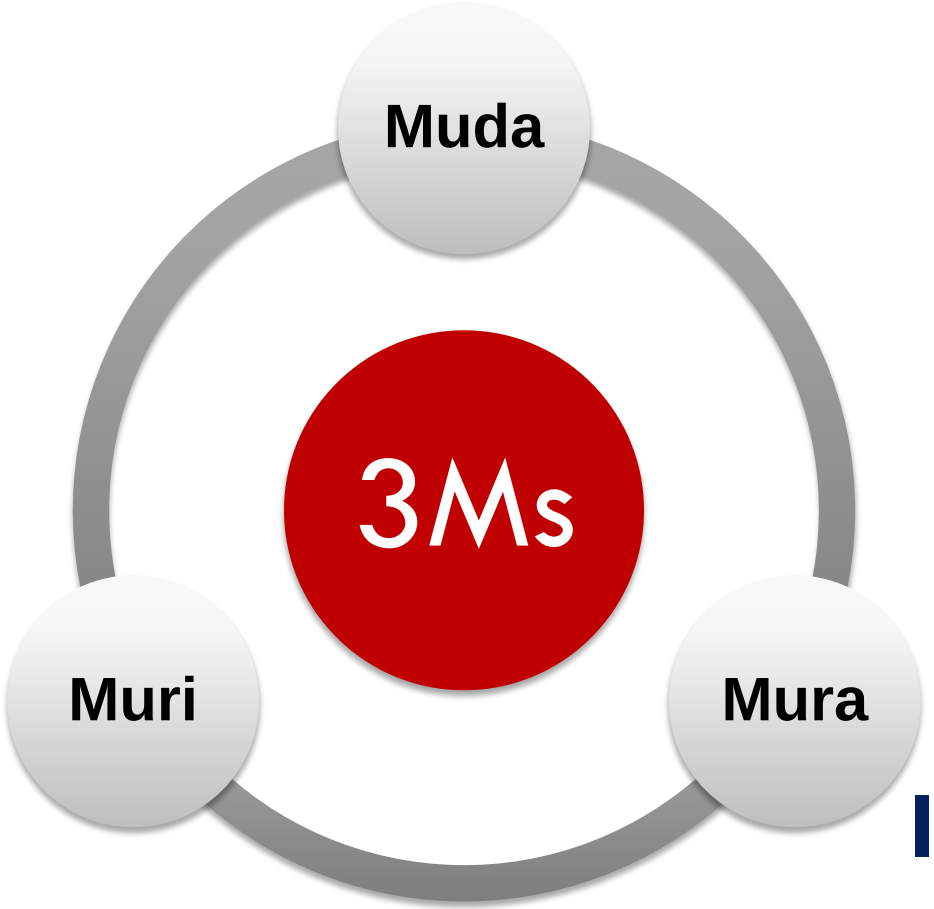
1. Define value precisely from the perspective of the end customer
2. Identify the entire value stream for each service, product or product family and eliminate waste
3. Make the remaining value-creating steps flow
4. As flow is introduced, let the customer pull what the customer wants when the customer wants it
5. Pursue perfection



The voice of the customer (VOC) process captures and analyzes customer requirements and feedback to understand what the customer wants.

The 3Ms of Lean

Waste



Overburden

Inconsistency

Both mura and muri cause muda (waste).

Sources of Waste = DOWNTIME

The goal of lean thinking is to create more value for customers with fewer resources and less waste. Waste is any activity that does not add value to the process.

Source	Purpose	Examples
Defects	Deviations from requirements; errors	Failures, known errors, misinformation
Overproduction	Producing more or faster than required	Excessive documentation or code
Waiting	Delays while waiting on a previous step	Delayed decisions, approvals, response
Non-use	Unused knowledge or creativity	Unused skill, innovation, communication
Transportation	Moving products from one location to another	Multiple hand-offs, emails or meetings
Inventory	Carrying more materials than needed	Unused software, infrastructure, excessive backlogs or emails
Motion	Moving people or assets more often than required	Moving code or infrastructure too much
Excessive processing	Doing more than is required	Over-engineering, failing to create templates and other reusable assets

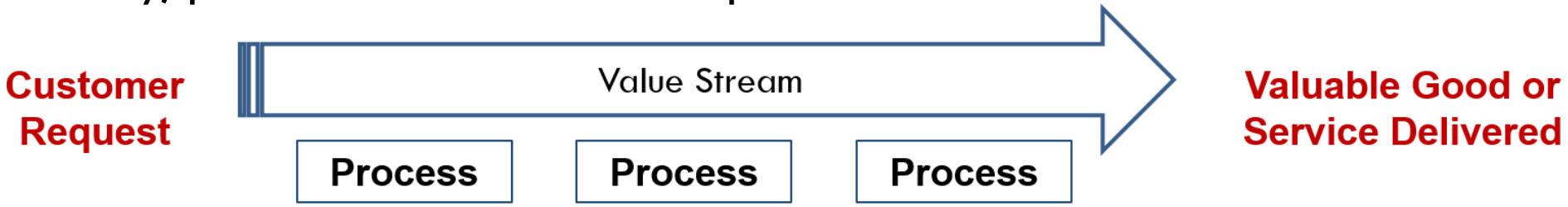
LEAN PRACTICES

Lean practices are so relevant to DevOps, Lean was recently added as one of the core DevOps values.

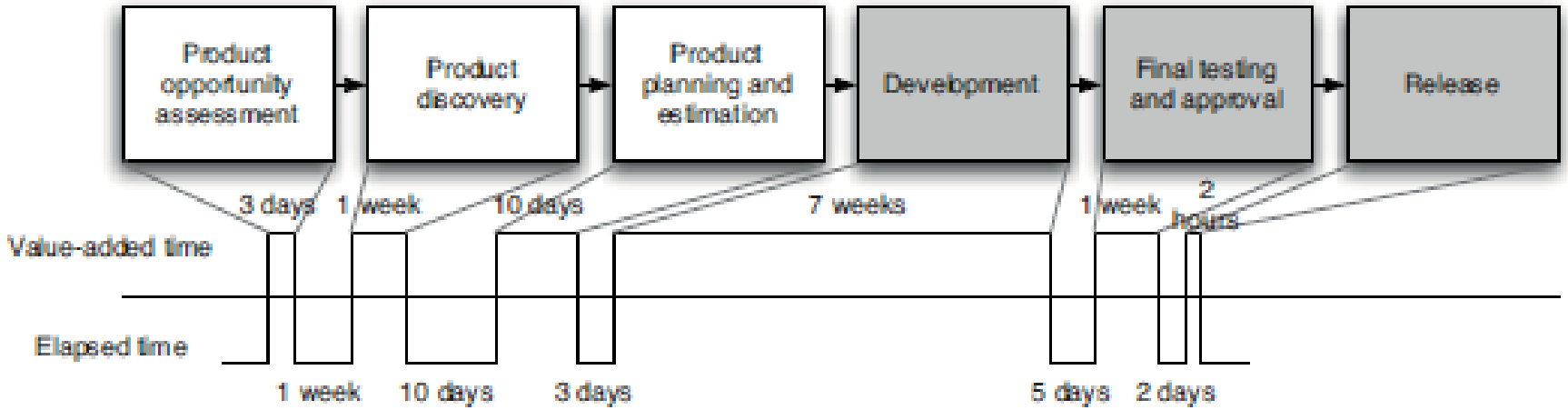
Value Stream Mapping

Value stream mapping is a lean tool that depicts the flow of information, materials and work across functional silos with an emphasis on quantifying waste, including time and quality.

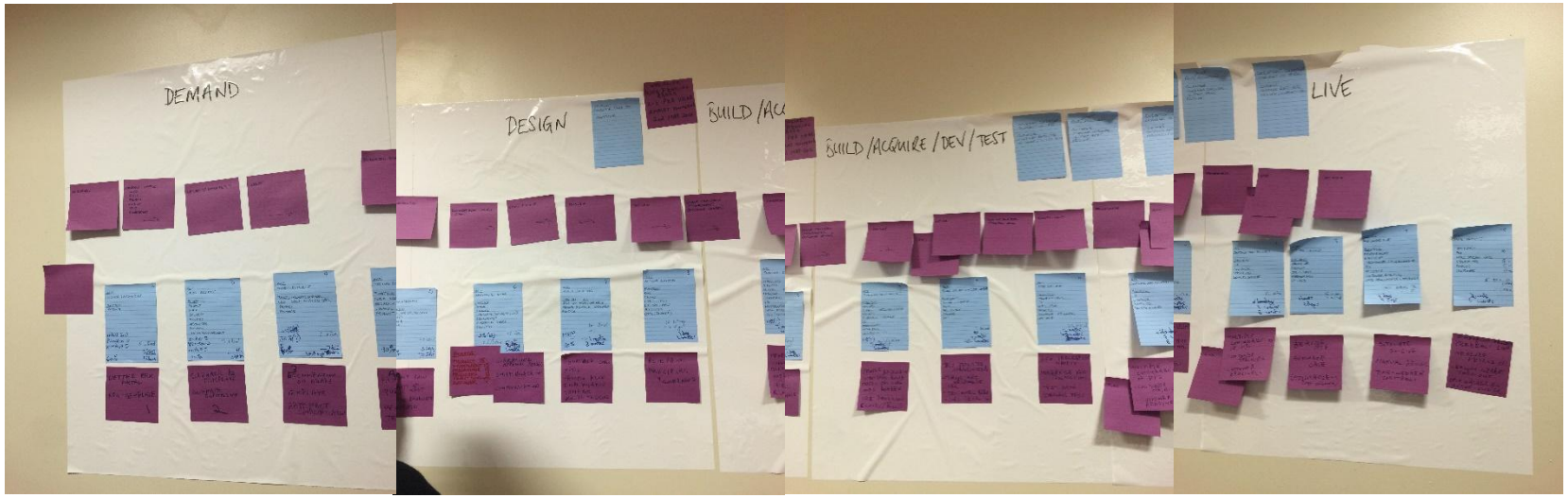
- A value stream is the sequence of activities required to design, produce, and deliver a specific product or service
- Value streams typically span multiple processes
- Value stream mapping enables cross-functional teams to
  - See an entire value stream from a work and information flow perspective
  - Identify areas of non-value waste that could be eliminated in an effort to improve flow and deliver greater value
  - Identify, prioritize and measure improvements



Sample Value Stream Maps



Source:
Jez Humble -*Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*



Source:
Daniel Breston, Ranger4

Improvement Kata

The Improvement Kata grew out of the Toyota Production System as a structured way to create a culture of continuous learning and improvement.

- A kata is any structured way of thinking and acting that you practice until the pattern becomes a habit
- Through practice, a pattern of behavior becomes second nature

Practicing the Improvement Kata should be a daily occurrence, not just an add-on or 'as time allows' project.

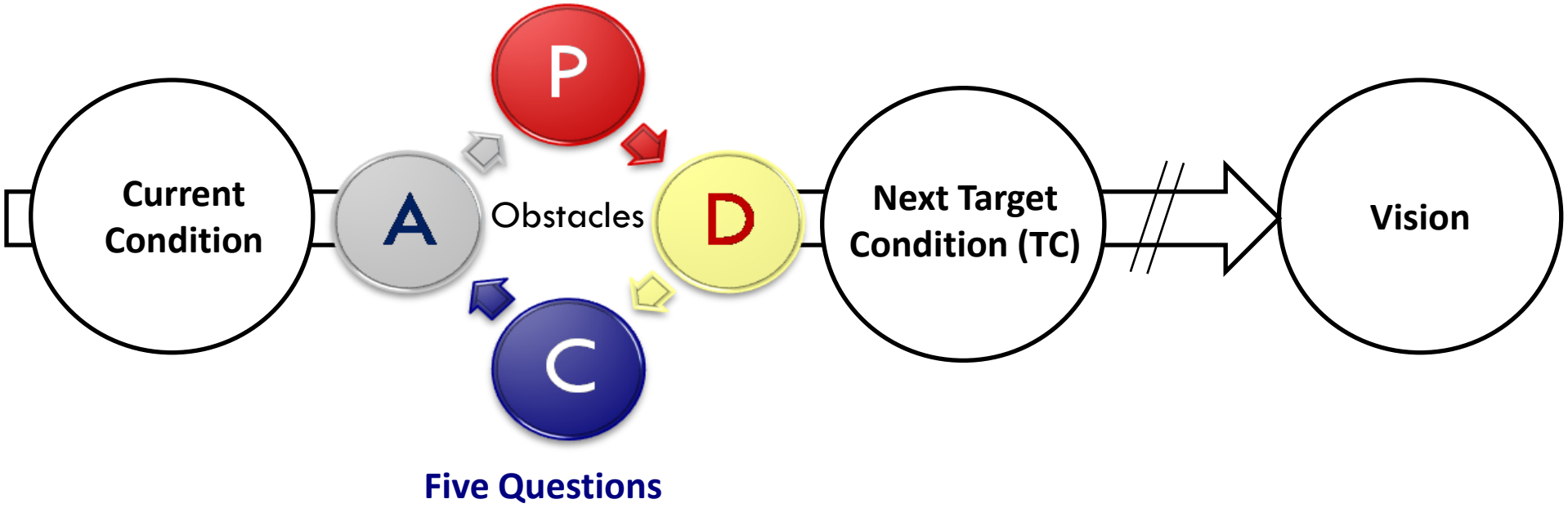
The Improvement Kata – Four Steps

2 Grasp the current condition

4 PDCA and experiment toward the target condition

3 Establish the next target condition

1 Understand the long-term vision or direction



The Improvement Kata is a four-step process that focuses on learning and improving work. It considers the organization's long-term vision or direction.

Improvement Kata – Five Questions

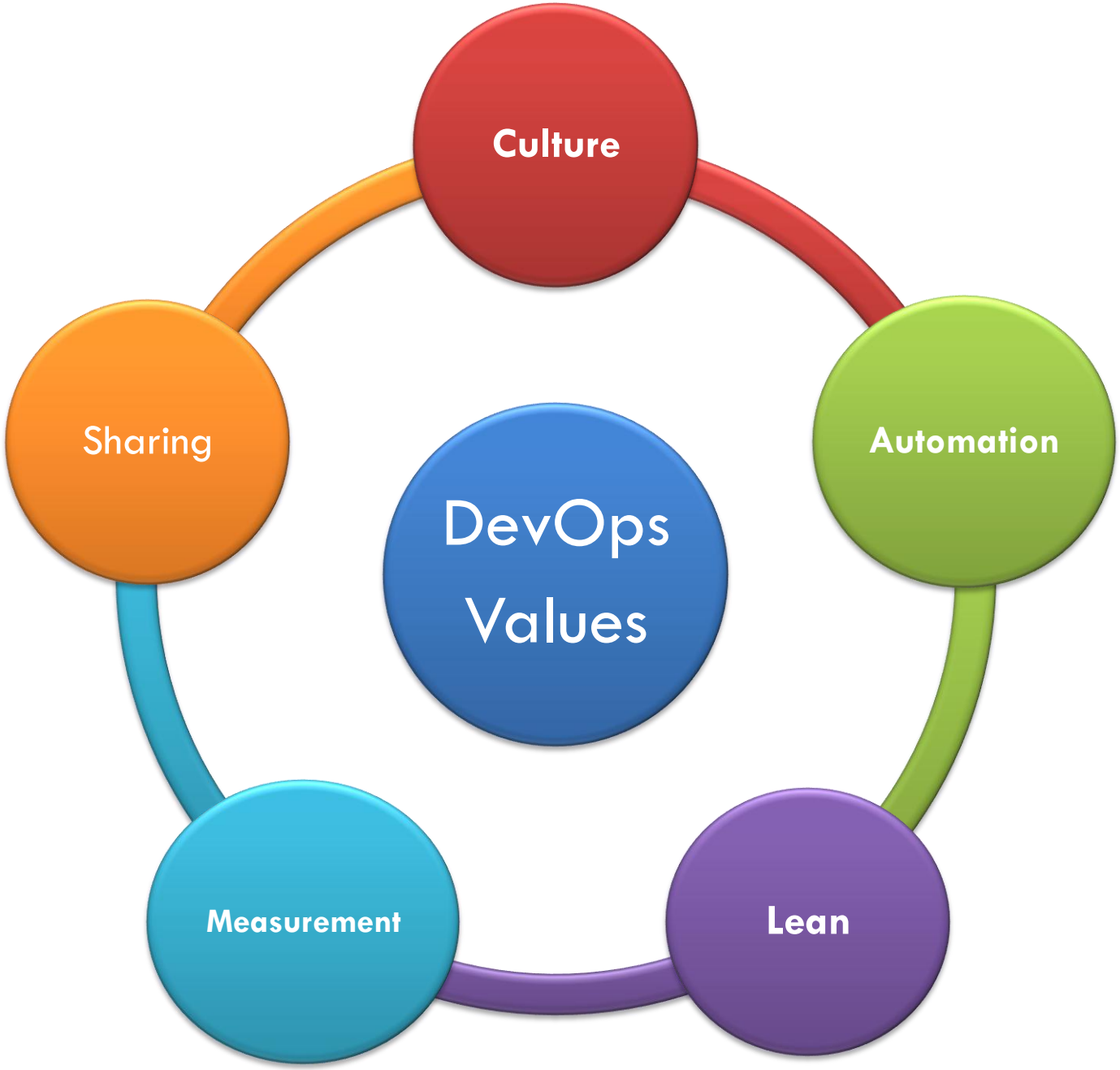
1. What is the target condition?
2. What is the actual condition now?
3. What obstacles do you think are preventing you from reaching the target condition?
4. What is your next step? (next PDCA/experiment)
5. When can we go and see what we have learned from taking that step?

Teams using the Improvement Kata learn as they strive to reach a target condition, and adapt based on what they are learning.

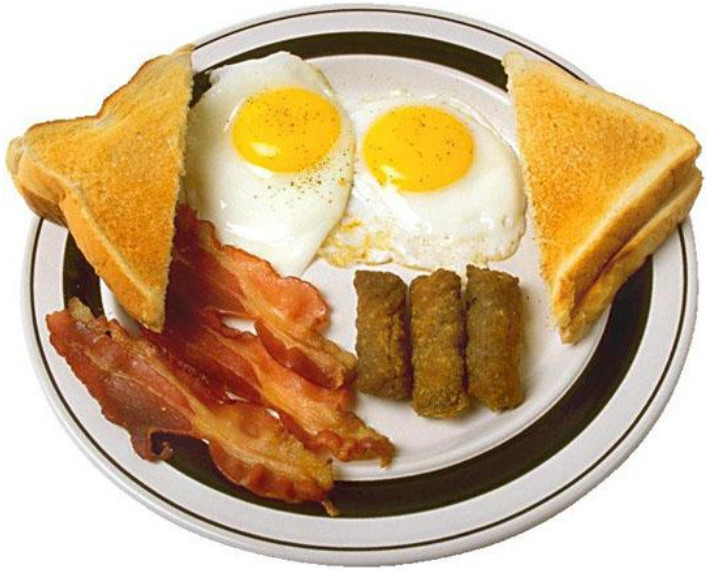
Practices Alone Are Not Enough

"Tools and processes are a reflection of your cultural choices."

Sascha Bates


Culture
Automation
Lean
Measurement
Sharing
DevOps
Values

CULTURE



"Culture eats strategy for breakfast."

Peter Drucker

What is Organizational Culture?

The values and behaviors that contribute to the unique social and psychological environment of an organization.

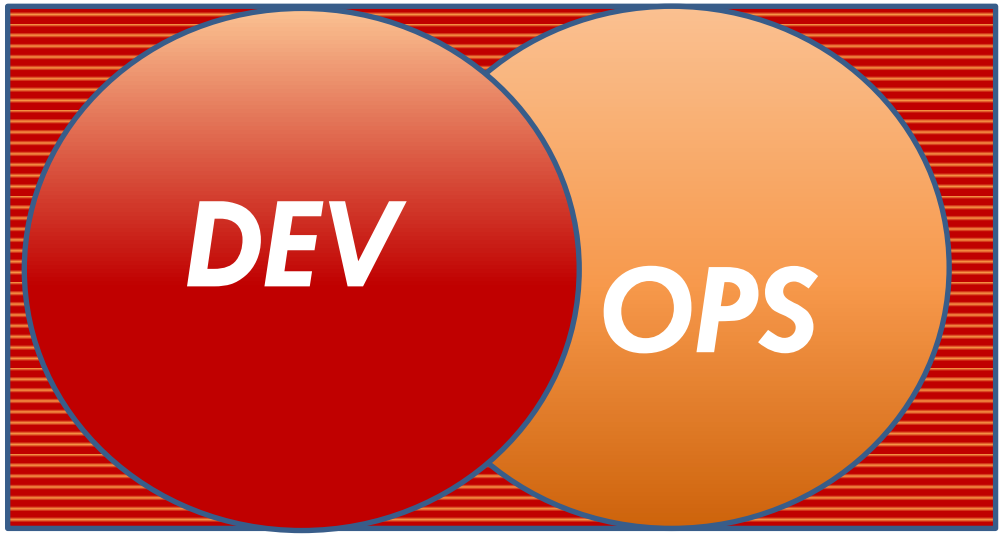
www.businessdictionary.com

"You can't directly change culture.
But you can change behavior, and behavior becomes culture."

Lloyd Taylor

DevOps Helps to Overcome Cultural Debt

Cultural debt occurs when cultural considerations are disregarded or deferred in favor of growth and innovation.



IT's silo culture and other organizational challenges are a direct result of disregarding cultural considerations in favor of rapid increases in corporate technology. The due date is today!

Characteristics of a DevOps Culture

- Shared vision, goals and incentives
- Open, honest, two-way communication
- Collaboration
- Pride of workmanship
- Respect
- Trust
- Transparency
- Continuous improvement
  - Experimentation
  - Intelligent risk taking
  - Learning and practicing
- Data-driven
- Safe
- Reflection
- Recognition

Organizational culture is one of the strongest predictors of both IT performance and overall performance of the organization.

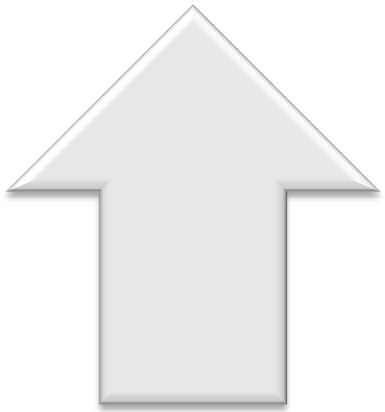
Shifting Thoughts and Behaviors

From	To
• IT focus (inside-out) • Silos • Command and control • Task-oriented • Blame • Reactive • Content • Resistant • Low trust	• Customer focus (outside-in) • Cross-functional teams • Collaborative • Outcome-oriented • Responsibility • Proactive • Courageous • Flexible • High trust

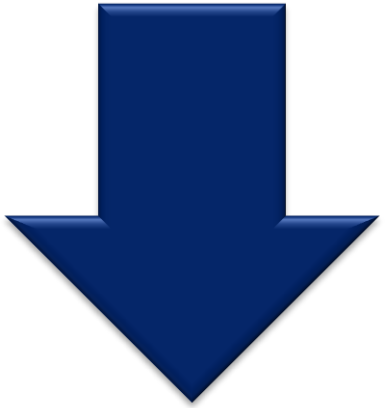
Real culture change takes time. It must be incremental and performed at a realistic pace.

High Trust vs. Low Trust

High Trust

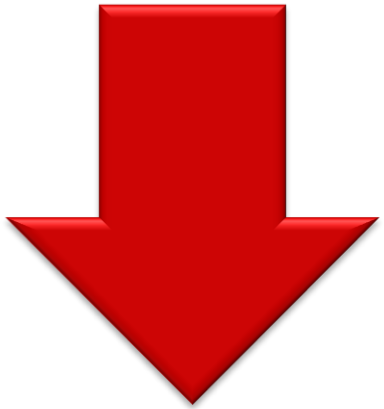


Speed

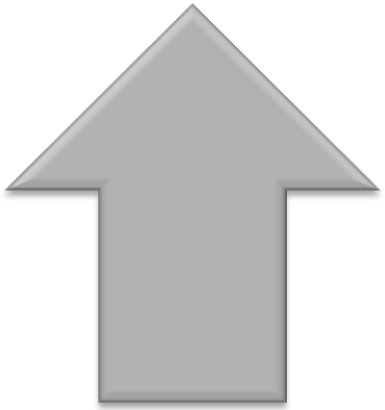


Cost

Low Trust



Speed



Cost

Culture and the Flow of Information

Pathological (Power-oriented)	Bureaucratic (Rule-oriented)	Generative (Performance-oriented)
Information is hidden	Information may be ignored	Information is actively sought
Messengers are ‘shot’	Messengers are isolated	Messengers are trained
Responsibilities are shirked	Responsibility is compartmentalized	Responsibilities are shared
Bridging is discouraged	Bridging is allowed but discouraged	Bridging is rewarded
Failure is covered up	Organization is just and merciful	Failure causes enquiry
Novelty is crushed	Novelty creates problems	Novelty is implemented

High-trust organizations encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas.

Source: Westrum, *A Typology of Organizational Cultures*

MANAGING CULTURE CHANGE

"People don't resist change.
They resist being changed."

Peter Senge

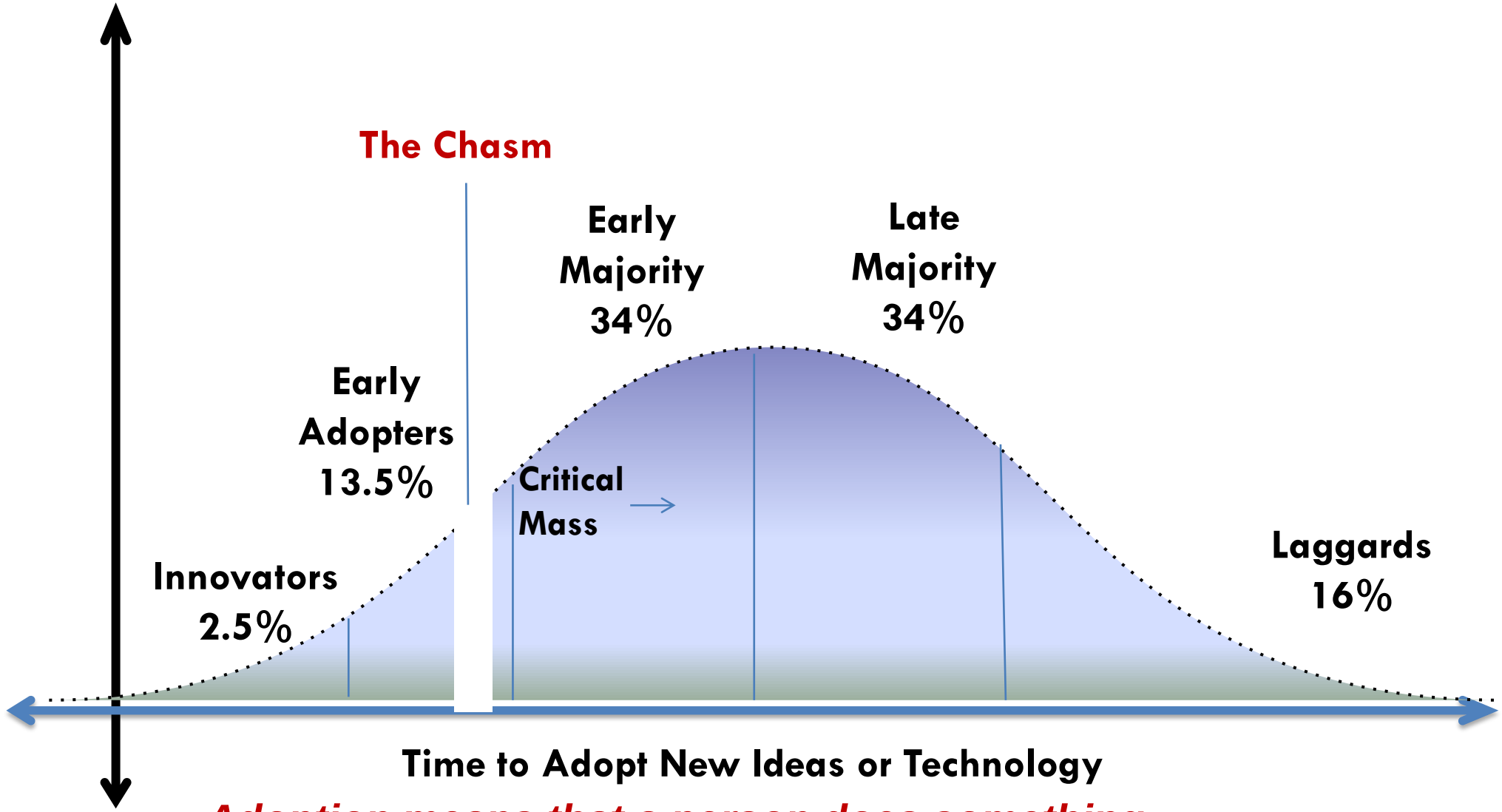
Culture Change is Never Easy

- You can't change people; they can only change themselves
- Change almost always takes longer and costs more than expected
- Stakeholder involvement is critical
- People who participate in what and how to change decisions are far more likely to accept change

People typically don't resist their own ideas.

Source: *The ITSM Process Design Guide*

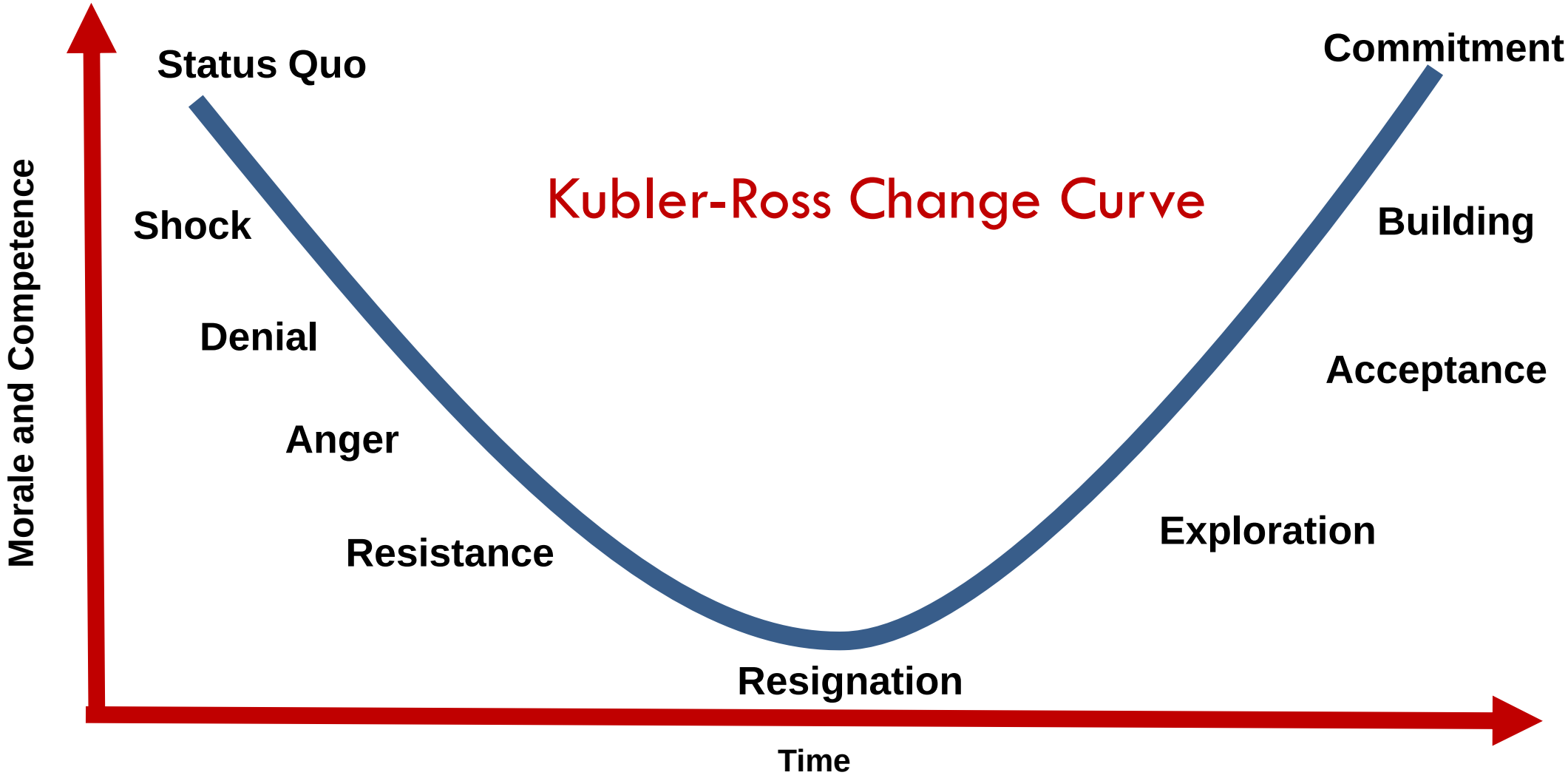
People Adapt to Change at Different Paces



Adoption means that a person does something differently than before.

Source: Rogers. *Diffusion of Innovations*

The Stages of Change Acceptance



Adapted from Kubler-Ross Model

Communication is Critical

- A DevOps culture requires timely and effective communication
- Shared tools facilitate timely and meaningful communication
  - Chat platforms
  - Task managers
  - Social tools
  - Alert management tools
  - Knowledge sharing platforms

Encourage Collaborative Relationships

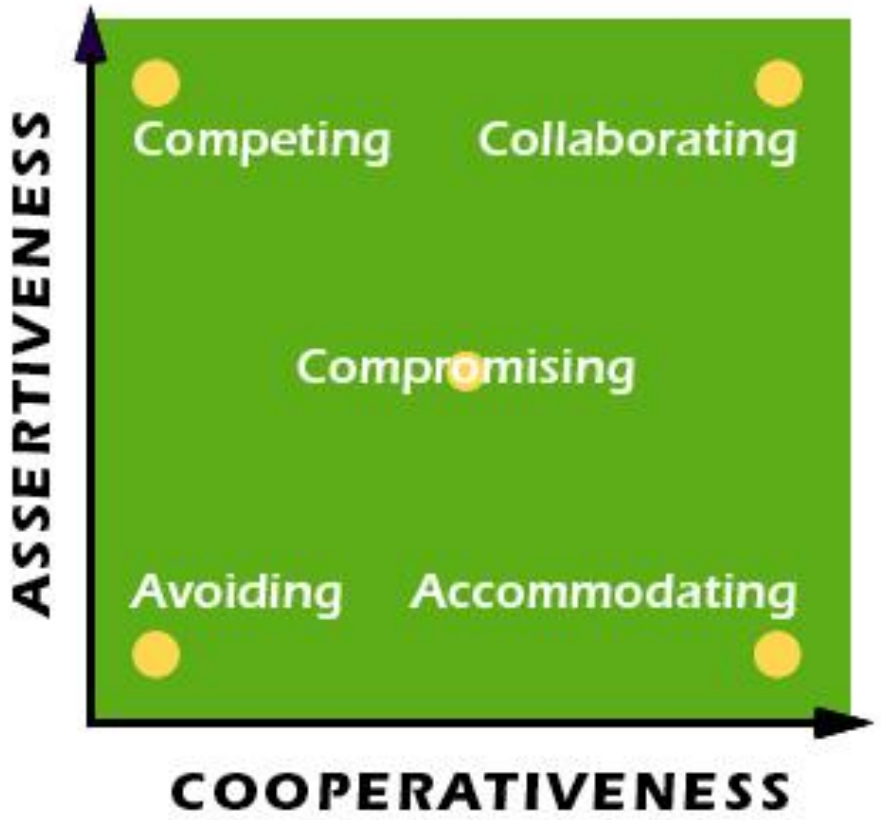
Collaboration involves people jointly working with others towards a common goal. In a collaborative environment, each person's contribution is valued.

- Collaboration
  - Is voluntary (ideally)
  - Involves sharing
    - Responsibility for outcomes
    - Resources
  - Requires cooperation, respect and trust
- Requires participation
  - Providing feedback
  - Identifying and solving problems
  - Learning and sharing knowledge and expertise
  - Sharing and even swapping responsibilities
  - Making and keeping realistic commitments

What's the difference between collaboration and communication?

Expect Some Conflict

Because no two individuals have exactly the same expectations and desires, conflict is a natural part of our interactions with others.



Source: www.kilmanndiagnostics.com

The Thomas-Kilmann Conflict Inventory (TKI) measures a person's behavioral choices under certain conflict situations.

Thomas-Kilmann Conflict Modes

Conflict Mode	Approach	Result
Competing	Assertive and Uncooperative	Win/Lose
Collaborating	Assertive and Cooperative	Win/Win
Compromising	Partially Assertive and Cooperative	Each Wins and Loses
Avoiding	Unassertive and Uncooperative	Lose/Lose
Accommodating	Unassertive and Cooperative	Lose/Win

The key to conflict management is knowing how and when to use each mode.

Avoid Change Fatigue

Change fatigue is a general sense of apathy or passive resignation towards organizational changes by individuals or teams.

- View resistance to change as normal
  - Listen, empathize
- Communicate the big picture
  - Explain the reason for *this* change
  - Show how changes are connected
  - Tie changes to business strategies and goals
- Ensure each change initiative has an intended outcome
- Empower people to contribute
- Celebrate (even if only small) successes
- Create visible feedback and improvement loops

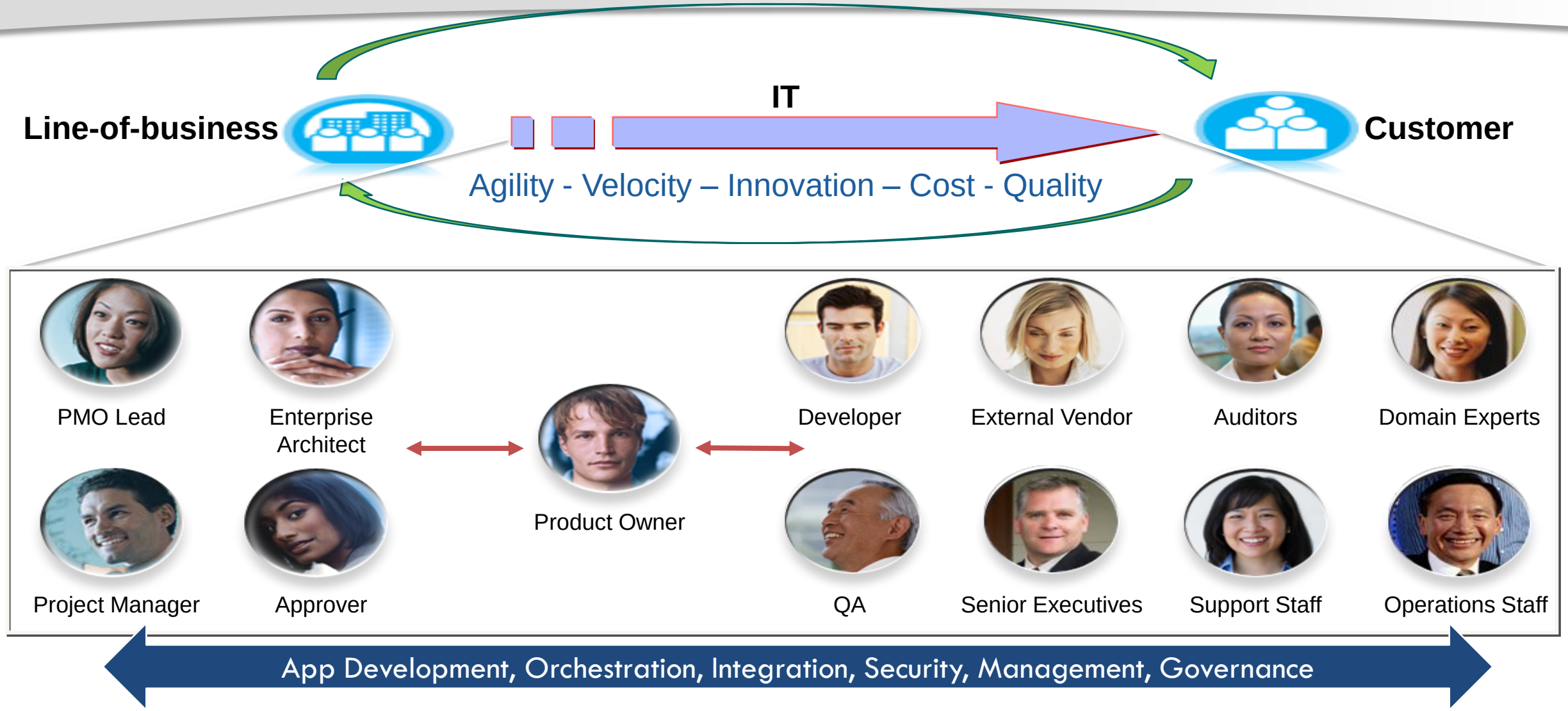
The amount of change fatigue that people experience is directly impacted by the way change is managed.

Empower New Behaviors

- Improve communication and collaboration practices and shared tools
- Create a common vocabulary
- Job shadowing
- Cross-skilling
- Immersion experiences
- Team building
- Communities of practice
- Internal DevOps Days
- Game days (hackathons)
- Simulations
- Social-media style idea sharing and problem solving

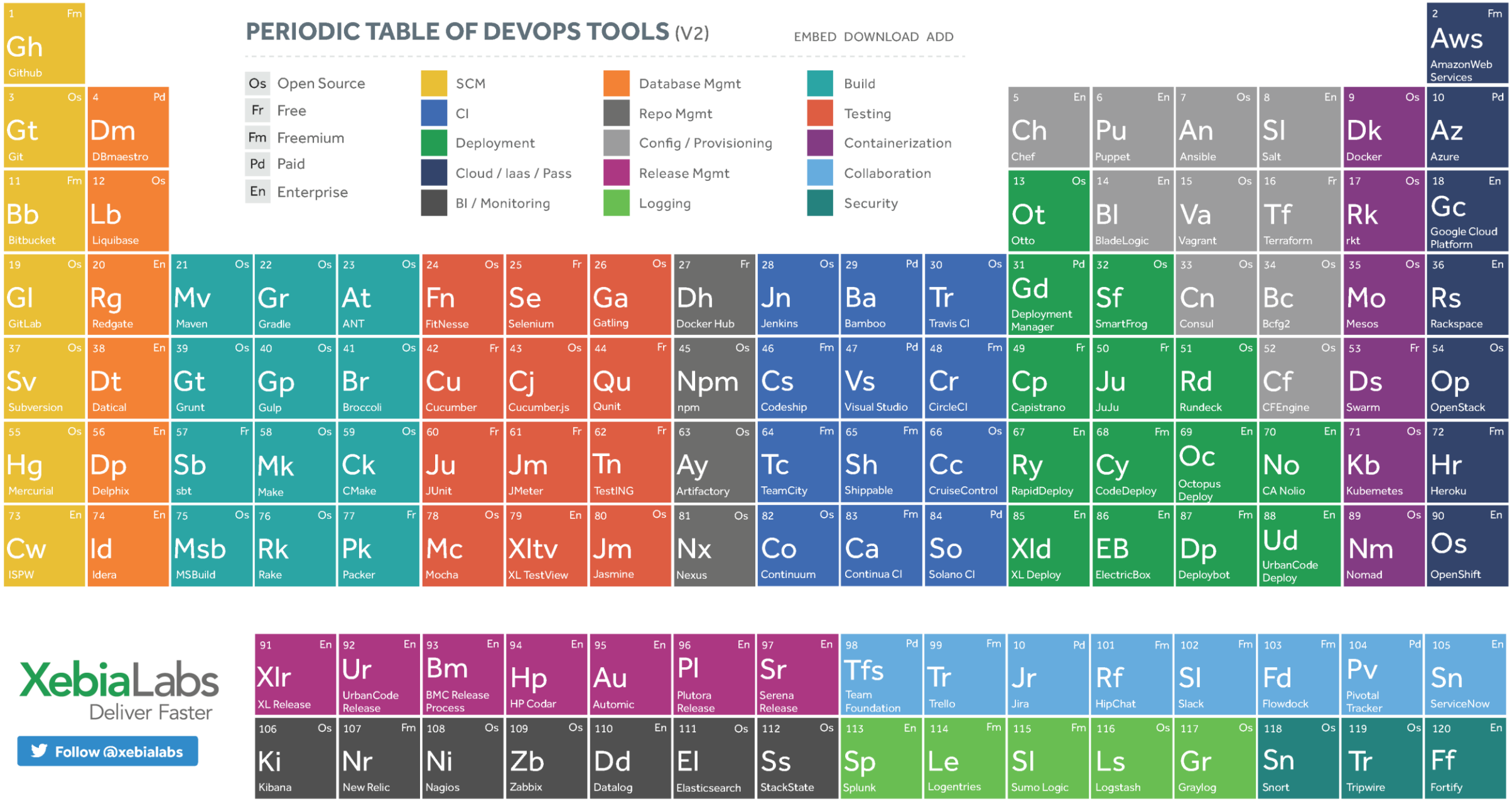
Sharing between peers, organizations and industries is a crucial factor in the growth and acceptance of DevOps.

Become One IT, One Team with Collaboration, Communication, Trust



Source: Sanjeev Sharma, IBM

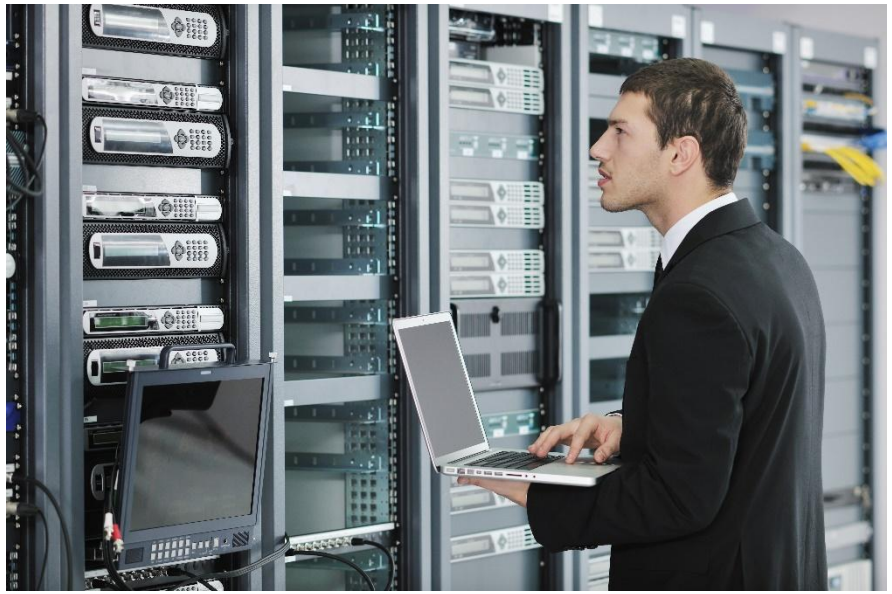
AUTOMATION



https://xebialabs.com/periodic-table-of-devops-tools

Automation Enablers

Automation alone cannot give you DevOps – but you cannot succeed without it!



- Continuous integration and continuous delivery
- Configuration management
- Build, testing and deployment processes
- On-demand creation of development, test, staging and production environments
- Treating infrastructure as code
- Monitoring and dashboards
- Experimentation
- Ongoing operations and support

Automation Benefits

Automation supports

- Faster lead times
- More frequent releases
- Less turbulent releases
- Fewer errors
- Higher quality
- Improved security and risk mitigation
- Faster recovery
- Business and customer satisfaction

Automation gives rote tasks to computers and allows people to

- Weigh evidence
- Solve problems
- Make decisions based on feedback
- Use their skills, experience and judgment

"Your tools alone will not make you successful."
Patrick Debois

Important Terms

- **Artifact**
  - Any element in a software development project including documentation, test plans, images, data files and executable modules
- **Application Programming Interface (API)**
  - A set of protocols used to create applications for a specific OS or as an interface between modules or applications
- **Microservices:**
  - A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system
- **Operating System (OS) Virtualization**
  - A method for splitting a server into multiple partitions called "containers" or "virtual environments" in order to prevent applications from interfering with each other
- **Containers**
  - An approach to software development that packages pieces of code so that they can quickly be plugged in and run on a Linux (or Linux-like) OS
- **Open source**
  - Software that is distributed with its source code so that end user organizations and vendors can modify it for their own purposes

DevOps Automation Practices

Avoid tools that enforce silos!

- Tool chain (vs. a single-vendor solution)
- Shared tools
- Self-service
- Architecting software in a way that enables
  - Test automation
  - Monitoring
- Experimentation

A tool chain philosophy involves using an integrated set of complimentary task specific tools to automate end-to-end delivery and deployment processes.

Communication and Collaboration Can Be Automated Too

Innovative tools and platforms facilitate and expedite communication and collaboration across the Dev and Ops spectrum.

How to	Tools
• Issue alerts and alarms • Improve response • Provide at a glance status updates • Improve workflow • Improve information flow • Enable virtual collaboration • Enable cross-functional, cross-skilling and job sharing	• Communication platforms • Dashboards • Kanban boards • Group chat rooms (ChatOps) • Workflow and project management tools • Document sharing • Wikis and knowledge management systems • ITSM tools • Socials tools • Shared backlogs

First Steps to Improving DevOps Automation

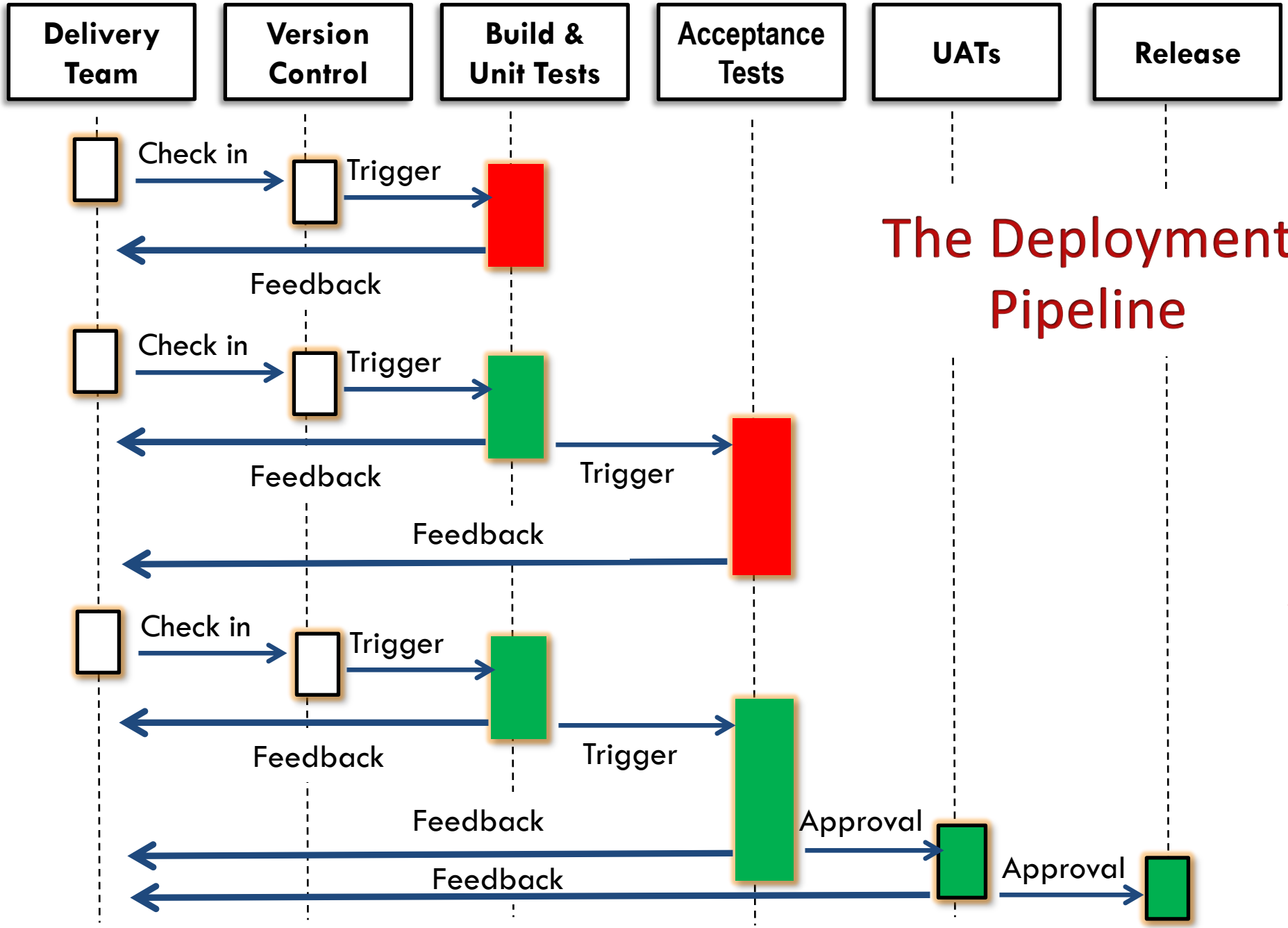
- Architect before automating
- Assess your existing tools and automation capabilities
- Simplify first – don't automate bad processes
- Identify critical gaps
- Seek vendors who can meet *your* requirements
- Automate high value, repetitive and error-prone work
- Optimize workflow bottlenecks and communication
- Improve automated monitoring and notification practices
- Expect this to be an iterative process - your toolchain will evolve over time

Do not underestimate the effort and cost of building toolchains from open source applications. Open source is not necessarily free. It means that you can modify the source to fit your needs.

DEVOPS TOOLCHAINS

"One way to enable market-oriented outcomes is for Operations to create a set of centralized platforms and tooling services that any Dev team can use to become more productive…a platform that provides a shared version control repository with pre-blessed security libraries, a deployment pipeline that automatically runs code quality and security scanning tools, which deploys our applications into known, good environments that already have production monitoring tools installed on them."

The DevOps Handbook



The Deployment Pipeline

The deployment pipeline is an automated process for managing all changes, from check-in to release. Toolchains span silos and automate the deployment pipeline.

Source: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*

DevOps Toolchains

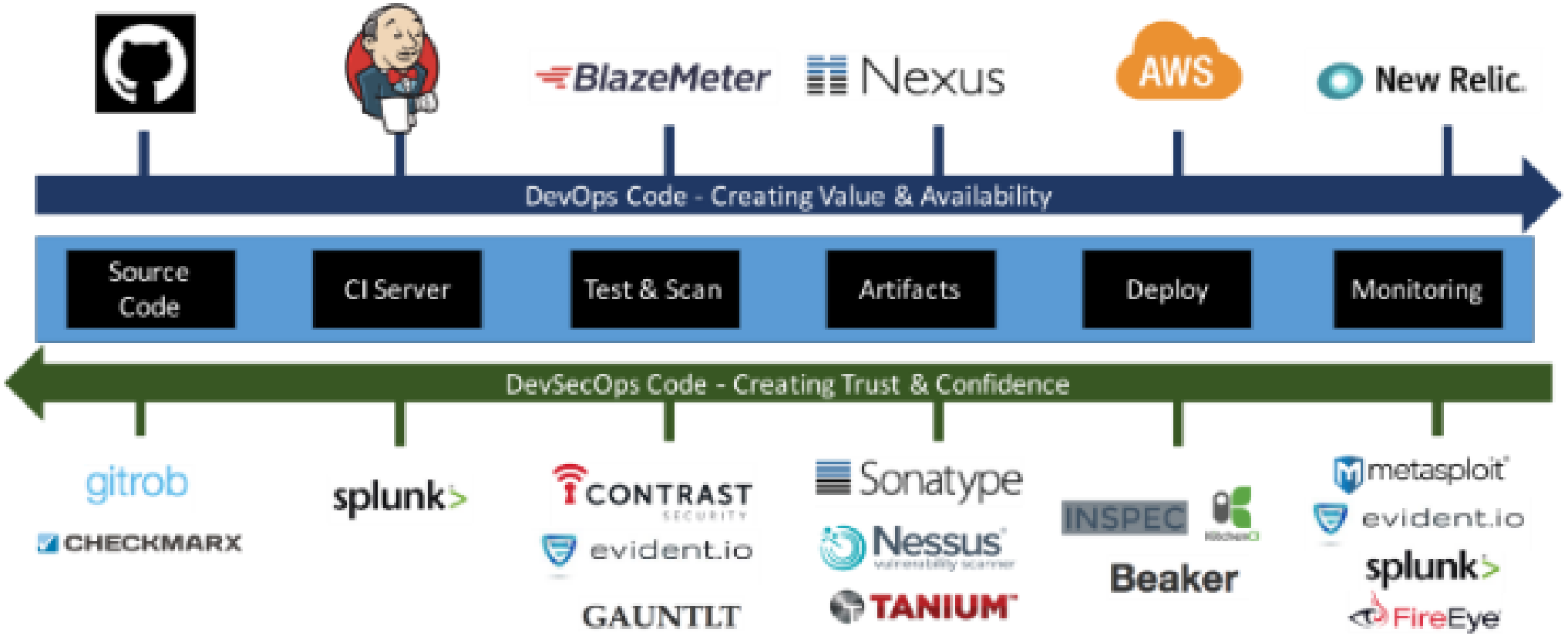
The DevOps toolchain is composed of the tools needed to support a DevOps continuous integration, continuous deployment, and continuous release and operations initiative.
Gartner

- Toolchains automate tasks in the deployment pipeline
- Each element of the toolchain serves a specific purpose (microservices)
- Applications within the toolchains are connected via APIs
- They do not have to be homogenous or from a single vendor
- Toolchains are usually built around open and closed source ecosystems
- Requires an architectural design to ensure interoperability and consistency

How should DevOps toolchains interface to operational tools such as monitoring or support applications?

Sample DevOps Toolchain

There are many established open- and closed-source DevOps-enabled tools with vibrant ecosystems.



How these tools are adapted and integrated into your deployment pipeline will determine their value.

Elements in a DevOps Toolchain

- The deployment pipeline breaks the software delivery lifecycle into logical stages
- Each stage provides
  - The opportunity to verify the quality of new features from a different angle
  - The team with fast feedback
  - Visibility into the flow of changes
- DevOps toolchains provide the capabilities needed to automate and expedite each stage

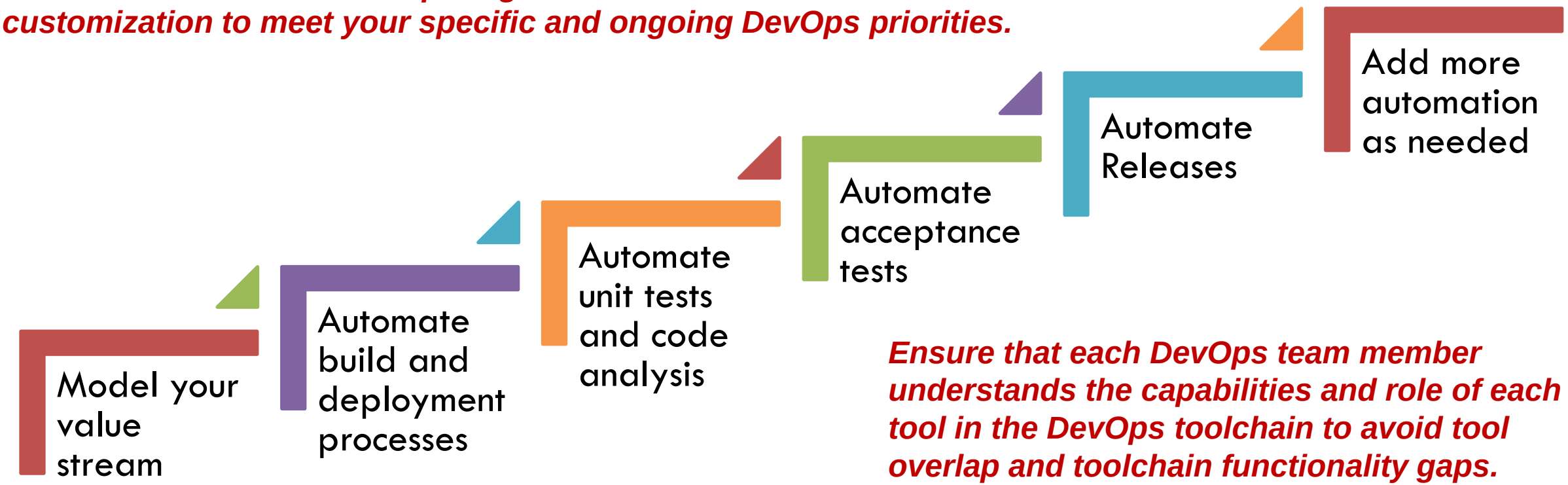
Typical Toolchain Elements

- Requirements management
- Orchestration and visualization
- Version control management
- Continuous integration and builds
- Artifact management
- Containers and OS virtualization
- Test and environment automation
- Server configuration and deployment
- System configuration management
- Alerts and alarms
- Monitoring

The DevOps Ecosystem Has Many Tool Options

Build Your Deployment Pipeline Toolchain Gradually

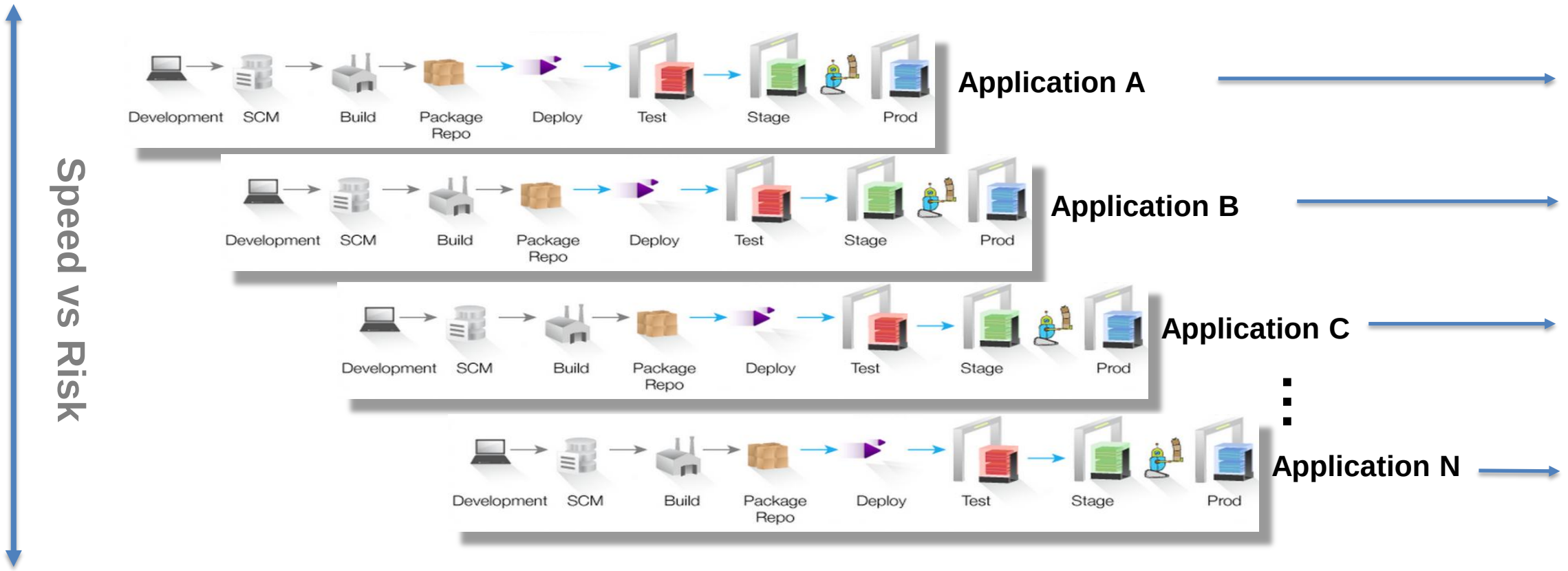
Do not create a definitive toolchain that applies to all DevOps projects. The toolchain is a foundation requiring continuous innovation and customization to meet your specific and ongoing DevOps priorities.



Ensure that each DevOps team member understands the capabilities and role of each tool in the DevOps toolchain to avoid tool overlap and toolchain functionality gaps.

Source: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble and Dave Farley
http://www.informit.com/articles/article.aspx?p=1621865&seqNum=8

Multiple Business Applications Require Multiple Deployment Pipelines and Toolchains



Avoid creating more pipeline silos by taking an enterprise architecture approach.

Source: Sanjeev Sharma, IBM

Lean as a DevOps Value

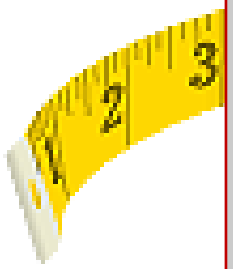
"I believe that most of the DevOps patterns are the emergent properties that arise when you apply the techniques like Lean, the Toyota Production System, the Theory of Constraints and so forth to the IT value stream."

Gene Kim

MEASUREMENT

Measuring Success

Speed	Quality	Stability	Culture
Change lead and cycle times	Change failure rate	Mean time to detect incidents (MTTD)	Retention and loyalty
Deployment frequency	Deployment success rates	Mean time to recover (MTTR) – Component	Engagement and moral
Deployment speed	Incidents and defects	Mean time to restore service (MTRS) – Service	Knowledge sharing



Showing proof that DevOps practices benefit the organization requires examining factors that influence overall IT performance.

Adapted from Splunk 2016

SHARING

DevOps Encourages a Sharing Culture

- Immersion opportunities are becoming more available in an effort to provide DevOps teams access to subject matter coaches on topics such as CI, CD, Lean and design methods
  - Dojos (Internal to Target)
  - Garages (IBM)
  - Lofts (Amazon)
  - More to come

- DevOps simulations and gamifications are also becoming more available

Games, hackathons, common workspaces, simulations and other innovations are helping to encourage the sharing of tools, knowledge, discoveries and lessons learned.

http://target.github.io/devops/the-dojo

Internal DevOps Days

- Some organizations are replicating the DevOps Days model as internal events
- DevOps Day events give teams and individuals an opportunity to learn, share, discuss, engage and provide input and feedback

While most effective in a physical location, internal DevOps Days can be conducted in a virtual environment.

The format can include

- Traditional 30 minute presentations from internal and external resources
- Ignite (5 minute rapid-fire) topic-specific sessions
- Openspace break-out discussions on suggested topics

DEVOPS IN THE ENTERPRISE

ROLES

Addressing the DevOps Skills Gap

- The demand for DevOps resources is making it difficult for organizations to attract and retain talent
- The breakneck pace at which technologies are evolving is making it difficult for individuals to maintain a current skill set
- Ensuring individuals have the needed soft skills and are a good cultural fit adds to the hiring challenge

Strategies

- Training and certification
- Immersion/coaching programs
- Restructuring pay and corporate culture
- Supplement internal teams with outsourced talent
- Recruiting bonuses

Todays CIOs are looking for workers who can shift gears and adapt to changing technology.

Skills and Characteristics of a DevOps Professional

Skills

- Business – Knowledge of business priorities and processes
- Technical – Specialist with broad generalist knowledge (T-shaped) – experience or at least an interest in writing code
- Soft – Communication, collaboration, team work
- Self-management – initiative, time and stress management, self-motivation, focus

Characteristics

- Adaptable
- Customer-focused
- Craftsmen
- Curious
- Data-driven
- Engaged
- Empathetic
- Transparent

Generalist technical knowledge includes an understanding of DevOps practices, modern software engineering practices and modern architectures.

DevOps Roles

The following roles are emerging as critical to DevOps' success

- DevOps evangelist or leader
- Software engineers, developers and testers
- Release manager
- Automation/continuous delivery architect
- Build engineer
- Security engineer
- Quality assurance (QA)/Experience assurance (XA)
- DevOps operations engineer
- IT Support
- Agile Service Manager®
- Agile Process Owner ®

What other roles do you think should be involved?

Source: http://techbeacon.com/7-devops-roles-you-need-succeed

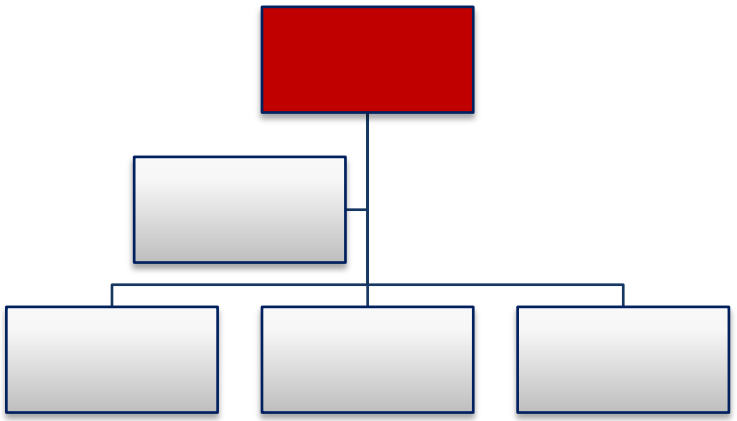
153

What is a DevOps Engineer?

- There is currently no 'industry recognized' job description or formal career track for a DevOps Engineer
- As with the concept of a DevOps team, the title has its pros and cons
- General characteristics include someone who
  - Wants to contribute his or her technical talent to business and process improvement initiatives
  - Is comfortable collaborating with others
  - Wants to be in a workplace that promotes a shared culture

ORGANIZATIONAL CONSIDERATIONS

DevOps Organizational Structures



There is debate about the pros and cons of DevOps teams.

Some organizations are

- Assigning Ops liaisons to Dev/Scrum teams
- Creating cross-functional product (vs. project) teams
- Adopting matrix or market-oriented (vs. function-oriented) structures
- Creating shared Ops services that support multiple Dev teams

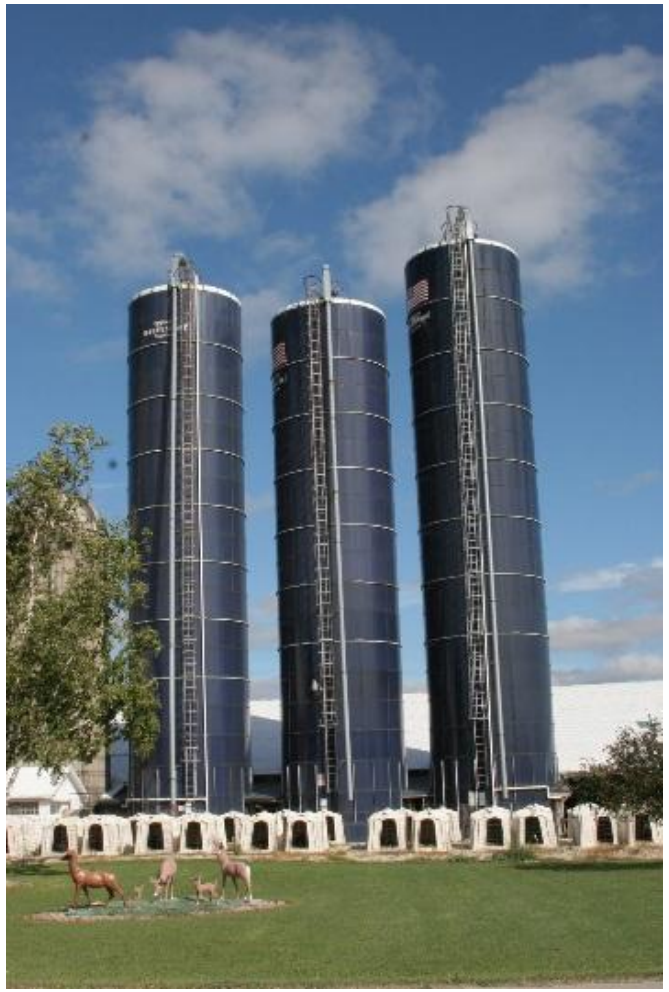
DevOps Teams (1)

The creation of DevOps departments or teams is a growing trend. 22% of 2016 State of DevOps survey respondents indicated they were part of a DevOps department.

- DevOps teams
  - Expand upon the concept of an Agile or Scrum team
  - Embed Dev and Ops skills into a single holistic group
  - May be temporary or dedicated to a specific product
  - May be cross-functional 'tiger teams' for short-term projects
  - May evolve to provide shared services
  - Have shared accountabilities
  - Should adhere to the defined standards for development, automation, risk and compliance that applies to all DevOps teams

There is no 'ideal' structure for a DevOps team.

DevOps Teams (2)



- Downsides of dedicated DevOps Teams
  - Less engagement across the IT value stream
  - Risk of being another silo
  - Dev and Ops wash their hands of accountability
  - DevOps activities become someone else's problem

Regardless of structure, a DevOps team should be flat, with continuous engagement and the right balance of people, practices and automation skills.

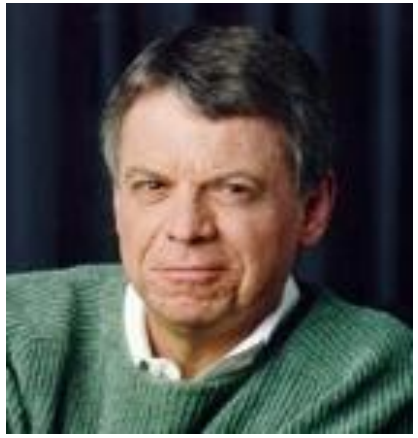
GETTING STARTED

"It's a journey, not a silver bullet, and leaders need to avoid getting caught in analysis paralysis. Start making the changes, get the wins and let the organization evolve."

Melissa Sargeant

Eight Steps to Transforming Your Organization

- Establish a sense of urgency
- Create a powerful guiding coalition
- Develop a vision and strategy
- Communicate that vision
- Empower broad-based action
- Create short-term wins
- Consolidate gains and produce more change
- Anchor new approaches in the culture



Source: John P. Kotter

Start Where You Are

"DevOps is not your why,
not your co-workers' why,
certainly not your business' why."
Damon Edwards

- Get clear on the business opportunity – the 'Why?'
- Get the right people together
- Get everyone on the same page
- Invest in training and skills development
- Build capabilities that lead to lasting change
- Focus on critical behaviors
- Experiment and learn
- Consolidate gains and produce more change
- Avoid inertia

Learn by Doing

- Create a pilot where you can maximize the probability of success
- It should be small enough where
  - Success is apparent and understood
  - Consequences of failure aren't so large that a mistake could shut down the entire initiative
- It should be large enough that
  - You can show proof of improvement
  - You earn the right to make future improvements

Consolidate Gains and Produce More Change

- Communicate successes, failures and lessons learned
- Document and make available reusable artifacts and measurements
- Expand your cycles of improvement
- Continuously invest in education
- Introduce advanced tools and techniques as needed

Anchor the Results

- Prove that the new way of doing things is better
- Reinforce new behaviors with incentives and rewards
- Be prepared to lose some people along the way
- Reinforce the new culture with every new employee

"Change sticks when it becomes 'the way we do things around here'."

John. P. Kotter

CHALLENGES, RISKS AND CRITICAL SUCCESS FACTORS

Critical Success Factors

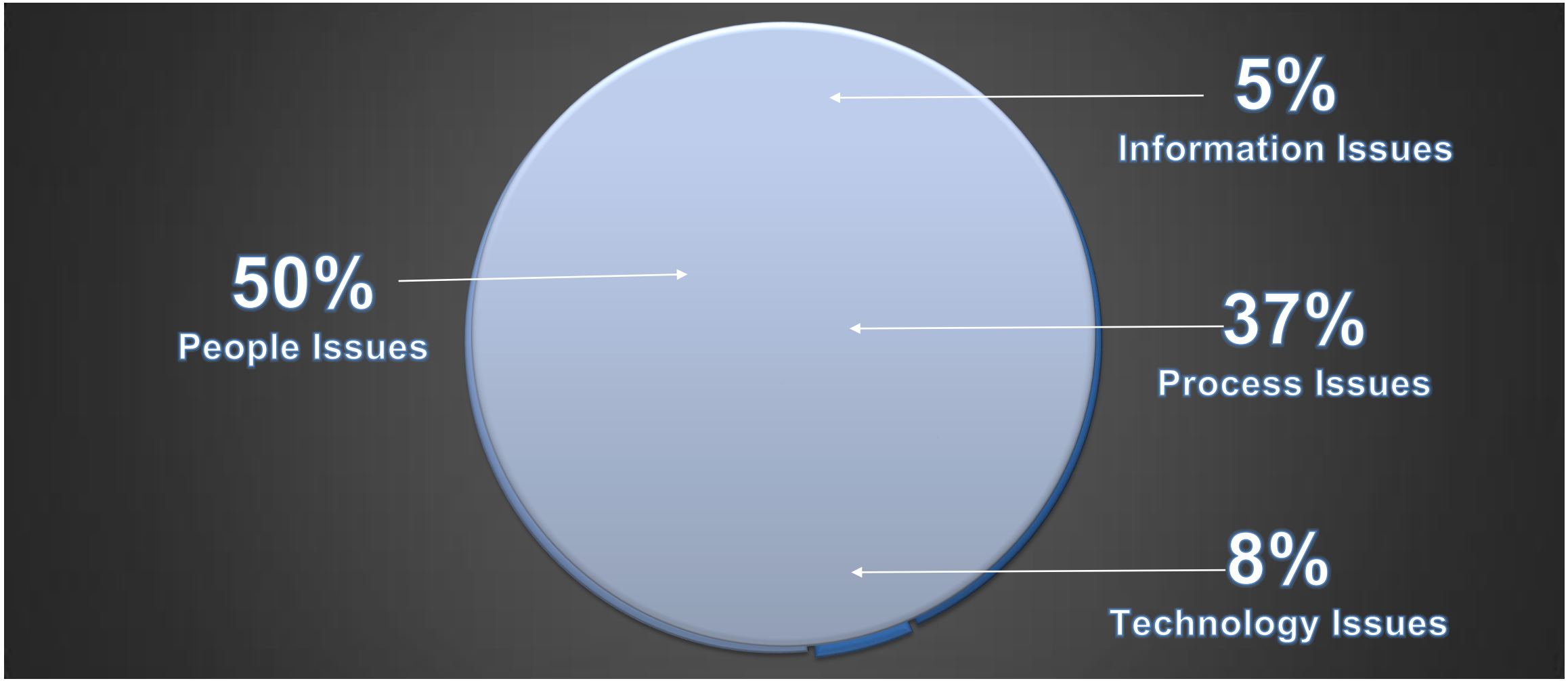
- Management commitment to culture change
- Creation of a collaborative, learning culture
- Training and continuous skills improvement
- Common values and vocabulary
- Systems engineering that spans Dev and Ops
- Meaningful metrics
- A balance between automation and human interaction
- Application of agile and lean methods
- Open and frequent communication

Challenges and Risks

- Lack of commitment or clarity
- Transforming a “them” and “us” culture
- Blending teams that are geographically dispersed, unfamiliar with each other and may include suppliers
- Lack of education, training and skill
- Immature service management processes
- Inadequate technologies
- Poor communication

Overcoming these challenges will require organizational change.

Which of these is the biggest challenge for your organization's expansion of the use of DevOps?



Source: Gartner

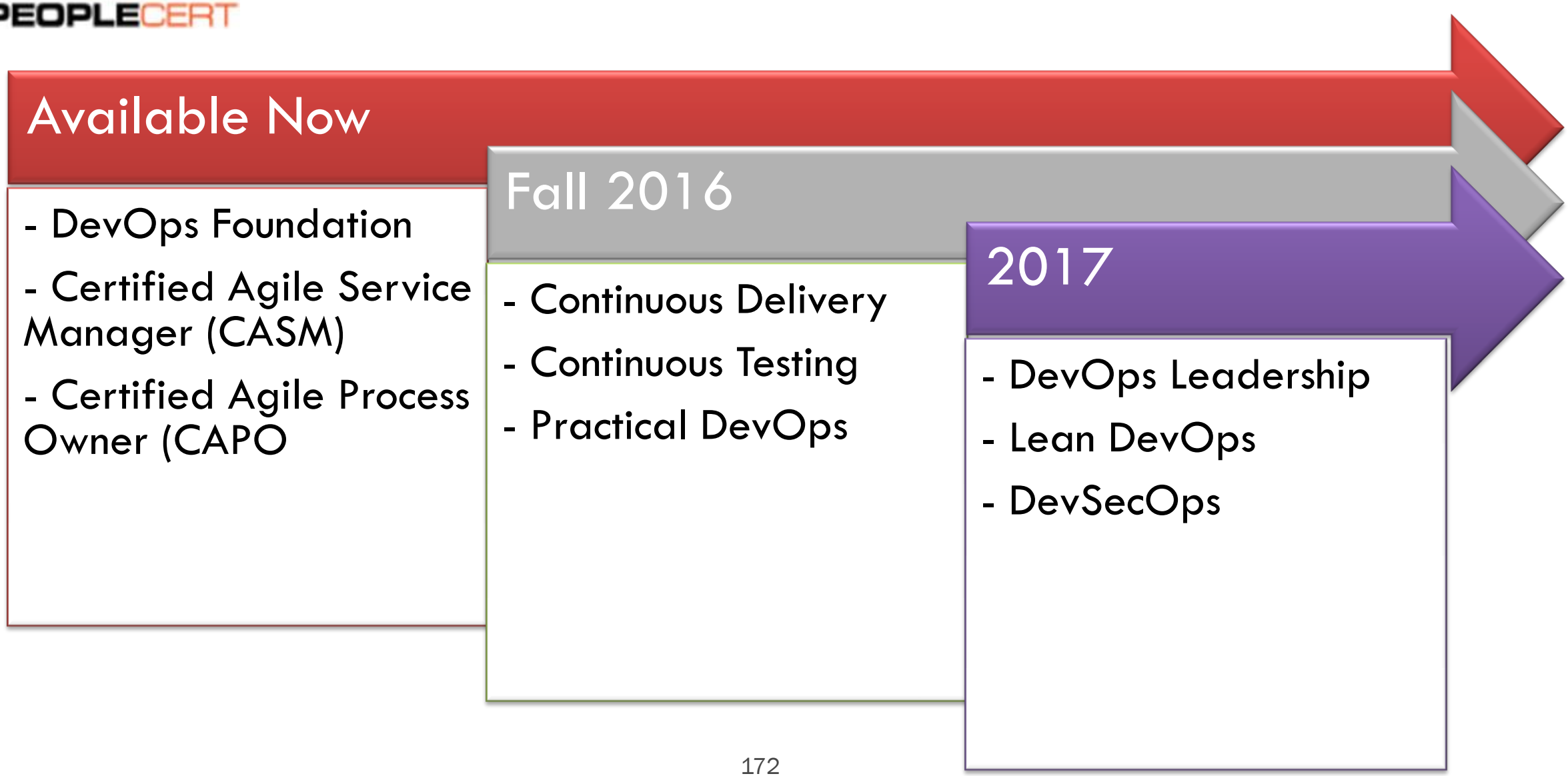
Summary

- DevOps enables companies to deliver better software faster and more reliably by…
  - Improving communication, collaboration and the integration of processes and tools across the IT value stream
  - Automating the process of software delivery and infrastructure changes
  - Leveraging agile, lean, ITSM and evolving DevOps practices

“DevOps is not only possible, it is necessary in the new world of business technology.”

Forrester Research

Continue Your Knowledge Journey - Training



172

Continue Your Knowledge Journey - Websites

- www.devopsinstitute.com
- www.devops.com
- www.devopsconnect.com
- www.devopsdays.org
- www.devopsenterprise.io
- www.itrevolution.com
- www.informit.com

Links to additional articles, blogs and videos are available in a separate document.

Continue Your Knowledge Journey - Publications

- 2013-2016 State of DevOps Report. Puppet Labs, IT Revolution Press
- The DevOps Handbook. G. Kim, et al., IT Revolution Press, To Be Published, 2016
- Leading the Transformation. G. Gruver and T. Mouser. IT Revolution Press, 2015
- The Phoenix Project. G. Kim, et al., IT Revolution Press, 2013
- Continuous Delivery. J. Humble, et al. Addison-Wesley Professional, 2010
- Lean Enterprise: Adopting Continuous Delivery, DevOps, and Lean Startup at Scale. J. Humble, et al, O'Reilly Media, 2014
- Lean IT: Enabling and Sustaining Your Lean Transformation. S. Bell and M Orzen. Productivity Press, 2010
- The Lean IT Field Guide, M. Orzen, Productivity Press, 2015

DevOps best practices will continue to evolve through a collective body of knowledge (cBok).

If I Could Wave a Magic Wand, Everyone Will…

- Be energized about how practitioners can contribute in this organizational journey
- Leave with some concrete steps to get some great outcomes
- Help create a team that starts putting DevOps practices into place

Source: Gene Kim – *How Can We Better Sell DevOps?*