

DEVOPS FOUNDATION

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Tell Us a Little About Yourself

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

HI my name is:
<i>Your Name Here</i>

DevOps Foundation Course Goals

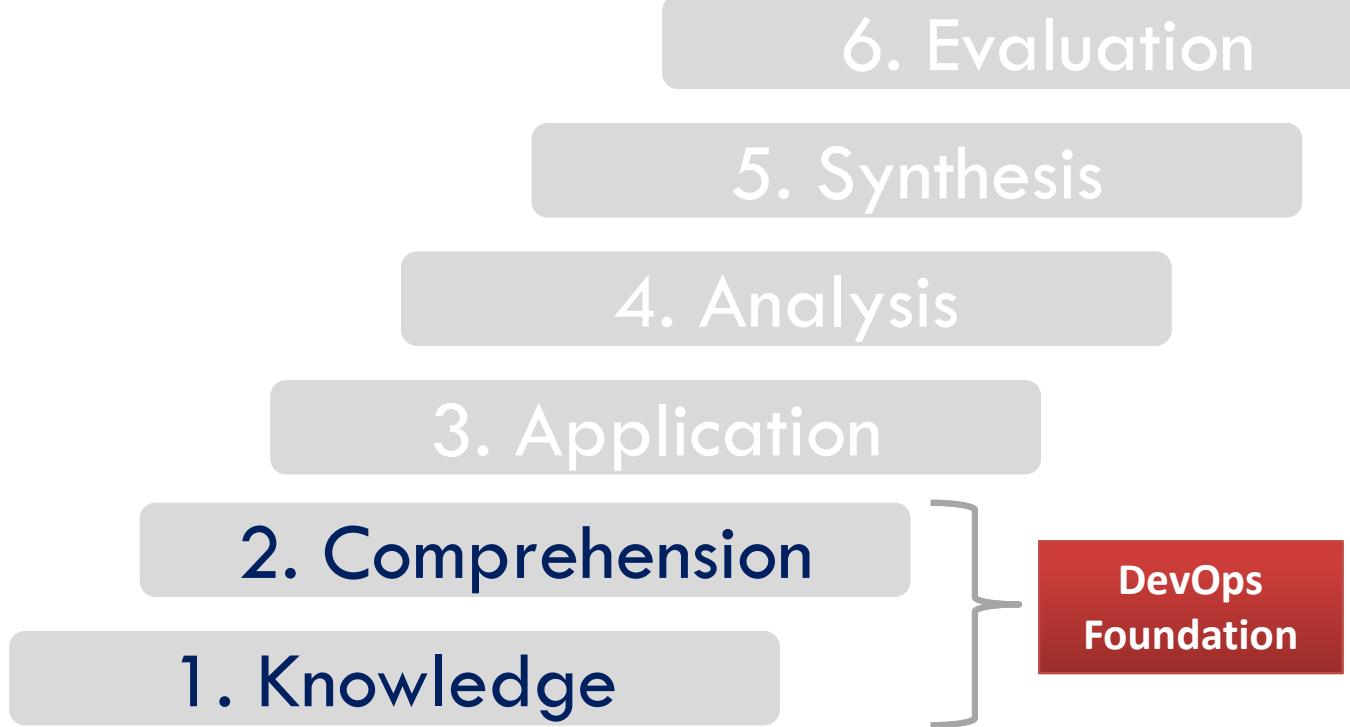
- Learn about DevOps
- Understand its core vocabulary, principles and practices
- Hear and share real life scenarios
- Earn credits – PMPs earn **16** PDUs
- Create a personal action plan
- Have fun!
- Pass the DevOps Foundation Exam
 - 40 multiple choice questions
 - 60 minutes
 - 75 minutes – If English is not native tongue
 - 65% is passing
 - Accredited by the DevOps Institute



© DevOps Institute

Copyrighted Material. Do Not Reproduce.

About Bloom's Taxonomy



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

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Foundation Course Agenda

Day 1	Day 2
<ul style="list-style-type: none">• Class Introductions• Course Introduction	<ul style="list-style-type: none">• Morning Review
<ul style="list-style-type: none">• Why DevOps?• What is DevOps?• DevOps Principles	<ul style="list-style-type: none">• DevOps Organizational Considerations• DevOps Practices
Lunch (One hour)	
<ul style="list-style-type: none">• DevOps and Other Frameworks• DevOps Culture	<ul style="list-style-type: none">• DevOps Automation Practices and Tool Categories• Adopting a DevOps Culture
Homework	Exam Review

Back of Book Alerts

- Refer to the Sample Documents and Templates section of this manual
- Provide
 - Larger views of complex figures
 - Alternative or expanded views of figures
 - Sample documents and templates



The material covered in the Sample Documents and Templates section is not examinable unless stated otherwise.

WHY DEVOPS? THE BUSINESS PERSPECTIVE

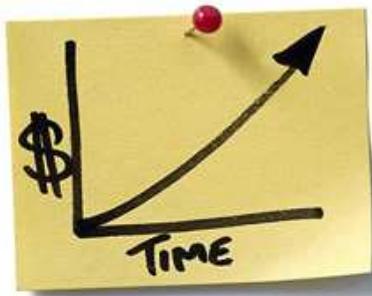
© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Our Cadence is Off

Historically...

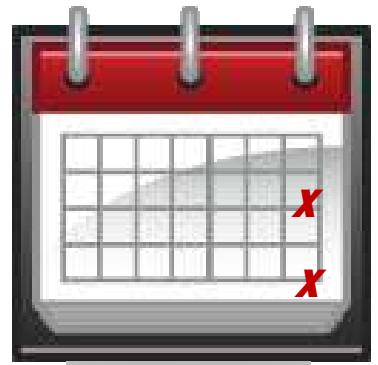
The Business



Dev



Ops



Innovation



Waterfall Projects



Rigorous Processes



Cadence – the flow or rhythm of events.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Impacts to the Business

- Slow business responsiveness
- Customer dissatisfaction
- Poor customer experience
- Reduced ability to compete
- Wasted resources
- Lost productivity
- High costs and risks

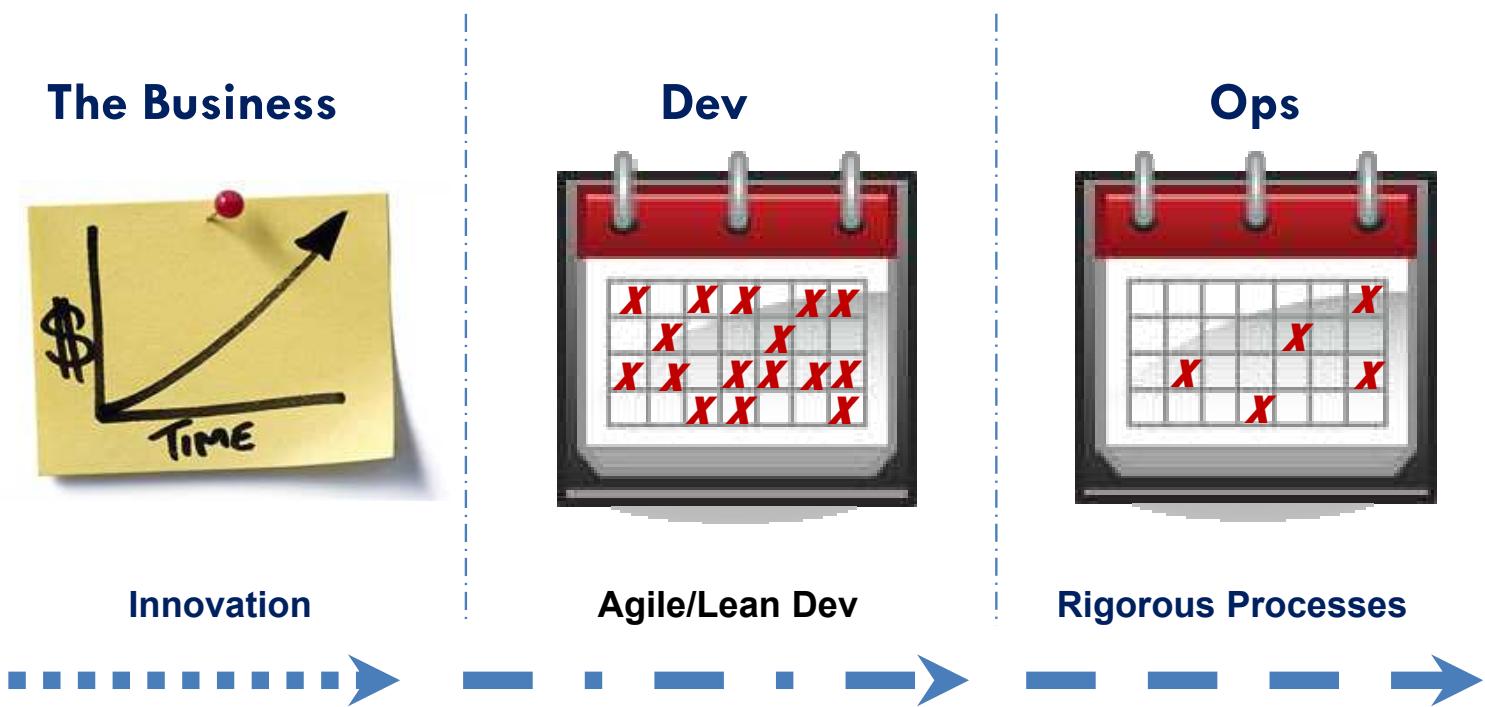


“Someone calling themselves a customer says they want something called service.”

<http://marketingevolution.wordpress.com/>

Better, But Not Quite There

Agile and lean development practices speed things up, but....



© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Improves Our Cadence

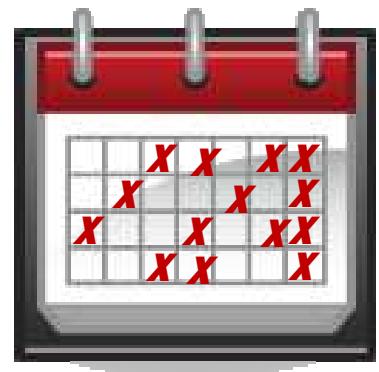
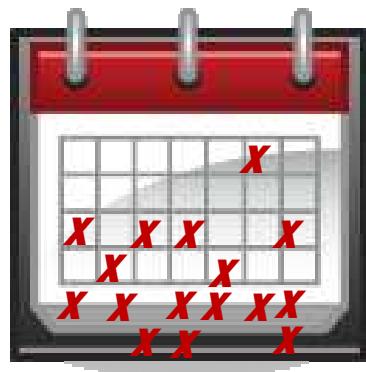
...agile and lean service management practices are also needed.

The Business



**Winning through
Innovation**

Agile/Lean DevOps



Continuous Delivery



Isn't This How It's Supposed to Be?



Speed – Productivity – Quality

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

“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**

WHY DEVOPS? THE IT PERSPECTIVE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

The Wall of Confusion (1)

Dev
wants
change



Ops
wants
stability



The Wall of Confusion (2)



***What does the business want?
All of the above***

The Downward Spiral

Source: Gene Kim, author of *The Phoenix Project*

Operations sees...

- Fragile applications failing
- Difficulties identifying root cause
- Taking too long to restore service
- Extensive firefighting and unplanned work
- Unfinished planned project work
- Frustrated customers leaving
- Market share decreasing
- Missed Wall Street projections
- Empty promises to shareholders

Dev sees...

- More urgent, date-driven projects
- Fragile code going into production
- Releases with “turbulent installs”
- Longer release cycles to amortize “cost of deployments”
- Difficulty diagnosing why deployments are failing
- Senior and constrained OPS resources have less time to fix process problems
- Backlog of infrastructure projects that could fix root cause and reduce costs
- Tension between Dev and Ops

These aren't IT problems...these are business problems!

When IT fails, the business fails.

Copyrighted Material. Do Not Reproduce.

© DevOps Institute

“I'm not sure anyone gets to do more with less these days but we can do more with less effort.”

Gene Kim

DEV AND OPS STEREOTYPES: PERCEPTIONS AND REALITIES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Dev and Ops Perceptions

Exercise:

- What is Ops' Perception of Dev?
- What is Dev's Perception of Ops?



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

Dev and Ops Realities

Dev and Ops are each taking steps to improve their quality and velocity.

- Agile and lean software development practices
- Agile and lean service management practices
- Virtualized and cloud infrastructure from internal and external providers
- Infrastructure as code
- Data center automation and configuration management tools
- Monitoring and self-healing technologies

Unfortunately, Dev and Ops are not working together on these initiatives.

“Without good interpersonal harmony, your business cannot adapt to changes in the competitive landscape.”

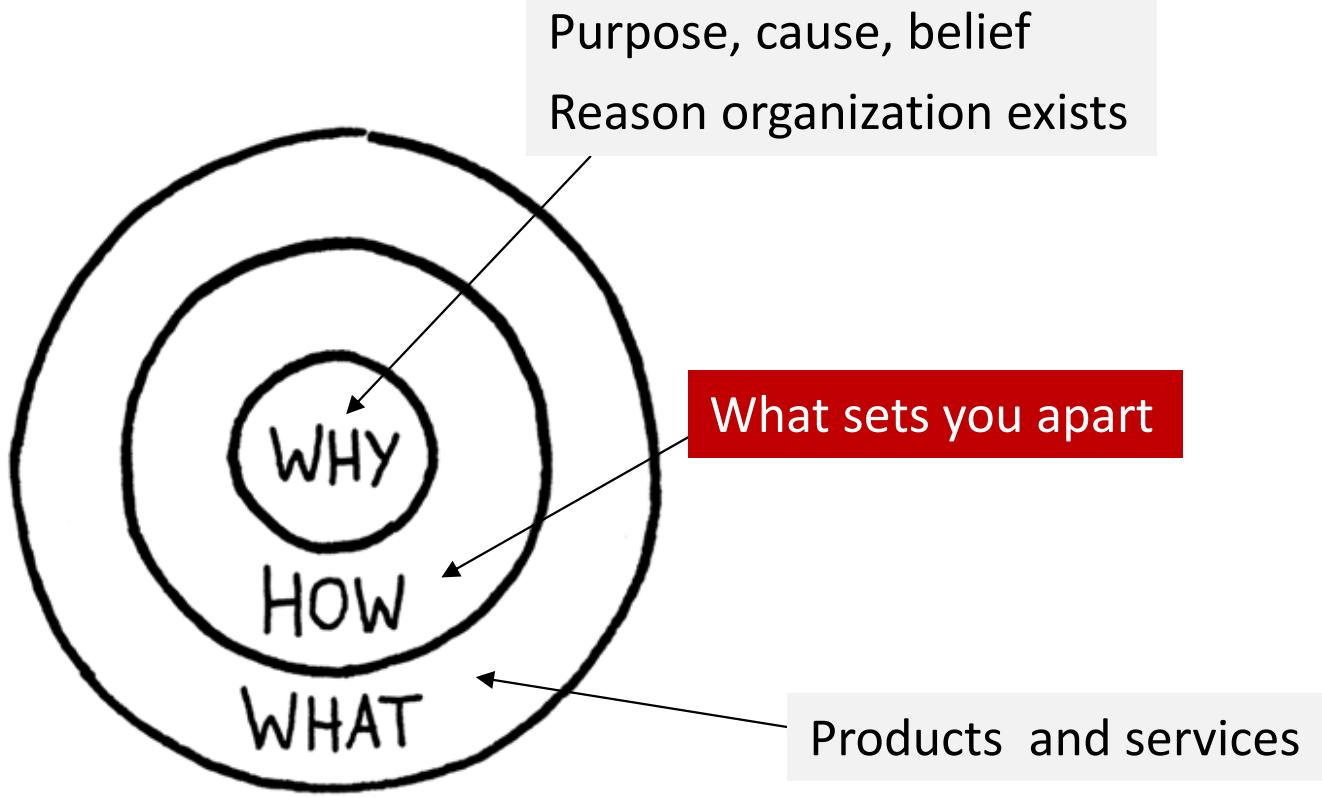
Forrester Research

WHY START WITH WHY?

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

The Golden Circle



© 2013 Simon Sinek, Inc.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

WHAT IS YOUR ORGANIZATION'S WHY?

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

WHAT IS DEVOPS?

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

What is DevOps?

DevOps is a cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.



- Responds to accelerated demand for quality software products and services
- Acknowledges the interdependence of IT functions
- Recognizes the need for cultural improvements
- Supports and leverages agile, lean and IT service management (ITSM) practices
- Encourages the use of automation

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.

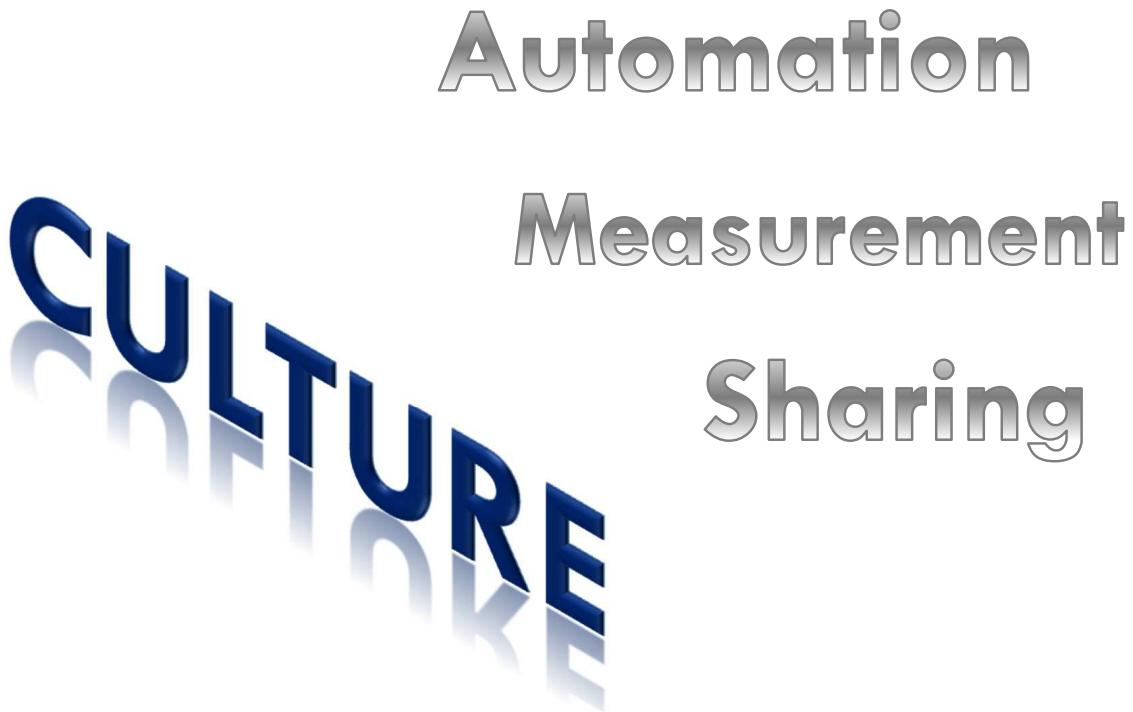
© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Goals

- Better align IT responsiveness and capabilities to business needs
- Produce smaller, more frequent software releases
- Reduce effort and risks associated with software development, transition and operation
- Improve time to market
- Improve quality of code
- Improve quality of software deployments
- Reduce cost of product iterations and delays
- Instill a culture of communication and collaboration
- Improve productivity
- Improve visibility into IT requirements and processes

DevOps Values - CAMS



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

DevOps Business Value

DevOps practices improve IT performance.

Companies with high IT performance are twice as likely to exceed their goals

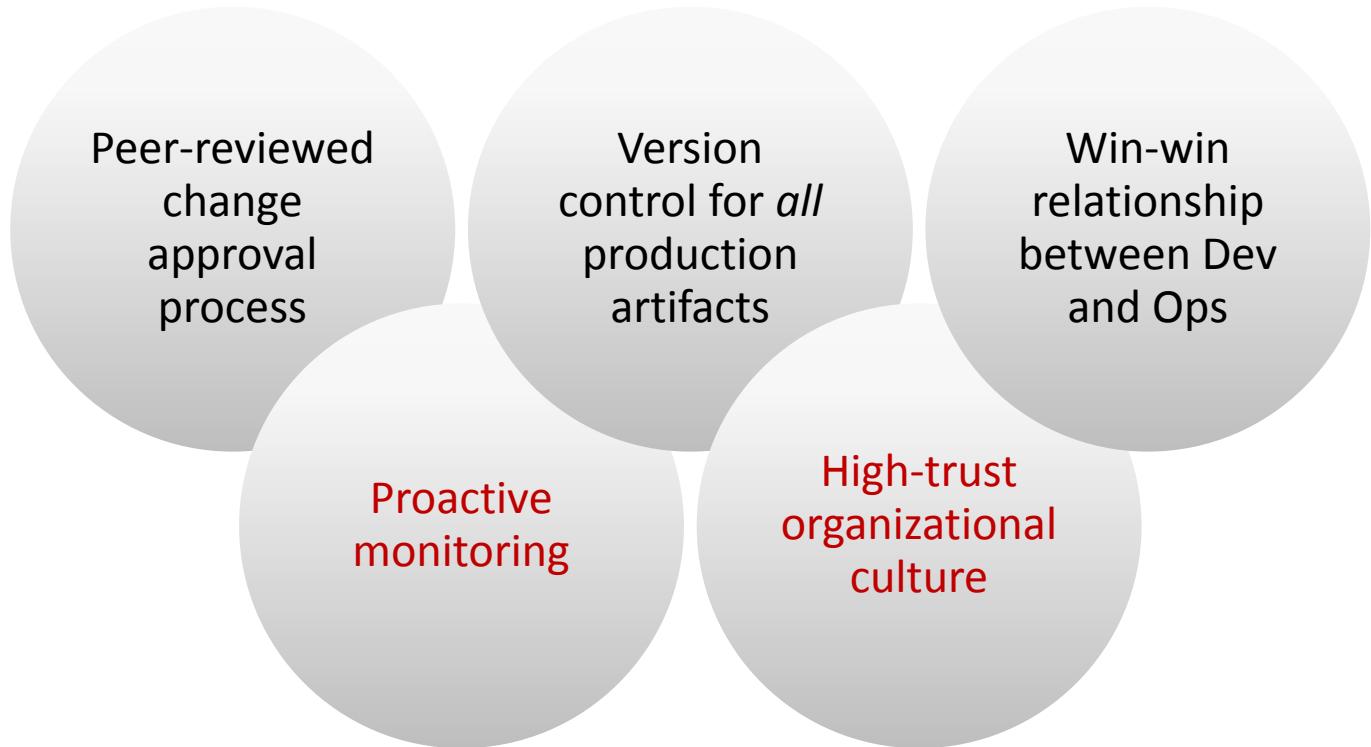
- Profitability
- Market share
- Productivity



Strong IT performance is a competitive advantage.

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

Top 5 Predictors of IT Performance



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

“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

What DevOps is NOT

- A title
- A team
- A tool
- Only culture
- Only automation
- NoOps
- The wild west



Who is Doing DevOps?

- **Web Pioneers**

- Amazon
- Netflix
- Etsy
- Facebook
- Google
- Twitter
- Pinterest

- **Academia**

- Portland State University
- Seton Hill University
- Kansas State University



- **Financial Institutions**

- BNY Mellon
- Bank of America
- World Bank

- **Retailers**

- Target
- The Gap
- Nordstrom
- REI
- Macy's

- **Insurance companies**

- Nationwide
- Travelers

and many more...

Startups vs. Enterprises



Unicorns



Horses

- Elusive web services or startup organizations
- Built upon agile and lean principles
- Active use of automation
- Inspiring but practices may be difficult to replicate
- Traditional enterprises that introduced technology later
- Built upon legacy systems and silos
- Less automation
- Recognize DevOps value but not sure how to proceed

The issues that DevOps addresses are universal: improving time to market, managing capacity, technical debt and maintaining stability and security.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Enterprise DevOps



Exercise:

- Based on the Wall Street Journal and Forbes articles, what is needed to make enterprise DevOps a reality?

The ability to gradually shape a new IT organizational culture will play a major role in creating a positive DevOps environment.

DEVOPS AND IT PERFORMANCE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Increases Agility and Stability

Organizations that implemented DevOps practices were up to five times more likely to be high performing.



- Organizations are more agile
 - Code is shipped 30 times faster
 - Deployments are completed 8000 times faster
- Services are more reliable
 - There are 50% fewer failures
 - Service is restored 12 times faster



Source: 2013 State of DevOps Report – Puppet Labs and IT Revolution Press

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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

Nassim Nicholas Taleb

MEASURING DEVOPS SUCCESS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Measuring Success (1)

Practices that enable organizations to understand and improve flow enable improved IT performance.

- IT performance is measured in terms of throughput and stability
- Throughput is measured by deployment frequency and lead time for changes
- Stability is measured by mean time to recover and the ability to preemptively detect and mitigate problems

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

© DevOps Institute

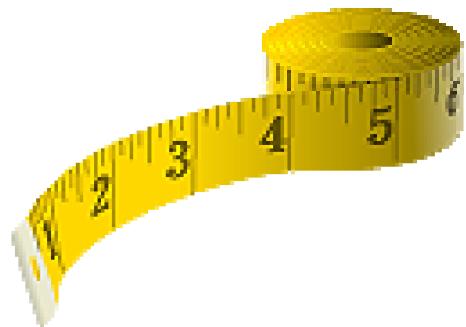
Copyrighted Material. Do Not Reproduce.

Measuring Success (2)

- Deployment frequency – Mean time between deploys
 - Change lead time – Time from request to delivery
 - Cycle time – Time from start of work to ready for delivery
- Change failure rate
 - Mean time to detect incidents (MTTD)
 - Mean time to recover (MTTR) – Component
 - Mean time to restore service (MTRS) - Service

© DevOps Institute

Copyrighted Material. Do Not Reproduce.



ULTIMATELY...

DevOps provides companies a competitive advantage by delivering better software, faster.

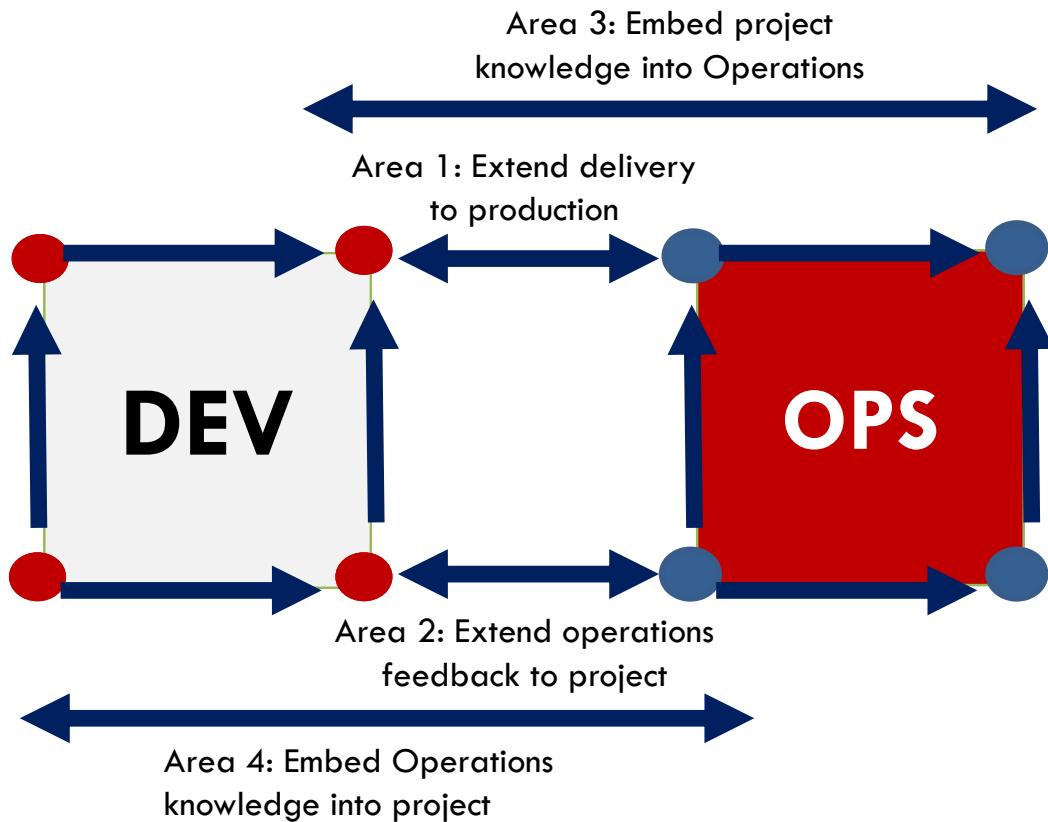


DEVOPS PATTERNS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Patterns



“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

DEVOPS PRACTICES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Practices

DevOps practices enable

- Improved flow
- Improved feedback loops
- Continuous experimentation and learning

DevOps practices use agile, lean, and ITSM methods and techniques to bring The Three Ways to life.

THE FIRST WAY: UNDERSTANDING AND IMPROVING FLOW

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

The First Way: Flow



- Understand the flow of work
- Always seek to increase flow
- Never pass a known defect downstream
- Never allow local optimization to cause global degradation
- Achieve a profound understanding of the system

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

Continuous Integration

Continuous integration is a development practice that requires developers to integrate code into a shared repository on a daily basis.



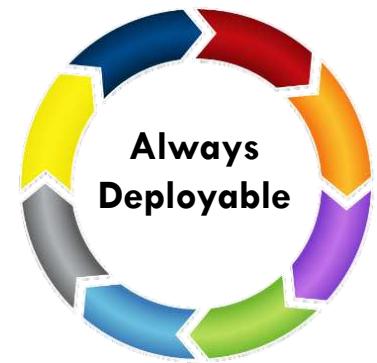
- Each check-in is validated by
- An automated build
 - Automated unit, integration and acceptance tests

Integrating regularly in production-like environments makes it easier to quickly detect and locate conflicts and errors.

Continuous Delivery (1)

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

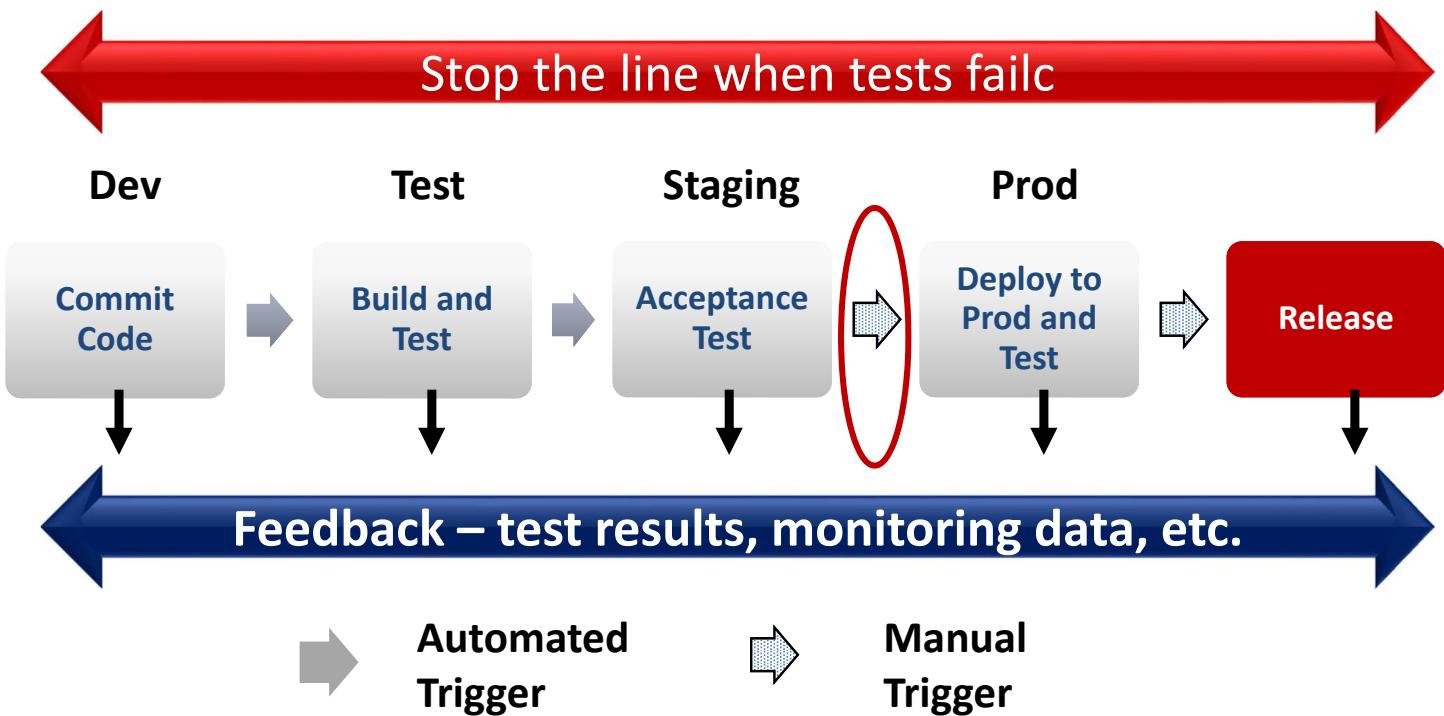
- Extends continuous integration
- Provides fast, automated feedback on the production-readiness of systems
- Prioritizes keeping software deployable over working on new features
- Enables push-button deployments on demand
- Reduces deployment risks and enables quicker user feedback



Continuous delivery does not mean that you are deploying every day or every hour. It means that you COULD release when needed.

Continuous Delivery (2)

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



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

© DevOps Institute
Copyrighted Material. Do Not Reproduce.

Continuous Deployment (1)

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

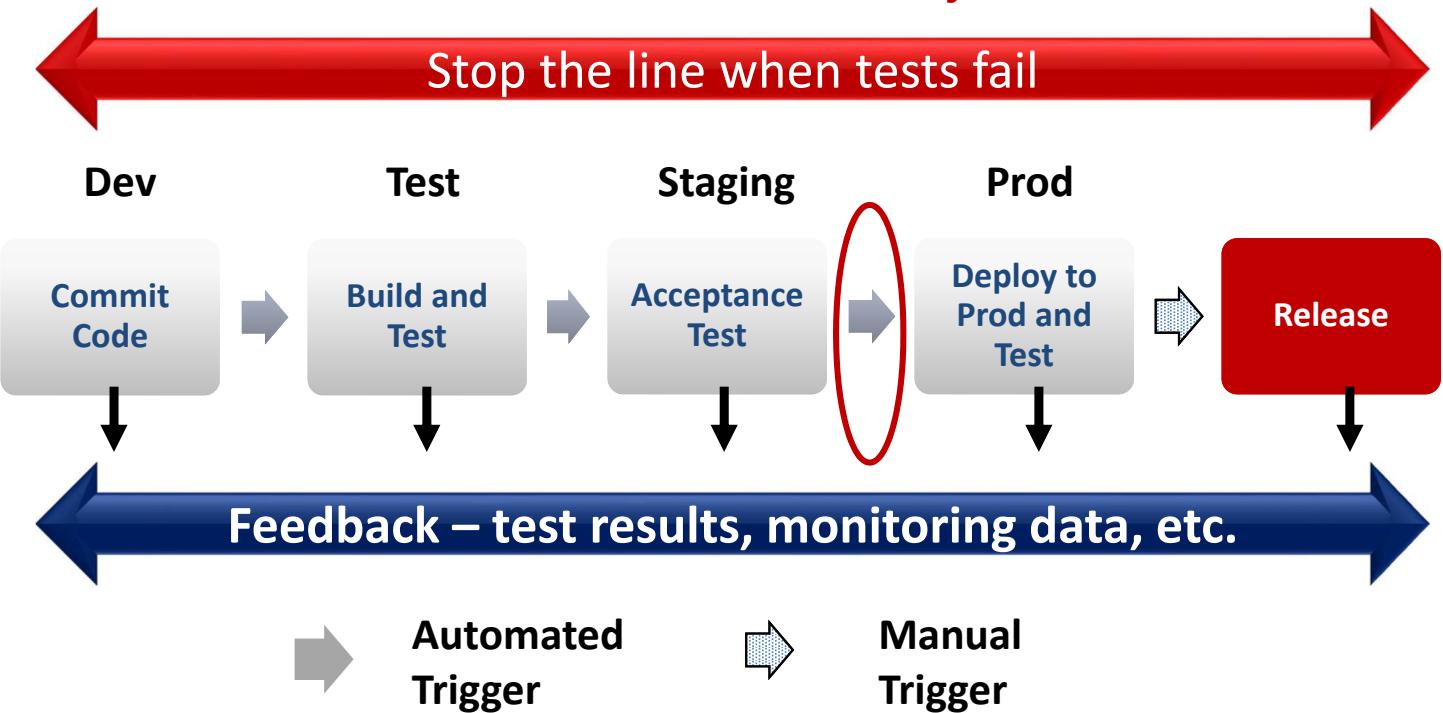
- Removes the manual step in the Continuous Delivery pipeline
- Results in multiple deployments per day



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.



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

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.

Enables cross-functional teams to

- See an entire process from a work and information flow perspective
- Agree on the current state of a value stream
- Analyze the current state
 - Timeline analysis
 - Waste analysis
- Identify, prioritize and measure improvements

Kanban

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

- Designed to reduce idle time and waste in a process
- Uses visual cards to trigger an action
- Teams pull work when they are ready for it
- Enables people to work collaboratively to improve flow

Improving Flow with a Kanban Board



- Make work visible
- Make policies explicit
- Limit work in progress (WIP) to capacity
- Visualize and manage workflow
- Measure velocity (quantity of work done in an iteration)

Kanban measures include lead time and cycle time.

Why is too much WIP bad?



© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Theory of Constraints (1)

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

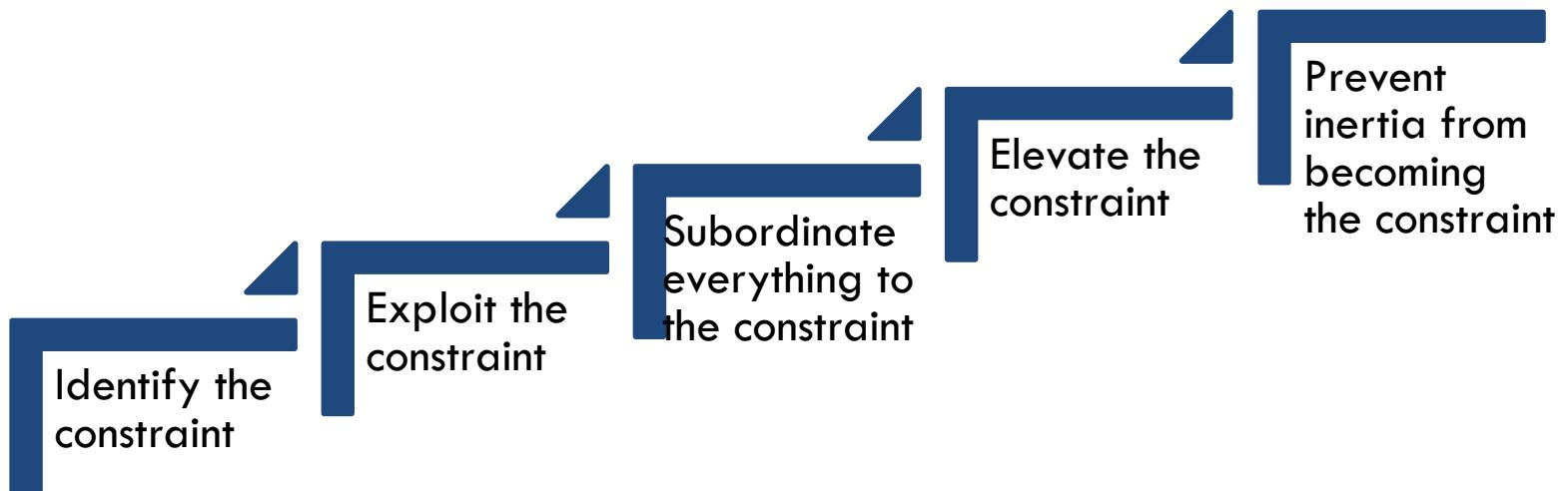
***The Theory of Constraints was introduced in the book
The Goal by Eliyahu M. Goldratt.***

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Theory of Constraints (2)

The Five Focusing Steps is a methodology for identifying and eliminating constraints.



Theory of Constraints (3)

The Five Focusing Steps	
Identify	Identify the constraint
Exploit	Make quick improvements using existing resources
Subordinate	Review all other activities in the process to ensure that they support the needs of the constraint
Elevate	If the constraint still exists, determine what other actions can be taken to eliminate it – continue until the constraint has moved
Prevent inertia	Start over – the Five Focusing Steps is a continuous improvement cycle

“We shouldn’t be looking at each local area and trying to trim it. We should be trying to optimize the whole system. Some resources have to have more capacity than others. The ones at the end of the line should have more than the ones at the beginning—sometimes a lot more.”

Eliyahu Goldratt

Exploiting Constraints

Exercise



- What can be done to exploit common constraints that affect the flow of work and feedback between Dev and Ops?

Common Constraints

- Environment creation (test, staging, production, etc.)
- Code deployment
- Test setup and run
- Overly tight architecture
- Development
- Product management

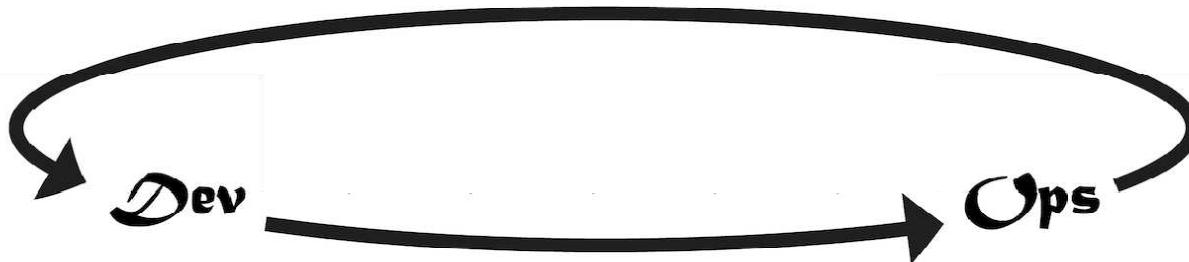


THE SECOND WAY: AMPLIFYING FEEDBACK LOOPS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

The Second Way: Feedback

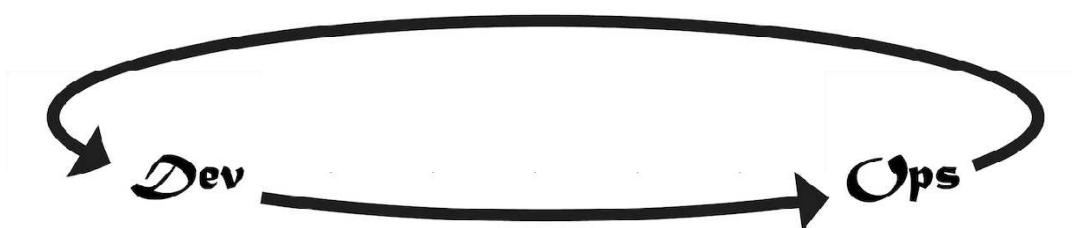


- Understand and respond to the needs of all customers – both internal and external
- Shorten and amplify all feedback loops
- Create and embed knowledge where needed

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: CONTINUOUSLY EXPERIMENTING AND LEARNING

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

The Third Way: Continual Experimentation and Learning

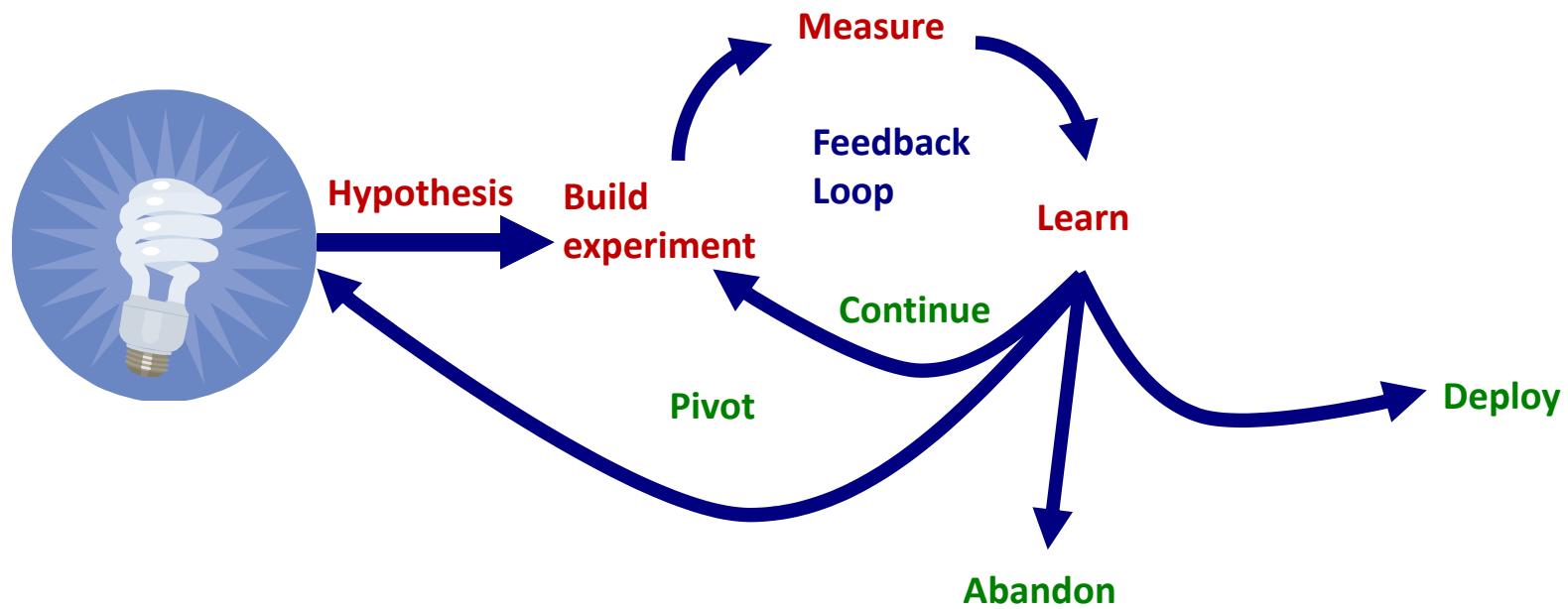
- Allocate time for the improvement of daily work
- Create rituals that reward the team for taking risks
- Introduce faults into the system to increase resilience

A goal of The Third Way is to create 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.

“The point is to extract learning FROM work, not impose more work.”

Charles Jennings

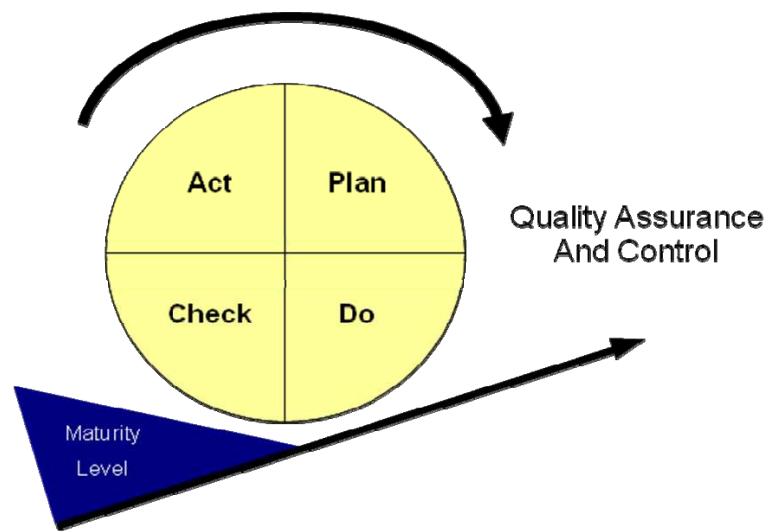
Experimentation and Learning



Experimentation and learning through feedback answers the question “Should this product be built?”

The Deming Cycle (PDCA)

- The Deming Cycle (also known as the Shewhart Cycle) enables incremental improvement by executing four simple steps
 - Plan
 - Do
 - Check
 - Act



“Even more important than daily work is the improvement of daily work.”

Mike Orzen

Improvement Kata

The Improvement Kata is 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

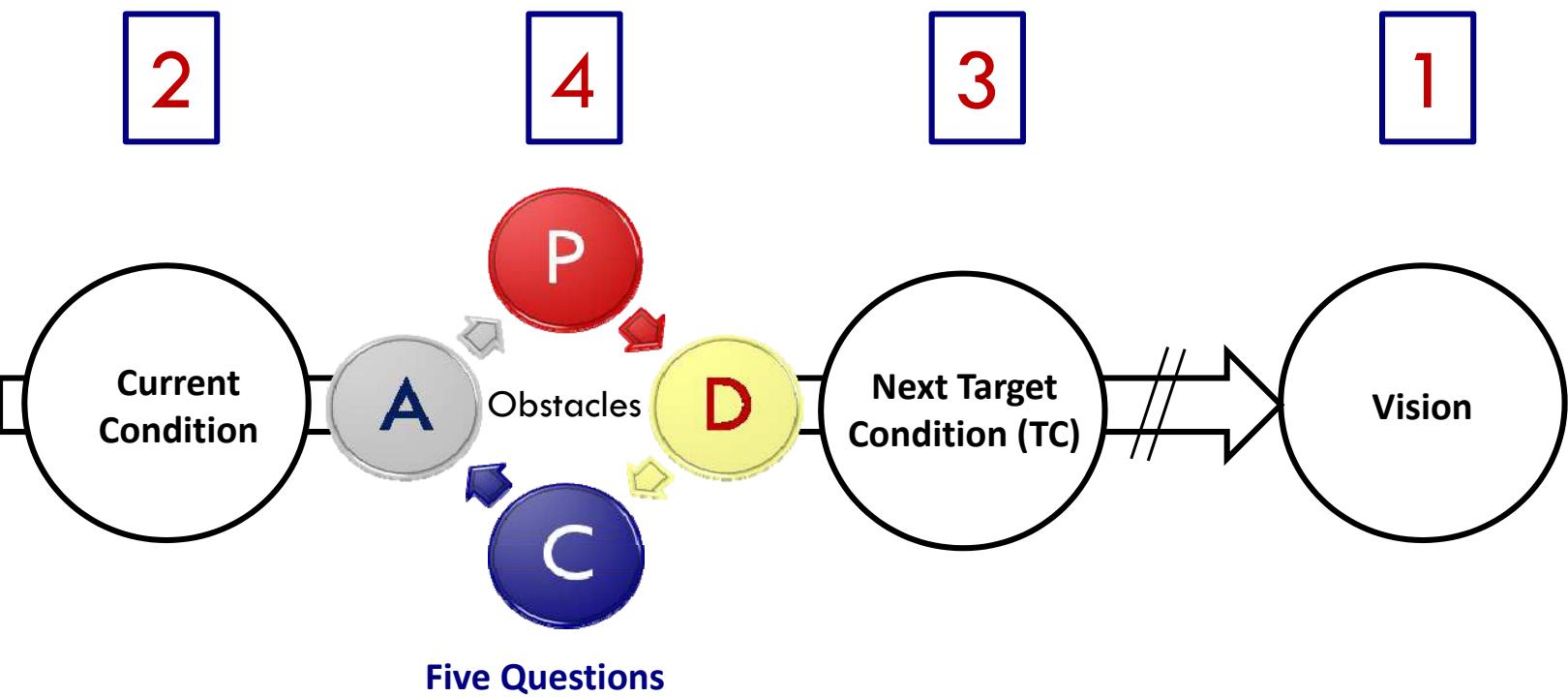
Practicing the Improvement Kata should be a daily occurrence, not just an add-on or ‘as time allows’ project.

Improvement Kata – Four Steps (1)

1. Understand the long-term vision or direction
2. Grasp the current condition
3. Establish the next target condition
4. PDCA/experiment toward the target condition
 - Use the 5 questions

The Improvement Kata is a four-step process that focuses on learning and improving work.

Improvement Kata – Four Steps (2)



The Improvement Kata considers the organization's long-term vision or direction.

Improvement Kata – Five Questions

- What is the target condition?
- What is the actual condition now?
 - Reflect on previous Kata
- What obstacles do you think are preventing you from reaching the target condition?
 - Which one are you addressing now?
- What is your next step? (next PDCA/experiment)
 - What do you expect?
- 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.

**“Excellent firms don’t believe in
excellence - only in constant improvement
and constant change.”**

Tom Peters

PREPARING FOR FAILURE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Using Failure to Improve Resiliency

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

- The ‘Simian Army’ concept was first adopted by Netflix
- The concept is designed to ensure your production environment can recover from inevitable failures
- Tools like the Chaos Monkey
 - Apply monitoring, diagnostics, randomization and disruption to the infrastructure
 - Ensure engineers apply automation to limit the use impact when big problems do occur

***Chaos Monkey is a service which identifies groups of systems and randomly terminates one of the systems in a group.
(Netflix)***

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Bringing The Three Ways to Life

Exercise:

- Describe how continuous deployment supports each of The Three Ways



ITSM PROCESS IMPROVEMENT PRACTICES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps and ITSM (1)

- Every organization has ITSM processes in place
- Many based on ITIL best practice
- ITIL encourages organizations to continuously adapt processes to current business needs
- Achieve this by applying agile, lean and DevOps principles over a foundation of ITIL best practice
- Start with your key ITIL processes

Honor the past but don't be bound to it.

DevOps and ITSM (2)

Improvement opportunities include

- Reduce handoffs
- Eliminate bottlenecks
- Eliminate unnecessary checks and reviews
- Eliminate or correct activities that result in rework
- Capture information once at the source
- Eliminate duplicate activities
- Substitute parallel activities for sequential activities
- Consolidate roles when possible



Prioritize improvements that enable improved communication, collaboration and integration.

Source: *The ITSM Process Design Guide*

Maximize the Use of Models

- 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

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Examples

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

Based on ITIL Text - ST 4.2.4.5

“Almost all quality improvement comes via simplification...”

Tom Peters

Change Management

Common Practices	DevOps Practices
Viewed as a barrier/bottleneck	View as a way to mitigate risk
Process isn't triggered/stakeholders aren't engaged early enough	Increase proactive collaboration between stakeholders
Change Advisory Board (CAB) must approve too many changes	Increase percentage of standard changes Increase number of automated approvals (e.g., if it integrates and passes tests, it is approved)
Find the guilty party or cover up mistakes when things go wrong	Capture – in a blameless way – and act on lessons learned Use Problem Management where needed

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Release and Deployment Management (1)

- What is your current deployment frequency?



- What does the frequency need to be?
- How fast do you need to get there?

The definition of “go faster” will depend on the service, customer requirements and available capabilities and resources.

Release and Deployment Management (2)

Common Practices	DevOps Practices
Infrequent release windows	Talk to the business! Determine appropriate rate of change for each service
Delays because environments aren't ready	Cross-functional teams – get Ops involved early – work in parallel Change agile sprint policies - require a working environment in addition to shippable code Automate the provisioning and synchronization of all environments Self-service environment creation
Scheduling delays caused by conflicts	Continuous integration Automated testing

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Release and Deployment Management (3)

Common Practices	DevOps Practices
Deployment delays	Plan the release in parallel with development Make impediments visible Automated deployment
Recurring problems	Shorten feedback loops by releasing more frequently Involve incident, Problem and Knowledge Management
Dev resources aren't available when change is released to production	Agree upon exit and entry criteria for early life support

Service Asset and Configuration Management (SACM)

Common Practices	DevOps Practices
Numerous disjointed configuration management databases – none of which are up to date	Build releases that can be automatically discovered
Traditionally focused on physical devices	Define new configuration item (CI) types

Problem Management

Common Practices	DevOps Practices
Viewed as reactive	Engage earlier in lifecycle
Not acted upon	Capture knowledge (known errors) as soon as it makes sense to do so Make changes as needed to prevent problems when justified
Perceived as slowing Dev down	Proactively identify problems
Viewed as brainstorming	Learn and practice proven techniques

Incident Management (1)

Common Practices	DevOps Practices
Errors reported late in the development cycle or by users	Aim for quality at the source Set thresholds Establish proactive monitoring
Ops feels the pain	Dev owns the quality of their code Dev provides Level 3 support Dev and Ops work together in cross-functional teams
Inefficient escalation practices	Clarify roles, responsibilities, accountabilities (ownership) Identify and exploit bottlenecks Improve feedback loops

Incident Management (2)

Common Practices	DevOps Practices
Finger pointing, blame game, cover up mistakes	View everything that happens as a learning opportunity Dev and Ops collaborate to restore service
Incidents recur	Capture knowledge Use Problem Management

Knowledge Management

Common Practices	DevOps Practices
Knowledge is transferred late in the software development cycle	Capture knowledge throughout the lifecycle
Knowledge is rediscovered	Implement or improvement Knowledge Management capabilities
Knowledge takes too long to be available	Consider ‘on-demand’ practices such as those introduced in Knowledge-Centered Support (KCS)

Exercise: Integrating DevOps and ITSM

- What other ITSM process(es) integrate closely with DevOps?
- What practices may need to be refined for these process(es) to integrate more effectively with DevOps?



ADOPTING A DEVOPS CULTURE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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

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

“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

Get Clear on the Business Opportunity

The ‘Why?’

- Understand
 - Existing values, beliefs and expectations
 - The real (vs. perceived) drivers for change
- Consider
 - Business strategies and goals
 - Customer needs
 - The organization’s tolerance for risk
 - The organization’s current capabilities and resources
- Seek input and buy-in from stakeholders

Your ‘why’ will determine which DevOps principles and practices the organization needs to adopt initially.

***“DevOps is not your why,
not your co-workers’ why,
certainly not your business’ why.”***

Damon Edwards

Get the Right People Together

The right people are

- Core stakeholders in the change
- Informal leaders
- Business oriented
- Committed to experimentation and learning
- Creative problem solvers
- Persistent
- Effective communicators
- Willing to ‘walk the talk’ and model new behaviors
- Trusting and trustworthy



Get Everyone on the Same Page

- Seek to understand each others' perspectives and concerns
- Determine what outcomes you want to achieve
 - Short and long term
- Set measurable goals
 - Focus on improving flow and amplifying feedback loops
 - Avoid goals that conflict with existing performance management programs
- Be realistic
 - Consider peoples' beliefs about the current culture
 - Consider the organization's capabilities

The only 'right way' to adopt a DevOps culture is the way that is right for your organization at this time.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Build Capabilities that Will Last

- Introduce a common vocabulary
- Leverage early adopters and informal networks of peer motivators
- Provide ongoing education and just in time training
- Foster a culture of continuous improvement and learning
 - Think agile! Think lean!
- Build trust through transparency
- Generate and celebrate short-term wins

Showing early and repeated success is crucial!

Focus on Critical Behaviors

- Agility
- Forward thinking
- Consistent messaging
- Communication
- Collaboration
- Decision making
- Problem solving
- Transparency
- Continuous learning
- Accountability
- Empowerment
- Conflict resolution

Every culture has behaviors that help enable change and others that hinder it. Find ways to nurture the enabling behaviors that matter most.

Changing Behavior

- Enable people to feel proud
 - Leverage cultural strength
 - Celebrate wins, risk taking, innovation
 - Encourage people to share
- Make the new behaviors
 - Easy
 - Rewarding
 - Normal



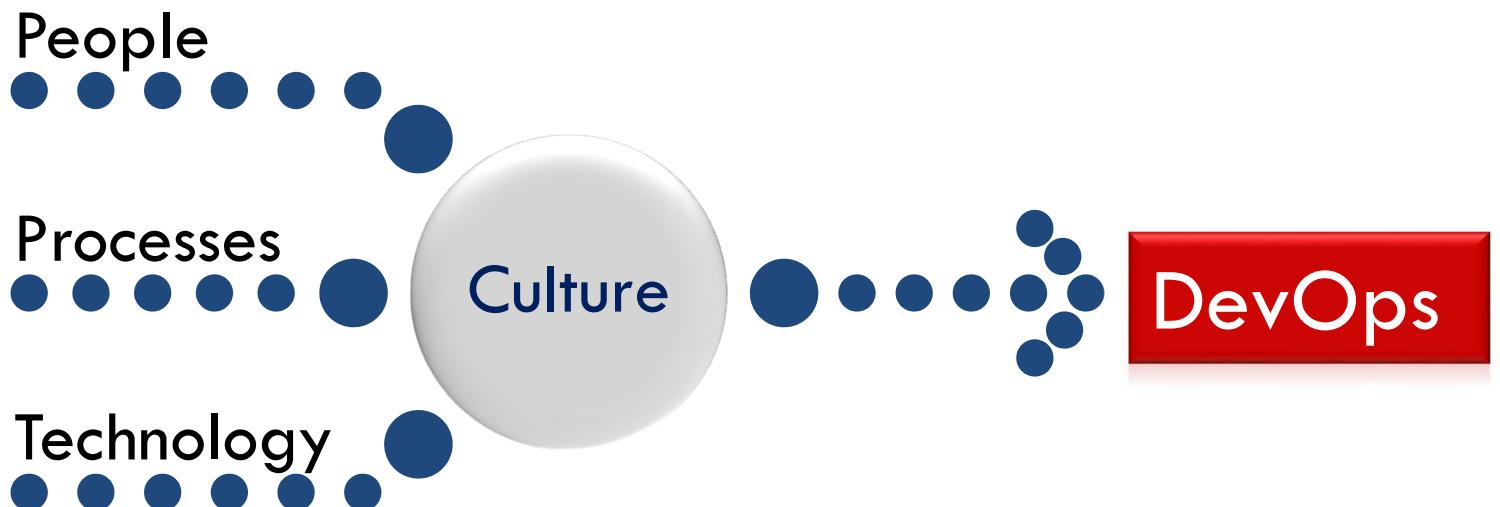
**“People don’t resist change.
They resist being changed.”**

Peter Senge

Experiment and Learn

- Prioritize improvement opportunities
- Take a holistic approach
- Select (carefully) and run pilot project(s)
- Accept that there will be failures
- Capture lessons learned
- Share, rather than enforce, improved practices
 - Include a ‘what NOT to do’ list
- Do it again

Taking a Holistic Approach



In that order!

Selecting a Pilot

- Maximize probability of success
- Small enough where
 - Success is possible
 - Consequences of failure aren't so large that a mistake could shut down the entire initiative
- Large enough that
 - We can show proof of improvement
 - We earn the right to make future improvements

Consolidate Gains and Produce More Change

- Communicate successes, failure 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



Cultural change comes last, not first.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

“Change sticks when it becomes ‘the way we do things around here.”

John. P. Kotter

CHALLENGES, RISKS AND CRITICAL SUCCESS FACTORS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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.

Critical Success Factors

- Management commitment to culture change
- Creation of a collaborative, learning culture
- 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



SUMMARY

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Summary



- DevOps enables companies to deliver better software, faster by...
 - Improving flow
 - Shortening and amplifying feedback loops
 - Fostering a culture of continuous experimentation and learning

DevOps benefits the business by improving communication, collaboration and the integration of processes and tools.

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

Forrester Research

DevOps Foundation

Study Aids Answer Key

DevOps Key Terms: Matching Exercise

In the blank before each term in Column A, write the letter corresponding to the definition from Column B.

Column A	Column B
<u>D</u> Agile Manifesto	A. Cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.
<u>G</u> IT Infrastructure Library (ITIL)	B. Software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
<u>A</u> DevOps	C. Production philosophy that focuses on reducing waste to improve overall customer value.
<u>E</u> IT Service Management	D. Formal proclamation of values and principles to guide an iterative and people-centric approach to software development.
<u>B</u> Agile Software Development	E. Implementation and management of quality IT services that meet the needs of the business. (ITIL definition)
<u>F</u> Scrum	F. Iterative and incremental Agile Framework within which complex products can be developed.
<u>C</u> Lean Production	G. Set of best practice publications for IT service management. Published in a series of five core books representing the stages of the IT service lifecycle.

The Three Ways

Specify the correct order of The Three Ways by entering 1-3 in the left column.

Three Ways Order	The Three Ways
3	Create a culture that fosters experimentation and learning
1	Understand and increase the flow of work
2	Create short feedback loops that enable continuous improvement

Agile Manifesto

Using arrows (as illustrated), align the values on the left to the corresponding values on the right.

Value	Over...	Value
Individuals and interactions		Comprehensive documentation
Working software		Following a plan
Customer collaboration		Processes and tools
Responding to change		Contract negotiations

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

DevOps Practices: Matching Exercise

In the blank before each term in Column A, write the letter corresponding to the definition from Column B.

Column A	Column B
<u>C</u> Kanban	A. Methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.
<u>D</u> Value Stream Mapping	B. Development practice that requires developers to integrate code into a shared repository on a daily basis.
<u>F</u> Theory of Constraints	C. Method for managing knowledge work with an emphasis on just-in-time delivery while not overloading the team members.
<u>B</u> Continuous Integration	D. Lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer.
<u>A</u> Continuous Delivery	E. Set of practices that enable every change that passes automated tests to be automatically deployed to production.
<u>E</u> Continuous Deployment	F. Set of management tools that views any manageable system as being limited in achieving more of its goals by a very small number of bottlenecks.

Theory of Constraints

Specify the correct order of the five focusing steps by entering 1-5 in the left column.

Focusing Steps Order	Five Focusing Steps
4	Elevate the constraint
1	Identify constraint
5	Prevent inertia from becoming the constraint
2	Exploit the constraint
3	Subordinate everything to the constraint

Improvement Kata: Fill in the blanks

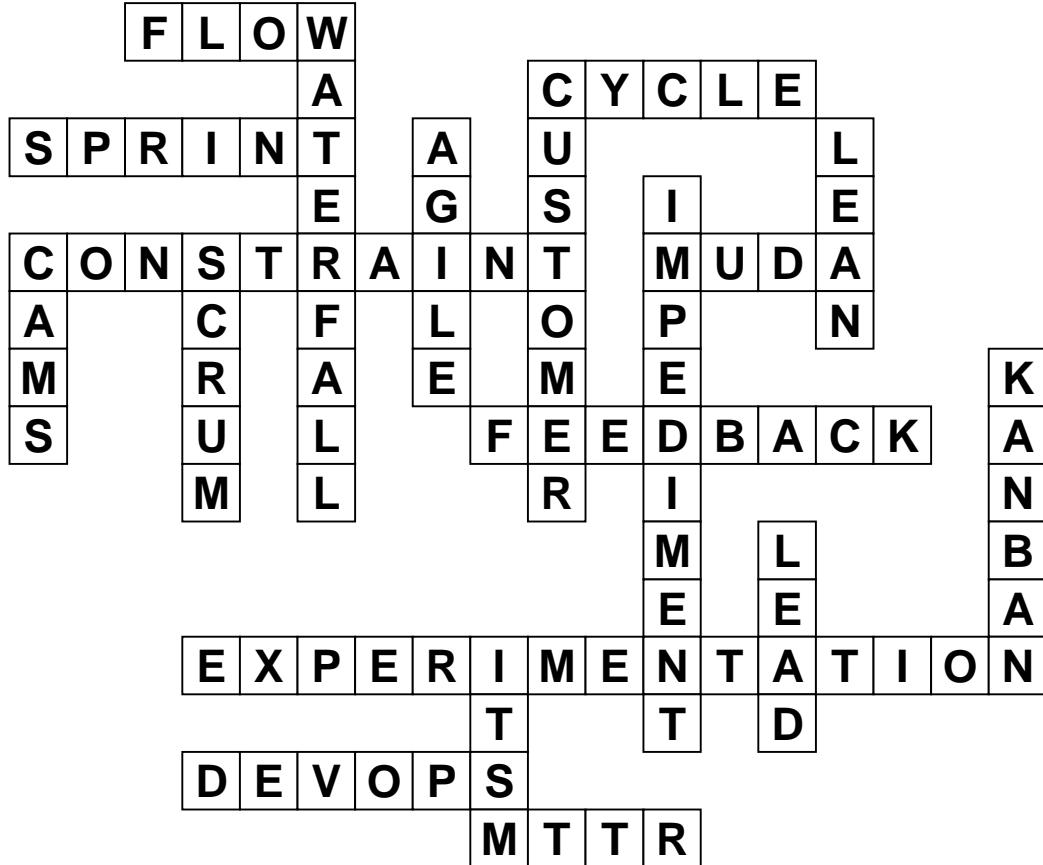
Correctly complete each of the following statements.

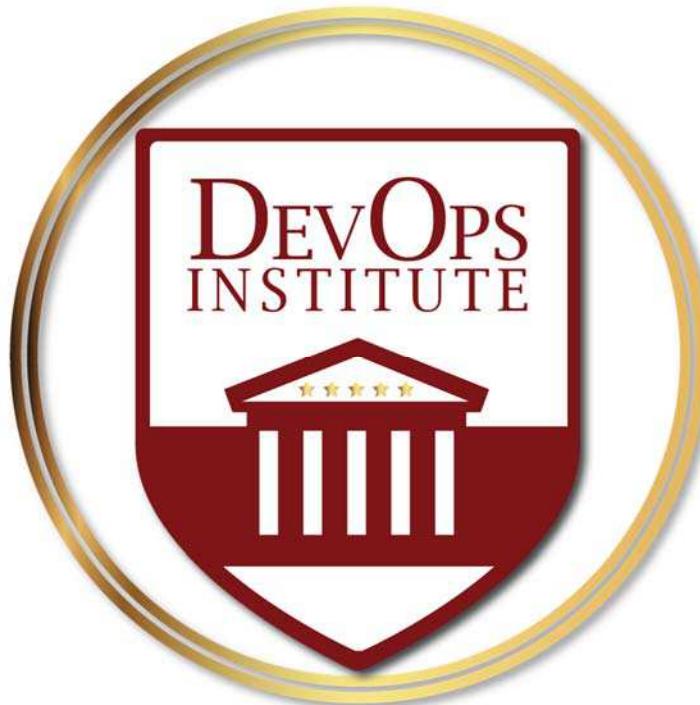
Improvement Kata
1. Understand the <u>vision or direction</u>
2. Grasp the <u>current condition</u>
3. Establish the next <u>target</u> condition
4. <u>PDCA</u> toward the target condition

DevOps

Vocabulary Terms and Acronyms

Solution:





DevOps Foundation Sample Exam

Sample Paper Version 1.0

Multiple Choice

- This paper contains 40 questions, each worth 1 point
- Circle the correct response for each question
- You have 60 minutes to complete this paper
- You must achieve 26 or more out of a possible 40 points (65%) to pass this examination

1. Which is NOT a goal of DevOps?

 - a. Improved productivity
 - b. Fewer but higher-quality software releases
 - c. Lower risk software deployments
 - d. Improved quality of code

2. Which metric BEST demonstrates DevOps business value?

 - a. Reduced time to deploy changes
 - b. Reduced cost to deploy changes
 - c. Increased profitability
 - d. Reduced deployment-related errors

3. Which statement about DevOps is CORRECT? DevOps is...

 - a. An agile software engineering standard
 - b. A team that combines Dev and Ops staff
 - c. A software continuous delivery tool
 - d. A commitment to communicate and collaboration

4. An organization's development teams are using agile methods. The changes are queued up and applied to production systems on a quarterly basis. How would adopting a DevOps approach benefit this organization?

 - a. Reduced risks due to smaller releases
 - b. No benefit – the approach is sound
 - c. Reduced cost of product iterations and delays
 - d. Both A and C

5. Which DevOps principle aims to shorten and amplify right to left feedback loops?
 - a. The First Way
 - b. The Second Way
 - c. The Third Way
 - d. The Fourth Way
6. Which is a goal of the First Way?
 - a. Always seek to increase flow
 - b. Embed Ops in Dev
 - c. Document known defects before passing them downstream
 - d. Understand and respond to the needs of all customers
7. What is the Three Ways?
 - a. Methodology for identifying and removing constraints
 - b. The key principles of DevOps
 - c. Disciplined, data-driven approach for reducing waste
 - d. A methodology for performing continuous improvement
8. In the context of agile software development, which of the following is NOT a responsibility of IT Operations?
 - a. Managing the product backlog
 - b. Defining non-functional requirements
 - c. Identifying security requirements
 - d. Provisioning the infrastructure

9. In the context of lean, what must happen to build trust and consolidate the gains?

- a. Communicate successes, failures and lessons learned
- b. Effective use of lean tools
- c. Limit work in progress
- d. Limit over-processing

10. What is the Agile Manifesto?

- a. Values and principles to guide an iterative and people-centric approach to software development
- b. Methodology that focuses on making sure software is always in a releasable state throughout its lifecycle
- c. Declaration of the benefits and intentions of DevOps
- d. Intentions and motives of being an agile enterprise

11. An organization has had unsuccessful software releases due to constantly changing customer requirements but management is reluctant to abandon the waterfall method of software development. Which is the BEST approach for a software development manager to take?

- a. Prepare a presentation outlining the benefits of agile software development
- b. Suggest adopting agile practices in a sub-segment of the business to show proof of concept
- c. Use the Theory of Constraints to eliminate bottlenecks in the waterfall process
- d. Work with process owners to improve the Change and Release Management processes

12. How do ITSM processes support DevOps?

- a. Enable continuous improvement
- b. Reduce constraints
- c. Increase flow
- d. All of the above

13. From whose perspective is value defined according to the principles of lean thinking?

- a. The end customer
- b. The end customer and IT
- c. All stakeholders
- d. The DevOps team

14. Which can be used to create a learning culture?

- a. Experimentation and sharing
- b. Problem solving
- c. Managers valuing and modeling learning behavior
- d. All of the above

15. Which statement BEST describes change fatigue?

- a. Aggressive resistance
- b. Apathy
- c. Finger pointing
- d. Exhaustion

16. Which is a characteristic of a DevOps culture?

- a. Effective one-way communication from the top down
- b. Recognizing the best and brightest for their successes
- c. Shared vision, goals and incentives
- d. All of the above

17. When trying to effect major change, who should be engaged in planning activities and serve as change agents?

- a. Early adopters
- b. Naysayers
- c. Management
- d. People who need proof

18. Which statement about DevOps teams is MOST accurate?

- a. Accountable for establishing DevOps practices across the enterprise
- b. Accountable for performing all DevOps practices
- c. Focuses on one project at a time
- d. The pros and cons aren't clear

19. Which approach would BEST enable an organization to adopt a more collaborative culture when developing a new product?

- a. Share tasks among team members
- b. Have management dictate tasks to team members
- c. Encourage team members to compete against each other
- d. Reward team members for working independently

20. DevOps practices in relation to Change Management include...

- a. Using change models
- b. Increasing the number of automated approvals
- c. Increasing the percentage of standard changes
- d. All of the above

21.Which statement about Kanban is CORRECT?

- a. Pushes work through a process
- b. Requires a workflow management tool
- c. Pulls work through a process
- d. Enables more work in progress

22.How can an organization adopting DevOps practices benefit most from Problem Management?

- a. Engage it reactively to solve recurring problems
- b. Learn and practice proven techniques
- c. Use it as a brainstorming method to solve complex problems
- d. Use it to restore service in the live environment

23.Which statement about continuous delivery is CORRECT?

- a. Ensures software is error free
- b. Ensures software is always in a releasable state throughout its lifecycle
- c. Ensures code is integrated into a shared version control system on a daily basis
- d. Software that passes automated tests is automatically deployed to production

24.Which of the following enable continuous delivery?

- a. Continuous integration
- b. Automated testing
- c. Organizing Dev into smaller teams
- d. Both A and B

25.What is the first step in the Theory of Constraints?

- a. Define the plan
- b. Assemble the team
- c. Identify the constraint
- d. Map the value stream

26.Which statement about the Improvement Kata is CORRECT?

- a. It focuses on short term goals
- b. It is a 7-step process
- c. It considers the organization's long-term vision or direction
- d. It should be performed as time allows

27.Which DevOps practice supports The Second Way

- a. Dashboards
- b. Kanban boards
- c. Continuous delivery
- d. Improvement Kata

28.Which statement about models is CORRECT?

- a. Cannot be automated
- b. Define steps for a specific type of transaction
- c. Work best with Change Management
- d. Define steps for handling all forms of a transaction

29.In a DevOps culture, which is the best way to determine the frequency of release windows

- a. Match the velocity of the development teams
- b. Talk to the business
- c. Defer to the Change Advisory Board (CAB)
- d. Base on the results of peer reviews

30.In the context of DevOps, which is an effective approach when selecting tools?

- a. Establish a tool chain
- b. Standardize on a single vendor's product
- c. Allow both Dev and Ops to select the best tools for their needs
- d. Select tools recommended by industry analysts

31.In the context of DevOps and automation, which is an effective practice?

- a. Automate error-prone work
- b. Automate monitoring and notification activities
- c. Automate complex tasks
- d. Both A and B

32.Which reflects the BEST place to start when automating work?

- a. Automate high-value tasks
- b. Automate to improve collaboration
- c. Simplify processes
- d. Automate monitoring and notification activities

33.Which DevOps practice BEST helps to reduce delays in deployment due to unready environments?

- a. Automate the provisioning and updating of all environments
- b. Agree upon exit and entry criteria for early life support
- c. Do not involve Ops until pilots
- d. Use Problem Management to determine the cause of the delays

34. In the context of a DevOps tool chain, which category of tool provides a single source of truth?

- a. Configuration management
- b. Version control
- c. Continuous delivery
- d. Monitoring

35. Which statement about configuration management tools is CORRECT?

- a. Ensure resource configurations comply with a desired state
- b. Used to track incremental versions of files
- c. Enable ITSM process integration
- d. All of the above

36. Which type of collaboration tool enables at a glance status updates?

- a. Document sharing tools
- b. Dashboards
- c. Wikis
- d. Group chat rooms

37. What determines which DevOps principles and practices will BEST benefit an organization?

- a. Business strategies and goals
- b. The commitment of early adopters
- c. The availability of advanced tools
- d. IT's capabilities and resources

38.An organization's management team is thinking of outsourcing the IT organization because extended outages in the production environment are affecting employees' productivity. Which measure should IT focus on to improve productivity?

- a. Change lead time
- b. Mean time to restore service
- c. Change failure rate
- d. Mean time to detect

39.In the context of lean thinking, which metric BEST reflects improvements to the end-to-end IT value stream?

- a. Change lead time
- b. Mean time between deployments
- c. Cycle time
- d. Mean time to recover

40.Which is a critical success factor for DevOps?

- a. Establishing a tool chain
- b. Hiring DevOps Engineers
- c. Management commitment to culture change
- d. Automating everything

DevOps

Glossary of Terms

Term	Definition
Agile (adjective)	Able to move quickly and easily; well-coordinated. Able to think and understand quickly; able to solve problems and have new ideas.
Agile Enterprise	Fast moving, flexible and robust company capable of rapid response to unexpected challenges, events, and opportunities.
Agile Project Management	Iterative and incremental method of software design and development in which decisions are made based on observation and experimentation rather than on detailed upfront planning.
Agile Software Development	Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
Antifragile	The ability to <i>improve</i> with stress rather than merely resist it.
Agile Manifesto	A formal proclamation of values and principles to guide an iterative and people-centric approach to software development. Also called the Manifesto for Agile Software Development.
Availability	Ability of a service, component or configuration item (CI) to perform its agreed function when required. Availability is usually calculated as a percentage based on agreed service time and downtime. (ITIL definition)
Baseline	Marker or starting data point that can be used for later comparison. (ITIL definition)
Best Practice	Proven practice that has been successfully used by multiple organizations. Best practices can include activities, processes, functions, standards and guidelines. (ITIL definition)
Bottleneck (Lean Manufacturing)	Step in a process that limits the total capacity of the process. See also <i>constraint</i> .
Business Case	Decision support and planning tool that projects the likely consequences – quantitative and/or qualitative – of a business action. A business case justifies the reason for a significant expenditure. (ITIL definition)
Business Relationship Management (BRM)	Process responsible for maintaining a positive relationship with customers. (ITIL definition)
Cadence	Flow or rhythm of events.
CAMS Model	Pillars of DevOps: Culture, Automation, Measurement, Sharing (as put forth by John Willis).
CALMS Model	Pillars of DevOps: Culture, Automation, Lean, Measurement, Sharing (as put forth by Jez Humble).
Change	Addition, modification or removal of anything that could have an effect on IT services. (ITIL definition)
Change Advisory Board (CAB)	Body that exists to support the authorization of changes and to assist Change Management in the assessment, prioritization and scheduling of changes. Includes representatives from IT, the business and suppliers. (ITIL definition)
Change Management	Process that controls all changes throughout their lifecycle. (ITIL definition)
Change Model	Repeatable way of dealing with a particular category of change. Describes specific pre-defined steps that will be followed for a change of this category. (ITIL definition)
Change Schedule	Details of changes authorized for implementation and their planned implementation dates. (ITIL definition)

Term	Definition
ChatOps	An approach to managing technical and business operations through a group chat room. (coined by GitHub)
Configuration Item (CI)	Service asset that must be managed to deliver an IT service. (ITIL definition)
Configuration Management	See Service Asset and Configuration Management.
Configuration Management Database (CMDB)	Database used to store configuration records throughout their lifecycle. Configuration records store attributes of CIs and their relationships with other CIs. (ITIL definition)
Configuration Management System (CMS)	Set of tools and databases used to manage configuration data and related information (e.g., incidents, problems, known errors, changes, releases and people). (ITIL definition)
Constraint	Limitation or restriction; something that constrains. See also <i>bottleneck</i> .
Continual Service Improvement (CSI)	One of the ITIL Core publications and a stage of the service lifecycle. Continual Service Improvement aligns IT services with changing business needs by identifying and implementing improvements to IT services that support business processes throughout the service lifecycle. (ITIL definition)
Continuous Delivery	A methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.
Continuous Flow	Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps.
Continuous Integration	A development practice that requires developers to merge their code into a shared version control system — ideally, multiple times per day.
Critical Success Factor (CSF)	Something that must happen for an IT service, process, plan, project or other activity to succeed. Key performance indicators are used to measure the achievement of CSFs. See Key Performance Indicators. (ITIL definition)
CSI Register	Database or document used to record and manage improvement opportunities throughout their lifecycle. (ITIL definition)
Culture (Organizational Culture)	The whole of the ideas, corporate values, beliefs, practices and expectations about behavior and daily customs that are shared by the employees in an organization.
Customer	Someone who buys goods or services. (ITIL definition)
Definition of Done (DoD)	Shared understanding of what it means for work to be complete.
Definitive Media Library (DML)	One or more secure libraries in which the definitive and authorized versions of media CIs are stored and protected.
Demand Management	Process that understands, anticipates and influences customer demand for services. (ITIL definition)
Deming Cycle	A four-stage cycle for process management, attributed to W. Edwards Deming. Also called Plan-Do-Check-Act (PDCA).
Design Coordination	Process that ensures Service Design goals and objectives are met by providing and maintaining a single point of coordination and control for all activities and processes within the stage. (ITIL definition)
Dev (in the context of DevOps)	Individuals involved in software development activities such as application and software engineers.

Term	Definition
DevOps	Cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.
Downtime	Time when a CI or IT service is not available during its agreed service time. (ITIL definition)
Early Life Support (ELS)	Period of time that occurs at the end of deployment and before a service is accepted into operation. (ITIL definition)
Emergency Change	Change that must be introduced immediately, typically to repair an error in an IT service that is negatively impacting the business to a high degree. A change intended to introduce required business improvements is NOT an emergency, rather it is handled as a normal change but assigned the highest urgency. (ITIL definition)
Emergency Change Advisory Board (ECAB)	Subgroup of the CAB that is authorized to make decisions about emergency changes. (ITIL definition)
Event	Any change of state that has significance for the management of a CI or IT service. (ITIL definition)
Event Management	Process that detects events, makes sense of them and determines the appropriate control action. (ITIL definition)
Financial Management	Process that secures the level of funding needed to design, develop and deliver services that meet the organization's strategy. (ITIL definition)
Flow	How people or products move through a process. Also known as single-piece flow.
Gemba	Japanese word for 'the real place.' In business it typically means where value is created.
Impact	Measure of the effect of an incident, problem or change on business processes. Impact and urgency are used to assign priority. (ITIL definition)
Incident	Any unplanned interruption to an IT service or reduction in the quality of an IT service. Includes events that disrupt or could disrupt the service. (ITIL definition)
Incident Management	Process that restores normal service operation as quickly as possible to minimize business impact and ensure that agreed levels of service quality are maintained. (ITIL definition)
Information Security Management (ISM)	Process that aligns IT security with business security and ensures that the confidentiality, integrity and availability of the organization's assets, information, data and IT services always matches the agreed needs of the business. (ITIL definition)
Information Security Management System (ISMS)	Framework used to design, implement, manage, maintain and enforce information security processes and controls throughout the organization. (ITIL definition)
Information Security Policy (ISP)	Policy that governs an organization's approach to Information Security Management. The ISP covers the use and misuse of all IT systems and services. (ITIL definition)

Term	Definition
Infrastructure	All of the hardware, software, networks, facilities, etc., required to develop, test, deliver, monitor and control or support IT services. The term IT infrastructure includes all of the information technology but not the associated people, processes and documentation. (ITIL definition)
ISO/IEC 20000	International standard for IT service management. ISO/IEC 20000 is used to audit and certify service management capabilities.
IT Infrastructure Library (ITIL)	Set of best practice publications for IT service management. Published in a series of five core books representing the stages of the IT service lifecycle which are: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement.
IT Service	Service provided to one or more customers by an IT service provider. An IT service is made up of a combination of information technology, people and processes. (ITIL definition)
IT Service Continuity Management (ITSCM)	Process that supports the overall Business Continuity Management (BCM) process by ensuring that, by managing the risks that could seriously affect IT services, the IT service provider can always provide minimum agreed business continuity-related service levels. (ITIL definition)
IT Service Continuity Plan	Documented plan to ensure that existing and future continuity requirements for IT services can be provided cost effectively. (ITIL definition)
IT Service Management (ITSM)	Implementation and management of quality IT services that meet the needs of the business. (ITIL definition)
ITIL	See IT Infrastructure Library.
Kaizen	The practice of continuous improvement.
Kanban	Method of work that pulls the flow of work through a process at a manageable pace.
Kanban Board	Tool that helps teams organize, visualize and manage work.
Kata	Cultural conditioning. In Japanese business, Kata is the idea of doing things the “correct” way. An organization’s culture can be characterized as its Kata through its consistent role modeling, teaching and coaching. See also <i>Improvement Kata</i> and <i>Coaching Kata</i> .
Key Performance Indicator (KPI)	Key metric used to measure the achievement of critical success factors. KPIs underpin critical success factors and are measured as a percentage. (ITIL definition)
Knowledge Management	Process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision. Knowledge Management is responsible for gathering, analyzing, storing and sharing knowledge and information within an organization. (ITIL definition)
Known Error	Problem with a documented root cause and a workaround. (ITIL definition)
Known Error Database (KEDB)	Database containing all known error records. The KEDB is created by Problem Management and used by Incident and Problem Management. The KEDB is part of the Service Knowledge Management System. (ITIL definition)
Lean (adjective)	Spare, economical Lacking richness or abundance
Lean (production)	Production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value.

Term	Definition
Lean Enterprise	Organization that strategically applies the key ideas behind lean production across the enterprise.
Lean IT	Applying the key ideas behind lean production to the development and management of IT products and services.
Lean Manufacturing	Lean production philosophy derived mostly from the Toyota Production System.
Lean Six Sigma	Management approach that combines the concepts of Lean Manufacturing and Six Sigma by removing 'waste' and reducing 'defects.'
Major Incident	Incident that has a significant impact on the business.
Mean Time Between Failures (MTBF)	Average time that a CI or IT service can perform its agreed function without interruption. Often used to measure reliability. Measured from when the CI or service starts working, until the time it fails (uptime). (ITIL definition)
Mean Time Between Service Incidents (MTBSI)	Mean time from when a system or IT service fails until the next time it fails. Often used to measure reliability. (ITIL definition)
Mean Time to Repair (MTTR)	Average time required to repair a failed component or device. MTTR does not include the time required to recover or restore service.
Mean Time to Restore Service (MTRS)	Used to measure time from when the CI or IT service fails until it is fully restored and delivering its normal functionality (downtime). Often used to measure maintainability. (ITIL definition)
Metric	Something that is measured and reported upon to help manage a process, IT service or activity.
Model	Representation of a system, process, IT service, CI, etc. that is used to help understand or predict future behavior. In the context of processes, models represents pre-defined steps for handling specific types of transactions.
Muda	Waste.
Mura	Inconsistency or excess variation.
Muri	Overburden, unreasonableness or absurdity.
Non-functional requirements	Requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors or functions (e.g., availability, reliability, maintainability, supportability); qualities of a system.
Normal Change	Unique change...not standard or emergency. (ITIL definition)
Operational Level Agreement (OLA)	Agreement between an IT service provider and another part of the same organization. (ITIL definition)
Ops (in the context of DevOps)	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists and service desk analysts.
Organizational Culture	Behavior of humans within an organization and the meaning people attach to those behaviors.
Outcome	Term used to refer to intended results, as well as to actual results. (ITIL definition)
Patterns of Business Activity (PBAs)	Workload profile of one or more business activities. Business activities achieve outcomes and generate demand for services. Business activities are typically performed in patterns. (ITIL definition)

Term	Definition
Plan-Do-Check-Act (PDCA)	Four-stage cycle for process management, attributed to W. Edwards Deming. Also known as the Deming Cycle.
Poka-yoke	Fail-safing or mistake proofing.
Policies	Formal documents that define boundaries in terms of what the organization may or may not do as part of its operations.
Post-implementation Review (PIR)	A review that takes place after a change or a project has been implemented; determines if the change or project was successful, and identifies opportunities for improvement. (ITIL definition) See also retrospective.
Priority	The relative importance of an incident, problem or change; based on impact and urgency. (ITIL definition)
Problem	The underlying cause of one or more incidents. (ITIL definition)
Problem Management	Process that manages the lifecycle of all problems from first identification through further investigation, documentation and eventual removal. (ITIL definition)
Process	Structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs.
Process Control	Activity of planning and regulating a process, with the objective of performing the process in an effective, efficient and consistent manner.
Process Manager	Role responsible for the operational (day-to-day) management of a process. The Process Manager role may be assigned to the same person who carries out the Process Owner role, but the two roles may be separate in larger organizations. (ITIL definition)
Process Owner	Role accountable for the overall quality of a process. May be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations. (ITIL definition)
Process Practitioner	Role that is responsible for carrying out one or more process activities. (ITIL definition)
RACI Model/ Authority Matrix	Matrix that indicates roles and responsibilities in relation to processes and activities. RACI stands for responsible, accountable, consulted and informed.
Release	One or more changes that are built, tested and deployed together. (ITIL definition)
Release and Deployment Management	Process that plans, schedules and controls the build, test and deployment of releases and delivers new functionality while protecting the integrity of existing services. (ITIL definition)
Release Package	Set of CIs that will be built, tested and deployed together as a single release. (ITIL definition)
Release Policy	Formal document defined for one or more services that specifies unique identification, number and naming conventions, expected frequency, etc. Part of the overall Service Transition policy. (ITIL definition)
Reliability	Measure of how long a service, component or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI. (ITIL definition)
Remediation Plan	Plan that determines the actions to take after a failed change or release. (ITIL definition)
Request for Change (RFC)	Formal proposal to make a change. The term RFC is often misused to mean a change record, or the change itself. (ITIL definition)

Term	Definition
Request Fulfillment	Process that manages the lifecycle of service requests from users. (ITIL definition)
Return on Investment (ROI)	Difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage.
Risk	Possible event that could cause harm or loss, or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks and managing risks.
Role	Set of responsibilities, activities and authorities granted to a person or team. A role is defined by a process. One person or team may have multiple roles.
Root Cause Analysis (RCA)	Act of identifying the root (underlying, original) cause of an incident or problem. (ITIL definition)
SCRUM	Iterative and incremental Agile Framework for completing complex projects.
Security	See Information Security Management.
Security Management	See Information Security Management.
Service	Means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks. Services facilitate outcomes by increasing performance and reducing constraints. (ITIL definition)
Service Acceptance Criteria (SAC)	Set of criteria used to ensure that an IT service meets its functionality and quality requirements and that the IT service provider is ready to operate the new IT service. (ITIL definition)
Service Asset and Configuration Management (SACM)	Process that controls the assets required to deliver services and ensures accurate and reliable information is available. (ITIL definition)
Service Catalog	Subset of the Service Portfolio that consists of services that are live or available for deployment. Has two aspects: the Business/Customer Service Catalog (visible to customers) and the Technical/Supporting Service Catalog. (ITIL definition)
Service Catalog Management	Process that provides and maintains a single source of information for all operational (live) services and those being prepared for the live environment. Service Catalog Management is responsible for managing the Service Catalog. (ITIL definition)
Service Continuity Management	See IT Service Continuity Management.
Service Design	One of the ITIL Core publications and a stage of the service lifecycle. Service Design provides guidance that enables IT service providers to design IT services and the governing IT practices, processes and policies required to realize the service provider's strategy. (ITIL definition)
Service Design Package (SDP)	Document(s) defining all aspects of an IT service and its requirements through each stage of its lifecycle. The SDP is passed from Service Design to Service Transition. (ITIL definition)
Service Desk	Single point of contact (SPOC) between the service provider and the users. A typical Service Desk manages incidents and service requests and also handles communication with the users. (ITIL definition)

Term	Definition
Service Improvement Plan (SIP)	Formal plan to implement improvements to a process or IT service. (ITIL definition)
Service Knowledge Management System (SKMS)	Set of tools and databases used to manage knowledge and information. The SKMS includes the Service Portfolio, CMS, data, information and knowledge sources (e.g., DML, AMIS, CMIS, SCMIS, KEDB, CSI Register, plans, procedures, forums). (ITIL definition)
Service Level Agreement (SLA)	Written agreement between an IT service provider and its customer(s) that defines key service targets and responsibilities of both parties. An SLA may cover multiple services or customers. (ITIL definition)
Service Level Management (SLM)	Process that ensures all current and planned IT services are delivered to agreed achievable targets. (ITIL definition)
Service Level Requirement (SLR)	Customer requirement for an aspect of an IT service. SLRs are based on business objectives and are used to negotiate agreed upon service level targets. (ITIL definition)
Service Level Target	Commitment that is documented in a Service Level Agreement. (ITIL definition)
Service Lifecycle	Structure of the ITIL Core guidance. Represents the progressive stages in the life of an IT service including: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement. (ITIL definition)
Service Management	Set of specialized organizational capabilities for providing value to customers in the form of services. (ITIL definition)
Service Operation	One of the ITIL Core publications and a stage of the service lifecycle. Service Operation coordinates and carries out the activities and processes required to deliver and manage services at agreed levels to business users and customers. (ITIL definition)
Service Owner	Role that is accountable for a specific service throughout its lifecycle, regardless of where the underpinning technology, processes or capabilities reside. (ITIL definition)
Service Pipeline	Subset of the Service Portfolio that consists of services that are under consideration or in development. The Service Pipeline is not normally published to customers. (ITIL definition)
Service Portfolio	Complete set of services that are managed by a service provider. Used to manage the entire lifecycle of all services and includes three subsets: Service Pipeline, Service Catalog and Retired Services. (ITIL definition)
Service Portfolio Management (SPM)	Process that ensures a service provider has the right mix of services to balance the investment in IT with the ability to meet business outcomes. SPM is responsible for managing the Service Portfolio. (ITIL definition)
Service Provider	Organization that supplies services to one or more internal or external customers. (ITIL definition)
Service Request	User request for a standard service from an IT service provider. (ITIL definition)
Service Strategy	One of the ITIL Core publications and a stage of the service lifecycle. Service Strategy provides guidance that enables an IT service provider to define the perspective, position, plans and patterns that the service provider must execute to meet an organization's business outcomes. (ITIL definition)

Term	Definition
Service Transition	One of the ITIL Core publications and a stage of the service lifecycle. Service Transition ensures that new, modified, retired or transferred services meet the expectations of the business. (ITIL definition)
Service Validation and Testing	Process responsible for validation and testing of a new or changed IT service.
Seven-step Improvement Process	Process that defines and manages the steps needed to identify, define, gather, process, analyze, present and implement service improvements. (ITIL definition)
Six Sigma	Disciplined, data-driven approach that focuses on reducing defects by measuring standard deviations from an expected norm.
Stakeholder	Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers. (ITIL definition)
Standard Change	Pre-approved, low risk change that follows a procedure or work instruction. (ITIL definition)
Supplier	External (third party) supplier, manufacturer or vendor responsible for supplying goods or services that are required to deliver IT services. Suppliers may be categorized as strategic, tactical, operational or commodity. (ITIL definition)
Supplier Management	Process that manages suppliers and their services to ensure seamless IT service quality and obtain value for money. (ITIL definition)
The Three Ways	Key principles of DevOps – Flow, Feedback, Continuous experimentation and learning.
Theory of Constraints	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.
Time to Value	Measure of the time it takes for the business to realize value from a feature or service.
Tool chain	A philosophy that involves using an integrated set of complimentary task specific tools to automate an end to end process (vs. a single-vendor solution).
Transition Planning and Support	Process that provides overall planning for service transitions and coordinates the required resources. (ITIL definition)
Underpinning Contract (UC)	Contract between an IT service provider and an external, third-party supplier. Contracts with suppliers underpin Service Level Agreements.
Urgency	Measure of how long it will be until an incident, problem or change has a significant impact on the business. Impact and urgency are used to assign priority.
User	Consumer of IT services.
Utility	Functionality offered by a product or service to meet a particular need. Utility is defined in terms of the business outcomes that customers expect a service to support and the constraints it will remove. Utility is “what the service does” or <i>fitness for purpose</i> . (ITIL definition)
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.

Term	Definition
Velocity (Agile Software Development)	Measure of the quantity of work done in a pre-defined interval.
Vital Business Function (VBF)	Part of a business process that is critical to the success of the business. (ITIL definition)
Warranty	Promise or guarantee that a service will meet its agreed requirements including: availability, capacity, continuity and security. Warranty is “how it is delivered” or <i>fitness for use</i> . (ITIL definition)
Waste (Lean Manufacturing)	Any thing or process that does not add value to a product.
Waterfall (Software Development)	Linear and sequential approach to software design and development in which progress is seen as flowing steadily downwards (like a waterfall) through the software design and development lifecycle. Uses a phased approach, and workers move to next phase only when the previous phase is complete.
Work in Progress (WIP)	Any work that has been started but has not been completed.
Workaround	Temporary way to reduce or eliminate the impact of incidents or problems. May be logged as a known error in the Known Error Database. (ITIL definition)

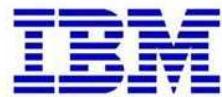
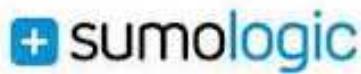
DEVOPS AND AUTOMATION

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Automation is an Essential Element

Automation enables agility, consistency, speed and reliability.



and many more...

Shared access to automated testing, deployment, monitoring and ITSM tools streamlines software delivery and prepares Ops for the long run.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Automation Enablers

- Treating infrastructure as code (i.e., programmatically provisioning and managing infrastructure resources)
- Repeatable and reliable deployment processes
 - Continuous integration, continuous delivery and continuous deployment
- Development and testing (preferably automated testing) performed against production-like systems
- On-demand creation of development, test, staging and production environments
- Proactive monitoring of infrastructure components, environments, systems and services

DevOps is not just about automation but there are common enabling practices.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Automation Benefits

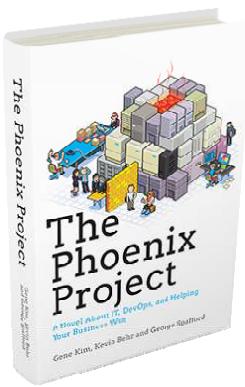
- Automation supports
 - Faster lead times
 - More frequent releases
 - Less turbulent releases
 - Fewer errors
 - Higher quality
 - Faster recovery
 - Business and customer satisfaction
- Automation gives rote tasks to computers and allows people to
 - Weigh evidence
 - Solve problems
 - Make decisions based on feedback
 - Use their skills, experience and judgment

DEVOPS PRINCIPLES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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

© DevOps Institute

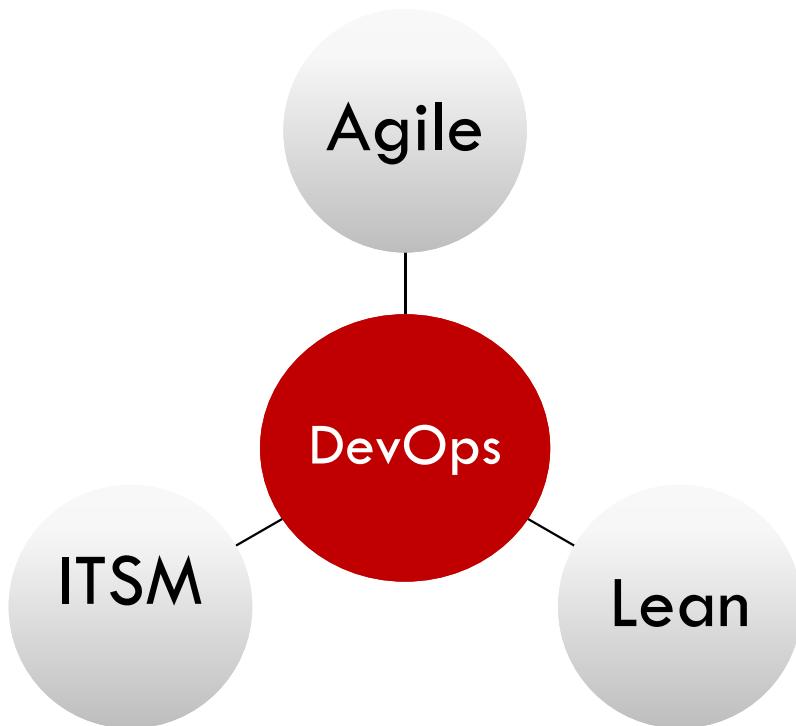
Copyrighted Material. Do Not Reproduce.

DEVOPS AND OTHER FRAMEWORKS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Cannot Stand Alone



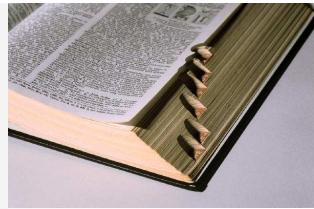
Successful DevOps relies on the adoption and integration of multiple frameworks and methodologies.

AGILE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Important Terms



- **Agile** (adjective)
 - Able to move quickly and easily; well-coordinated
 - Able to think and understand quickly; able to solve problems and have new ideas
- **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

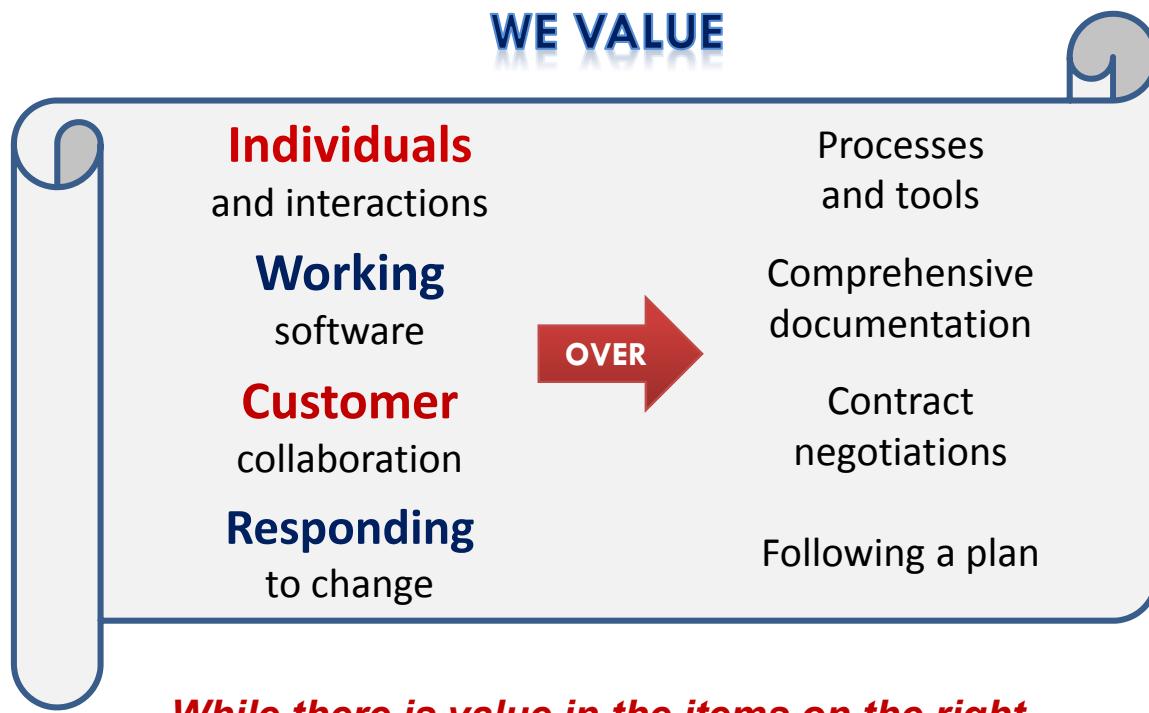
Agile software development methods deliver working software in smaller and more frequent increments.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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.*

Agile vs. Waterfall (1)

It's an ongoing debate... which methodology for managing software design and development projects is better, agile or waterfall?

Agile

- Iterative
- Incremental
- Decisions are made based on observation and experimentation rather than on detailed upfront planning

Waterfall

- Linear
- Sequential
- Phased approach
- Move to next phase only when previous phase is complete

Agile vs. Waterfall (2)

- Agile development is increasingly commonplace
- Waterfall is still used when
 - Requirements can be clearly defined upfront
 - Customer involvement is difficult
 - Factors such as time and budget are key considerations
 - Robust documentation is needed
- Some organizations adopt agile practices in only a sub-segment of the business in an effort to
 - Experiment and learn
 - Show proof of concept
 - Identify constraints and limitations

Popular agile software development methodologies include Extreme Programming (XP), Kanban, Lean Software Development, Scrum, and others.

Scrum Basics (1)

Scrum is an iterative and incremental agile framework for completing complex projects.

- **Impediment** – anything that prevents a team member from performing work as efficiently as possible
- **Product backlog** – requirements for a system, expressed as a prioritized list of product backlog items
 - Prioritized by the Product Owner
 - Includes functional, non-functional and technical team-generated requirements
- **Sprint**
 - A time-boxed iteration of work during which an increment of product functionality is implemented

While Scrum was originally intended for software development, it can been successfully applied to other types of complex projects.

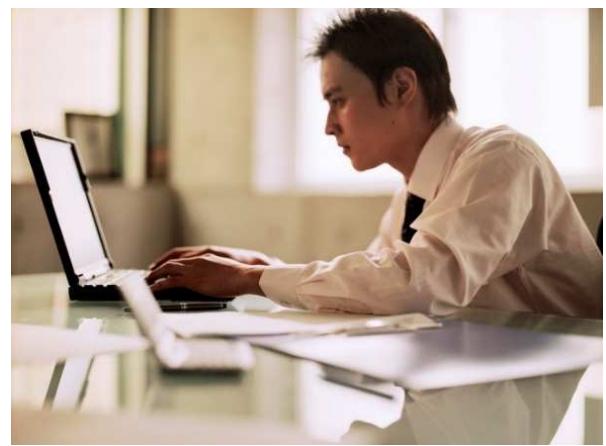
Scrum Basics (2)

Scrum Master



Provides process leadership
Removes impediments

Product Owner



Manages product backlog
Maximizes product value
Determines priorities/what to do

Scrum Basics (3)

- **Scrum Team**

- a self-organizing, cross-functional team optimally comprised of seven plus or minus two people



A self-organizing team

- Decides how best to allocate its resources to take advantage of each team member's strengths
- Has the ability and authority to make decisions and adapt to changing demands

Self-organizing teams decide how to do their work.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Scrum Basics (4)

- **Daily Scrum meeting –**
 - A fifteen-minute daily meeting for each team member to answer three questions
 - What have I done since the last Scrum meeting?
 - What will I do before the next Scrum meeting?
 - What impediments prevent me from performing my work as efficiently as possible?



Daily Scrum meetings are often ‘Standup’ meetings.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

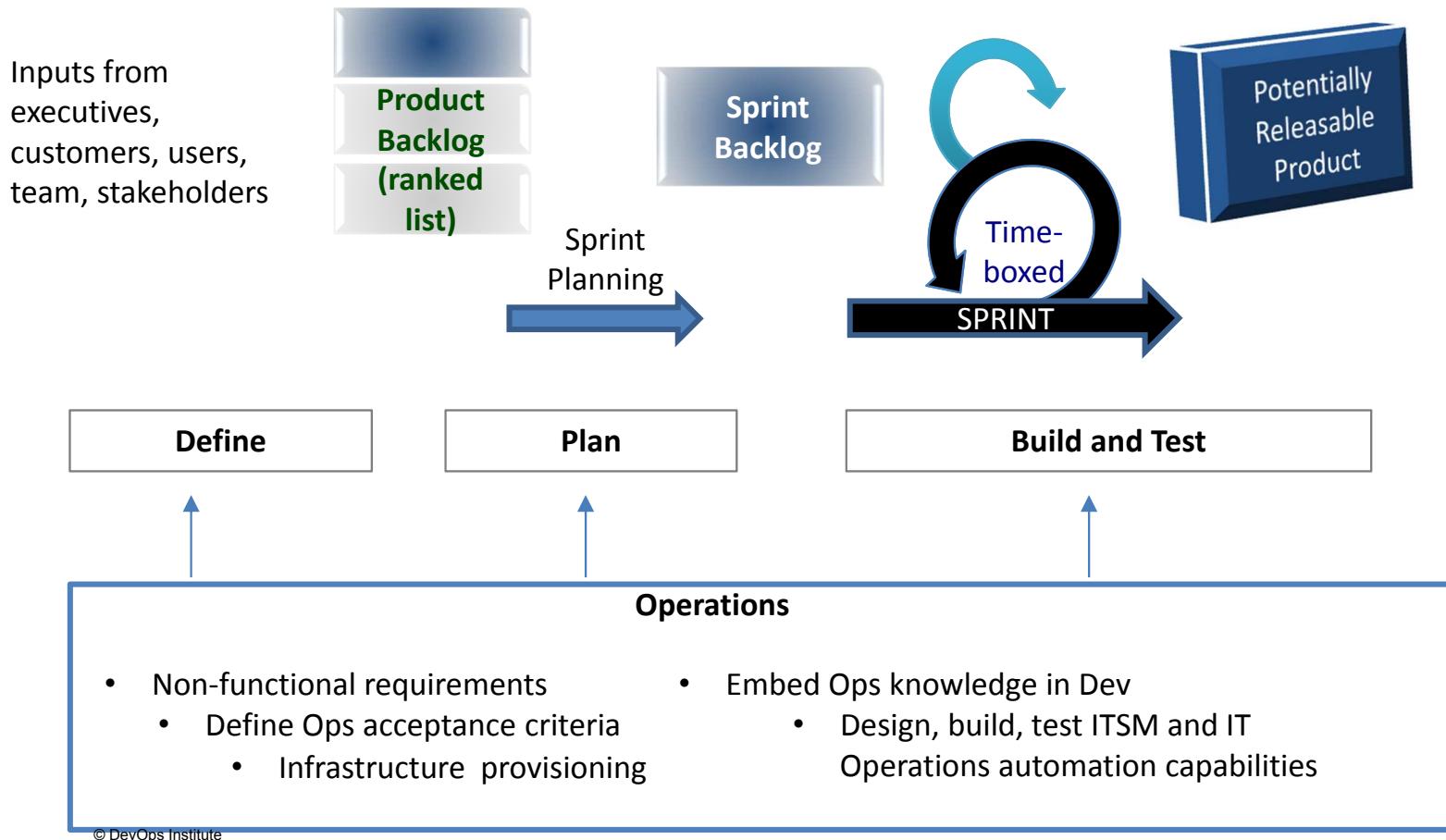
DEVOPS AND AGILE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Ops Involvement in Agile

DevOps and agile complement each other to deploy working functionality into production faster.



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.

© DevOps Institute

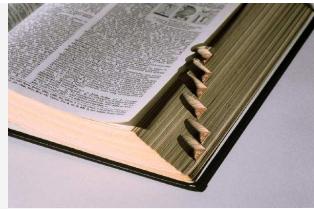
Copyrighted Material. Do Not Reproduce.

LEAN

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Important Terms (1)



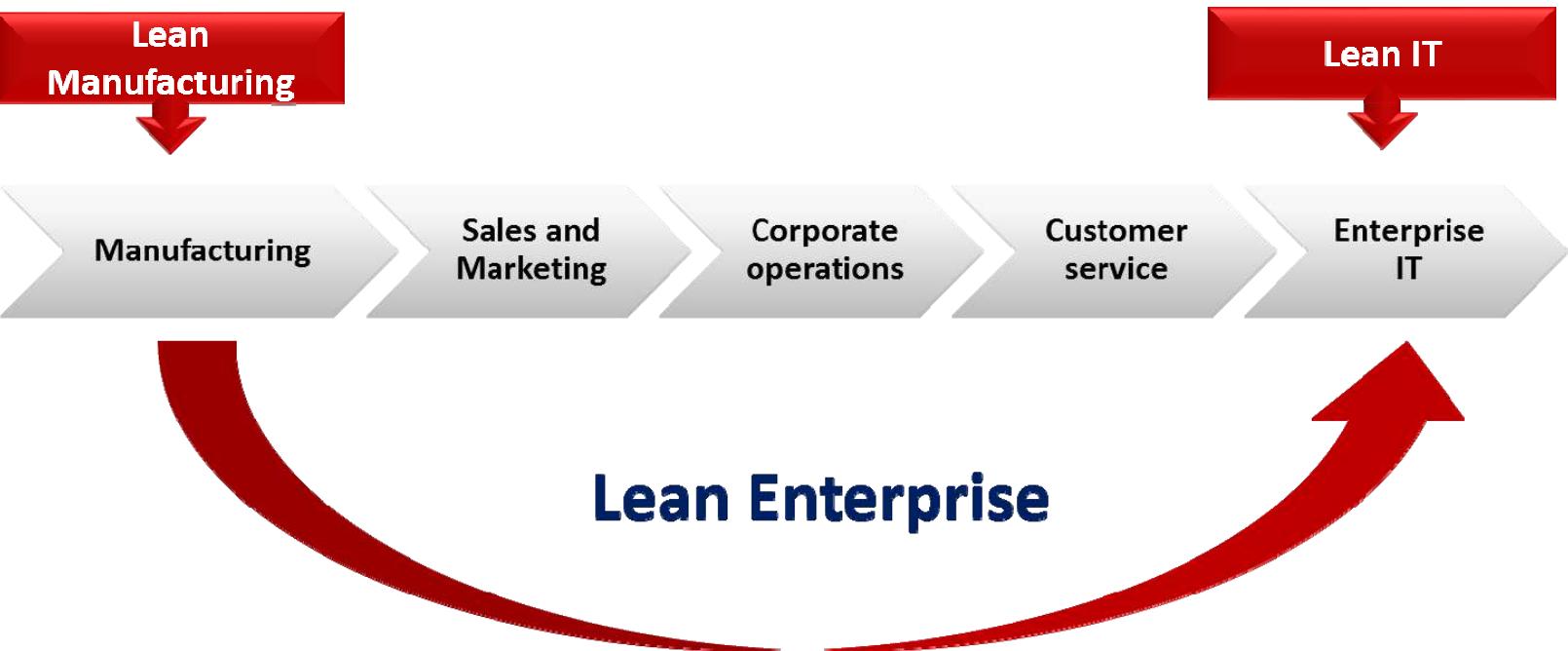
- **Lean** (adjective)
 - Spare, economical
 - Lacking richness or abundance
- **Lean** (production) – a production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value
- **Flow** – (also known as single-piece flow) describes how people or products 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

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

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Lean Perspectives



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

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Lean Tools

Lean tools enable dramatic improvements. Culture change is required, however, to avoid slipping back to old habits.

Commonly used tools include

- A3 thinking (problem solving)
- Continuous flow (eliminates waste)
- Kaizen (continuous improvement)
- Kanban (pull system)
- KPI (key performance indicator)
- Plan, do, check, act
- Root cause analysis
- SMART goals (specific, measurable, achievable, relevant, time-bound)
- Value stream mapping (depict flow)

Five Principles of Lean Thinking

Lean enterprises embrace the five principles of lean thinking.

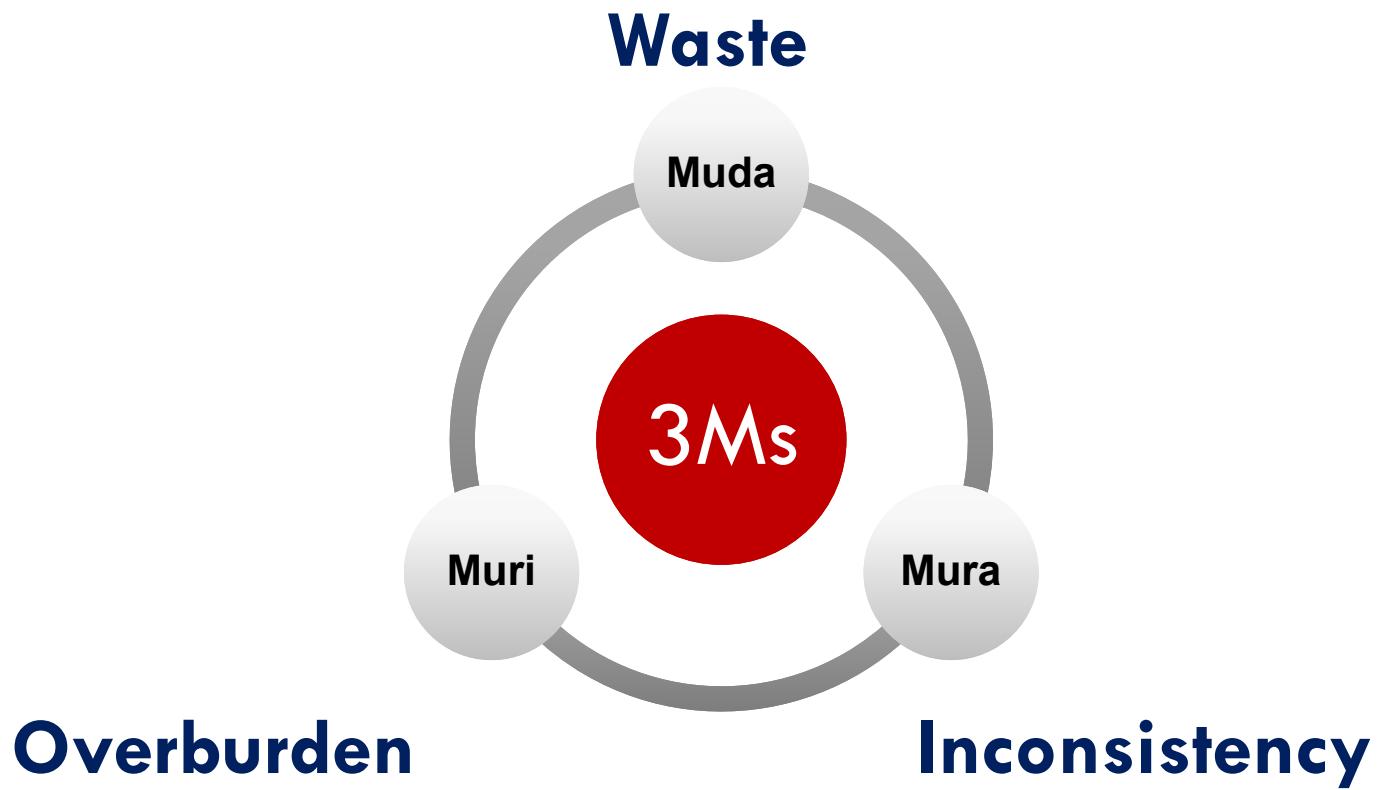
1. Define value precisely from the perspective of the end customer
2. Identify the entire value stream for each service, product or product family and eliminate waste
3. Make the remaining value-creating steps flow
4. As flow is introduced, let the customer pull (i.e., provide what the customer wants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

“Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.”

Antoine de Saint-Exupery

The 3Ms of Lean



Both mura and muri cause muda (waste).

Sources of Waste

The goal of lean thinking is to create more value for customers with fewer resources and less waste.

- **Defects** – deviations from requirements
- **Overproduction** – producing more or faster than required
- **Inventory** – carrying excess raw materials, work in progress (WIP) or finished goods
- **Over-processing** – doing more work than is required
- **Motion** – moving people or equipment more often than is required
- **Transportation** – moving products from one location to another
- **Waiting** – doing nothing or moving slowly while waiting on a previous step

Waste is any activity that does not add value to the process.

DEVOPS AND LEAN

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps and Lean

Improving the flow of work between Dev and Ops will remove many types of waste.

Exercise:

Where is the waste?

1. Defects
2. Overproduction
3. Inventory
4. Over-processing
5. Motion
6. Transportation
7. Waiting

IT SERVICE MANAGEMENT

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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 management (ITSM)** – the implementation and management of quality IT services that meet the needs of the business
- **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*
Copyright © AXELOS Limited 2011.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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

DevOps practices support ITIL process improvement.

AXELOS®, ITIL® and IT Infrastructure Library® are registered trade marks of AXELOS Limited.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

“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

DEVOPS AND IT SERVICE MANAGEMENT

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

IT Service Management



- DevOps does not stop at deployment and early life support
- ITSM does not start at deployment and early life support
- ITSM processes are essential to ongoing operations, continual improvement and value creation
- DevOps needs ITSM best practices to meet the goal of deploying faster changes without causing disruption
- The integration of DevOps and ITSM helps to identify, reduce and eliminate ongoing risks and constraints

Repeatable service management processes can lead the way to stable continuous delivery and increased flow.

ITSM Processes Enable DevOps

Key ITSM processes that help enable DevOps include

- Change Management
- Release and Deployment Management
- Service Asset and Configuration Management
- Event Management
- Incident Management
- Problem Management
- Knowledge Management

All ITSM processes will ultimately play a role in supporting DevOps by increasing flow, reducing constraints and creating business value.



DEVOPS CULTURE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Understanding Organizational Culture

Organizational culture is the behavior of humans within an organization and the meaning people attach to those behaviors.

- Improvement efforts must consider the organization's current culture
- Culture can
 - Enable continuous improvement
 - Be the source of resistance



Simply put, culture is ‘the way we do things.’

Characteristics of Organizational Culture

Culture includes the organization's vision, values, norms, systems, symbols, language, assumptions, beliefs and habits.

- Culture is shown in
 - The way an organization conducts business
 - How an organization treats its employees, customers and the wider community
 - The freedoms allowed in decision making, developing new ideas and personal expression
 - How power and information flow through the organization's hierarchy
 - How committed employees are towards collective objectives

Cultural Debt

- Occurs when cultural considerations are disregarded or deferred in favor of growth and innovation
- Hinders IT's ability to progress because of
 - Extensive silos
 - Workflow constraints
 - Insufficient collaboration
 - Miscommunications
 - Project delays
 - Waste
 - Additional cost

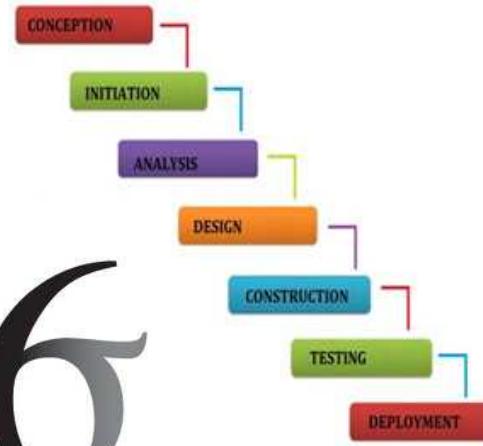
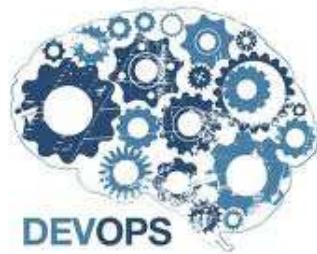


DevOps pays down cultural debt by focusing on creating a trust relationship that includes human and technical interactions.

“Culture eats strategy for breakfast.”

Peter Drucker

IT is a Multicultural Society...



© DevOps Institute

Copyrighted Material. Do Not Reproduce. ITIL® is a registered trade mark of AXELOS Limited.

...Which Results in a Lot of Conflicting Messages

DevOps lacks a standard of practice

ITIL is too commoditized

More tools!
Better tools!

Agile is only for developing software

IT Guy



“Tools and processes are a reflection of your cultural choices.”

Sascha Bates

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
 - 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

- IT focus (inside-out)
- Silos
- Command and control
- Task-oriented
- Blame
- Reactive
- Content
- Resistant
- Low trust

To

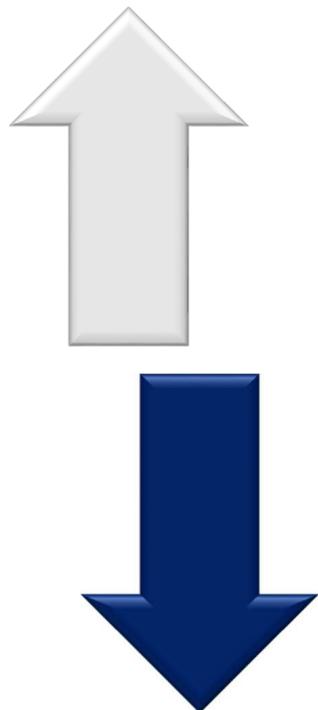
- Customer focus (outside-in)
- Cross-functional teams
- Collaborative
- Outcome-oriented
- Responsibility
- Proactive
- Courageous
- Flexible
- High trust

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

High Trust vs. Low Trust

High Trust

Low Trust



Speed

Cost

Culture and the Flow of Information

Pathological	Bureaucratic	Generative
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

Source: Westrum, A Typology of Organizational Cultures

High-trust organizations encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas; they are also the most likely to perform at a high level.

© DevOps Institute

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

Culture and Learning

To make learning a part of the organization's culture

- Incorporate learning into processes
- Encourage daily learning and knowledge sharing
- Use technology to accelerate learning
- Make work educational
 - Experimentation
 - Problem solving
 - Demonstrations
- Allow and use mistakes as sources of learning
- Make the results of learning visible



To create a learning culture, leaders and managers must value learning and model the behavior.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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

Jack Welsh

MANAGING CULTURE CHANGE

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps is All About People



- The way they think
- The way they interact
- The way they behave
- The way they work
- The way they perceive themselves and their colleagues

People will adapt to the values and practices of your organizational culture.

Culture Change is Never Easy



People typically don't resist their own ideas.

- 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

Source: *The ITSM Process Design Guide*

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



Buckminster Fuller

Role of Early Adopters

Willingness to Change	Management Action
13.5% of workers are early adopters and about 2.5% of early adopters are innovators who initiate change	Engage forward thinkers and empower them to serve as change agents
68% of workers will change with encouragement and proof that the change is worth the pain and effort	Provide a clear vision, training and proof of concept
16% of workers are naysayers who will view the change negatively and may never accept the change	Focus on the upper 84%

Source: Rogers. *Diffusion of Innovations*

- Engage early adopters in change planning activities
- Assign them responsibility for some aspects of the plan
- Give them a reason to get involved and get excited

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Empowering New Behaviors

- Improve communication and collaboration practices and shared tools
- Create a common vocabulary
- Job shadowing
- Cross-skilling
- Team building
- Communities of practice
- Internal DevOps Days
- Game days (hackathons)
- 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.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Management Responsibilities

- Management cannot expect people to change without management commitment in the form of resources, training and support
- Management support requires soft skills that individual managers may need to hone through a training program



Required Management Skills	
Coaching	Dealing with conflict
Team building	Negotiation
Listening	Delegation

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Source: *The ITSM Process Design Guide*

Managing 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.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

“You can’t directly change culture. But you can change behavior, and behavior becomes culture.”

Lloyd Taylor

ORGANIZATIONAL CONSIDERATIONS

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Skills (1)

The demand for DevOps skills is rapidly rising.



Desired skills	%
Coding or scripting	84%
People skills – communication and collaboration	60%
Process reengineering (e.g., using agile and lean practices)	56%
Experience with specific tools	19%

© DevOps Institute

Source: 2013 State of DevOps Report – Puppet Labs and IT Revolution Press

Copyrighted Material. Do Not Reproduce.

DevOps Skills (2)

- Sought after capabilities also include a more business- and customer-oriented perspective (vs. just a technology perspective)
 - Knowledge of business priorities
 - Familiarity with business processes



DevOps Teams (1)

- There has been a dramatic increase in the creation of DevOps departments or teams that
 - Have Dev and Ops report to same management
 - Build on an agile small team approach
 - Are successfully applying DevOps principles
- The pros and cons of this approach are not clear
 - Are they doing anything noticeably different than other IT departments?

16% of 2014 survey respondents had or were creating DevOps departments. These were mostly departments with 20-499 employees but the trend is growing.

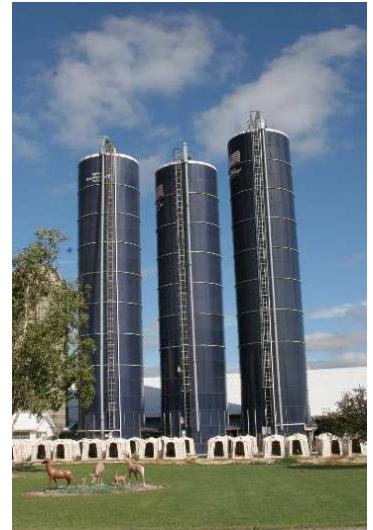
Source: 2014 State of DevOps Report – Puppet Labs, IT Revolution Press and ThoughtWorks

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

DevOps Teams (2)

- **Downsides of DevOps Teams**
 - Yet another silo
 - Dev and Ops wash their hands of accountability

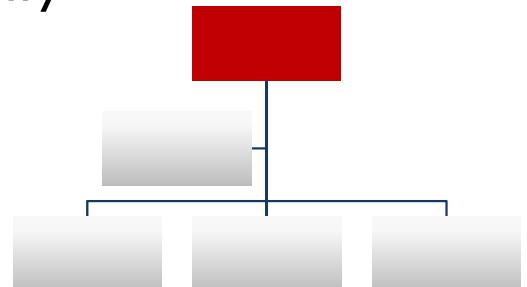


DevOps activities become ‘someone else’s problem.’

DevOps Organizational Structures

Some organizations are

- Assigning Ops liaisons to each Dev team
- Creating cross-functional product (vs. project) teams to facilitate greater collaboration
- Adopting matrix or market-oriented structures (vs. function-oriented structures)
- Creating shared services within Ops that Dev teams can use to improve productivity



What is a DevOps Engineer?

- There is currently no ‘industry recognized’ job description or formal career track for a DevOps Engineer
- Individuals filling these roles may be
 - Developers interested in deployment
 - System Administrators who enjoy scripting and coding
- 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

WIIFY?

What's in it for you?



- Members of teams that adopt a DevOps culture
 - Lead a more balanced life
 - Spend more time on automation and infrastructure improvement
 - Spend less time fighting fires
- DevOps oriented teams spend more time doing positive or neutral tasks
 - Automating repetitive tasks
 - Infrastructure improvements
 - Self education,

Fire fighters are important, but as quality improves through the effective use of DevOps practices, there will be fewer fires to fight and more time to work on improvements.

© DevOps Institute

Source: 2013 IT Ops & DevOps Productivity Report – RebelLabs

Copyrighted Material. Do Not Reproduce.

Addressing the IT Skills Gap

- Organizations may lack competency with new tools, practices and methods
- The talent to fill needed roles may be
 - Within enterprise IT
 - Within shadow IT groups
 - Outside the organization
- Some organizations are
 - Creating teams of enthusiasts to build capabilities
 - Restructuring their pay and corporate culture to attract and retain technical talent
 - Working with local universities

Organizations are addressing the IT skills gap by promoting collaboration and improved communication.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

PROMOTING COLLABORATION

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Improving Collaboration Skills

Behavior and attitude changes can enable people to improve their collaboration skills.

Exercise:

- Describe the characteristics of good and poor collaborators



© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Characteristics of Collaboration

Collaboration involves people jointly working with others towards a common goal.

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

In a collaborative environment, each person's contribution is valued.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

“The traditional model is that you take your software to the wall that separates development and operations, and throw it over and forget about it. Not at Amazon. You build it, you run it. This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer.”

Werner Vogels

Creating and Maintaining a Collaborative Environment

- Establish systems for providing feedback and holding members accountable for their role
- Encourage tolerance and a willingness to accept others' differences
- Construct an atmosphere of trust
- Discuss conflicts openly and develop the ability to compromise
- Use consensus decision making
- Provide opportunities for members to talk and meet (when possible) in more relaxed, informal environments
- Communicate frequently
- Develop effective communication systems

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Cross-functional teams

- DevOps teams often align around products or services
- Team members might specialize in development, business analysis, database administration, operations, etc.
- Limiting teams to seven (plus or minus two) people enables collaboration and more effective communication



© DevOps Institute



Copyrighted Material. Do Not Reproduce.

IMPROVING COMMUNICATION

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Effects of Poor Communication

- Wasted time
- Wasted effort
- Mistrust
- Misunderstandings
- Unmet goals
- Stagnation



DevOps and Communication

DevOps-oriented teams

- Spend 2 fewer hours communicating each week (e.g., meetings, writing emails, planning)
- Communicate more effectively
 - Improved collaboration
 - Shared tooling (e.g., group chats, task managers and social tools)

Source: 2013 IT Ops & DevOps Productivity Report – RebelLabs

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Building Effective Communication Systems

- Decide how communication will occur
- Establish
 - Rules of engagement for meetings
 - Appropriate uses for communication channels
 - Phone, email, social tools
- Improve and automate (where possible) the flow of information
- Encourage sharing
 - Ideas, problems, feedback

“Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work.”

Steve Jobs

DEVOPS AUTOMATION PRACTICES AND TOOL CATEGORIES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

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 an end-to-end process.

Where to begin?

- Simplify first – don't automate bad processes
- Automate high value and repetitive tasks
- Automate error-prone work
- Automate to optimize workflow bottlenecks and communication flows
- Improve automated monitoring and notification practices

Make it easy for people to do the right thing!

“Your tools alone will not make you successful.”

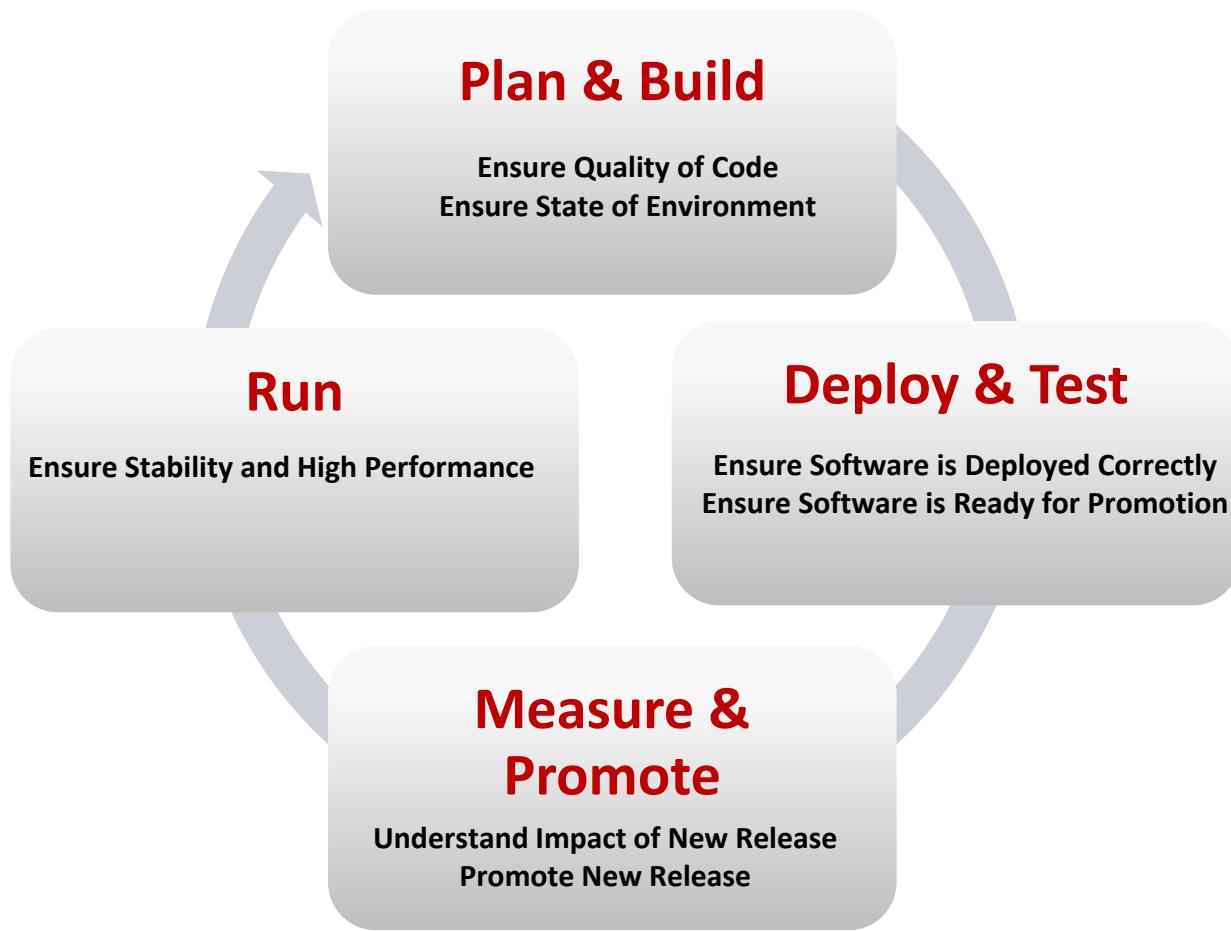
Patrick Debois

APPLICATION LIFE CONCERNS AND TOOL CATEGORIES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Application Life Concerns



Plan & Build

- Key Concerns:
 - Ensure Quality of Code
 - Ensure State of Environment

	How to	Tools
Ensure Quality of Code	<ol style="list-style-type: none">1. Ensure a “single source of truth” for all application code2. Run automated integration and functional tests for each code check-in	<ul style="list-style-type: none">• Version (Source) Control• Test Automation
Ensure State of Environment	<ol style="list-style-type: none">1. Automate infrastructure and environment configuration and compliance checks2. Leverage infrastructure monitoring to ensure network, storage and the various underlying components are stable	<ul style="list-style-type: none">• Configuration Management• Infrastructure Monitoring

Version control tools enable change control and tracking for all production artifacts.

Deploy & Test

- Key Concerns:
 - Ensure Software is Deployed Correctly
 - Ensure Software is Ready for Promotion

	How to	Tools
Ensure Software is Deployed Correctly	<ol style="list-style-type: none">1. Automate the entire deployment process2. Run automated post-deployment compliance checks	<ul style="list-style-type: none">• Release Automation• Configuration Management
Ensure Software is Ready for Promotion	<ol style="list-style-type: none">1. Run automated/synthetic tests to simulate user experience2. Perform automated functional, user experience and user acceptance testing to ensure all existing features are up and running and any new features function as expected	<ul style="list-style-type: none">• Synthetic Monitoring• Test Automation• Application Performance Management

Configuration management tools are used to standardize resource configurations and enforce their state across the IT infrastructure.

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Measure & Promote

- Key Concerns:
 - Understand Impact of New Release
 - Promote New Release

	How to	Tools
Understand Impact of New Release	<ol style="list-style-type: none">1. Measure performance by executing synthetic transactions and gathering the results2. Measure real end-user experience by monitoring page load times or how they interact with the software3. Monitor performance of application at code level4. Monitor impact at infrastructure level	<ul style="list-style-type: none">• Synthetic Monitoring• Real User Monitoring• Application Performance Management• Infrastructure Monitoring
Promote New Release	<ol style="list-style-type: none">1. Promote application to next level of the software release lifecycle (e.g., staging or production) and make it visible to customers	<ul style="list-style-type: none">• Release Automation

Run

Key Concerns:

Ensure Stability and High Performance

	How to	Tools
Ensure Stability and High Performance	<ol style="list-style-type: none">1. Continuously measure performance by executing synthetic transactions and gathering results2. Ensure infrastructure configuration remains the same by running automated compliance checks3. Monitor infrastructure performance and configure automated alerts for specific problems4. Predict performance impact and quickly triangulate root-cause by leveraging data from various monitoring sources including application code, infrastructure and end user experience5. Automate remediation by building event driven problem resolution (self-healing)	<ul style="list-style-type: none">• Synthetic Monitoring• Configuration Management• Infrastructure Monitoring• Real User Monitoring• Application Performance Management• Proactive Analytics Engine• Event Management Platform

OTHER TOOL CATEGORIES

© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Communication and Collaboration

Facilitate Communication and Collaboration	How to	Tools
	How to	Tools
	<ol style="list-style-type: none">1. Provide at a glance status updates2. Improve workflow3. Improve information flow4. Enable virtual collaboration5. Enable cross-functional cross-skilling and job sharing	<ul style="list-style-type: none">• Dashboards• Kanban boards• Group chat rooms (ChatOps)• Workflow and project management tools• Document sharing• Wikis• Knowledge management systems• Socials tools• Shared backlog

IT Service Management

	How to	Tools
Manage lifecycle of IT Services	<ol style="list-style-type: none">1. Support ITSM processes2. Enable process integration3. Improve workflow4. Improve information flow5. Improve process efficiency and effectiveness	<ul style="list-style-type: none">• Enterprise-wide suites• Process and task specific tools• Service Portfolio• Service Catalog• Service Knowledge Management System• Configuration Management System

Additional Sources of Information (1)

Websites

- www.devopsinstitute.com
- www.devops.com
- www.devopsconnect.com
- www.devopsdays.org
- www.itrevolution.com
- www.itsmprofessor.net
- www.puppetlabs.com
- www.devopsentreprise.io



© DevOps Institute

Copyrighted Material. Do Not Reproduce.

Additional Sources of Information (2)

Published Works

- *2013-2015 State of DevOps Report.* Puppet Labs, IT Revolution Press
- *The Phoenix Project.* G. Kim, et al, IT Revolution Press, 2013
- *7 Habits of Successful DevOps.* Forrester Research, 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

DevOps best practices will continue to evolve through communities of practice. Seek out opportunities to collaborate with others.

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?*

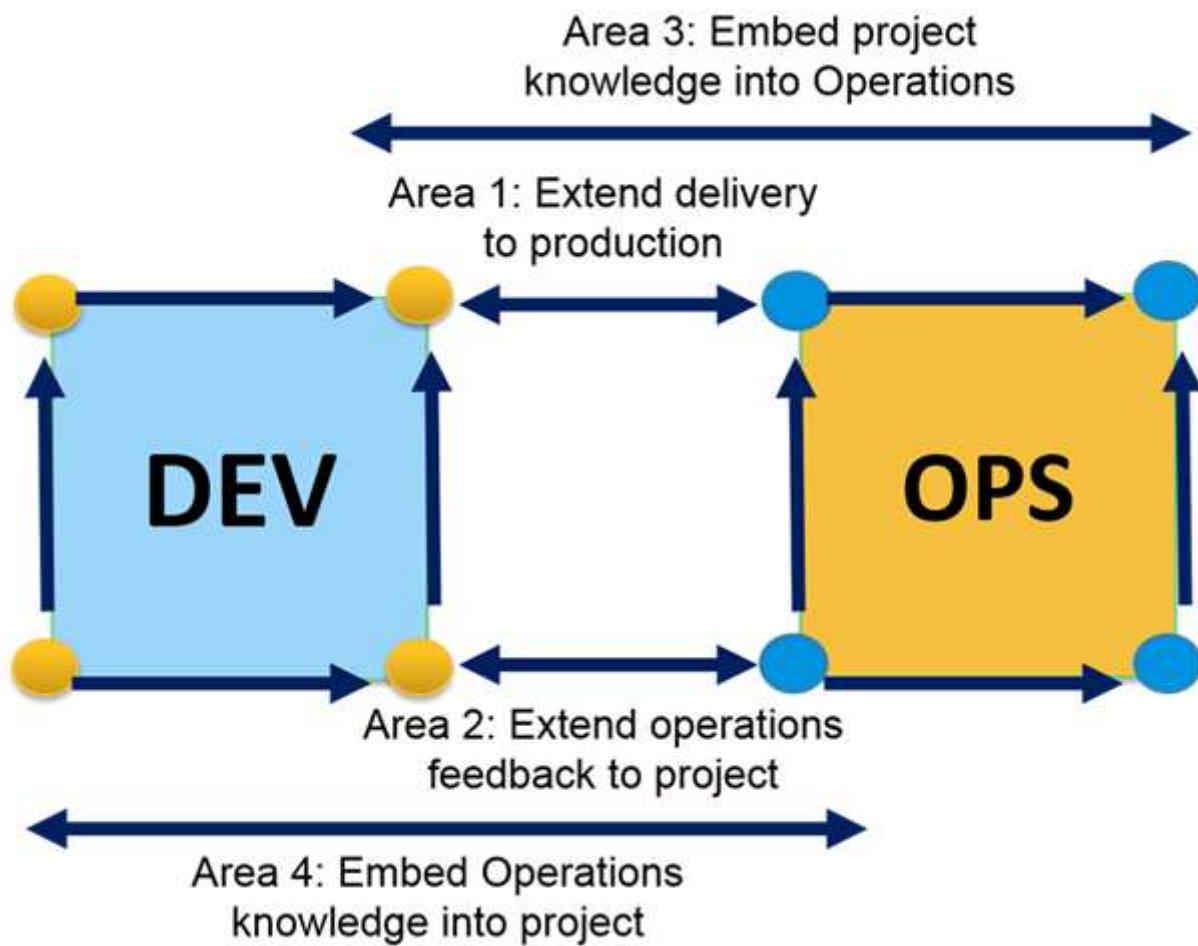
DevOps principles and practices affect organizational culture.

Culture change and continuous improvement cannot happen without the support of people like you.

Take action!

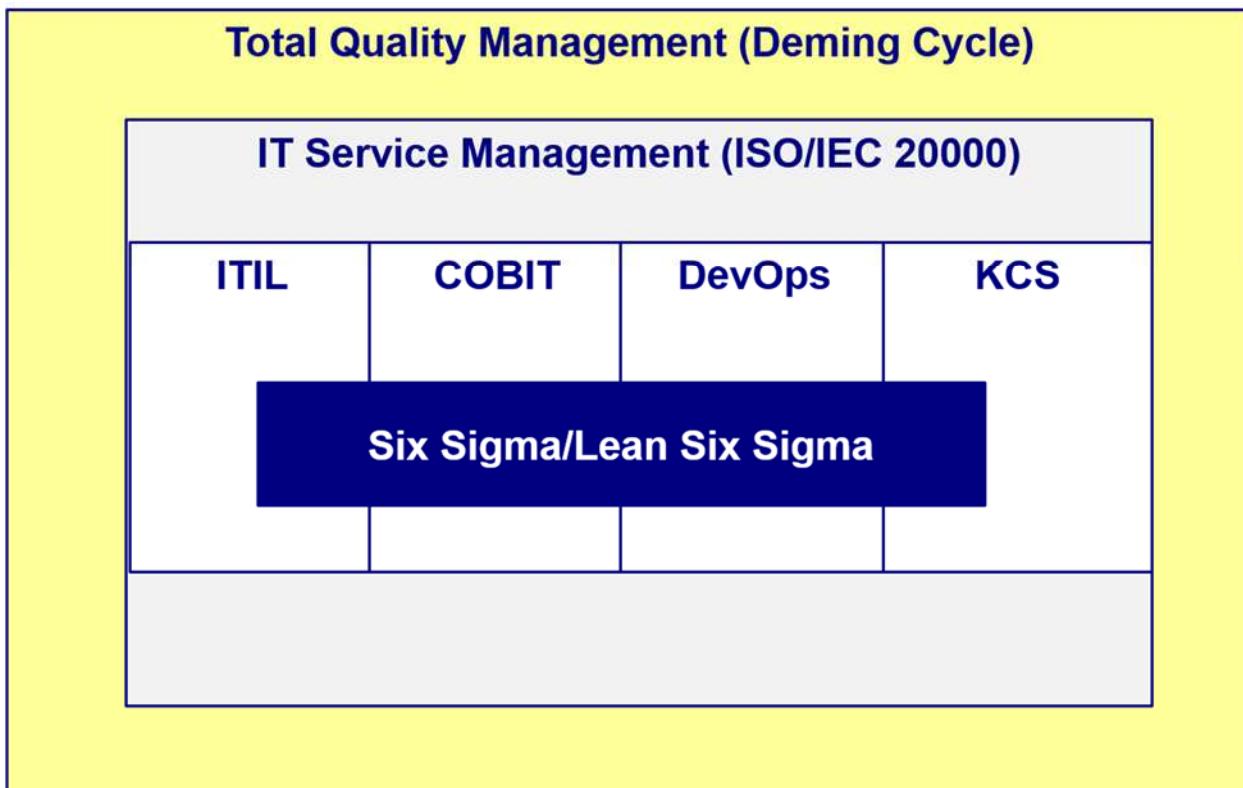
Sample Diagrams

SD 1 – DevOps Patterns



Excerpted from *Why Everyone Needs DevOps Now*. Reprinted with permission from Gene Kim.

SD 2 – DevOps and Other Frameworks



Based on *The ITSM Process Design Guide: Developing, Reengineering, and Improving Service Management* material. Copyright 2014 ITSM Academy.

SD 3 – Culture and the Flow of Information

Pathological	Bureaucratic	Generative
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

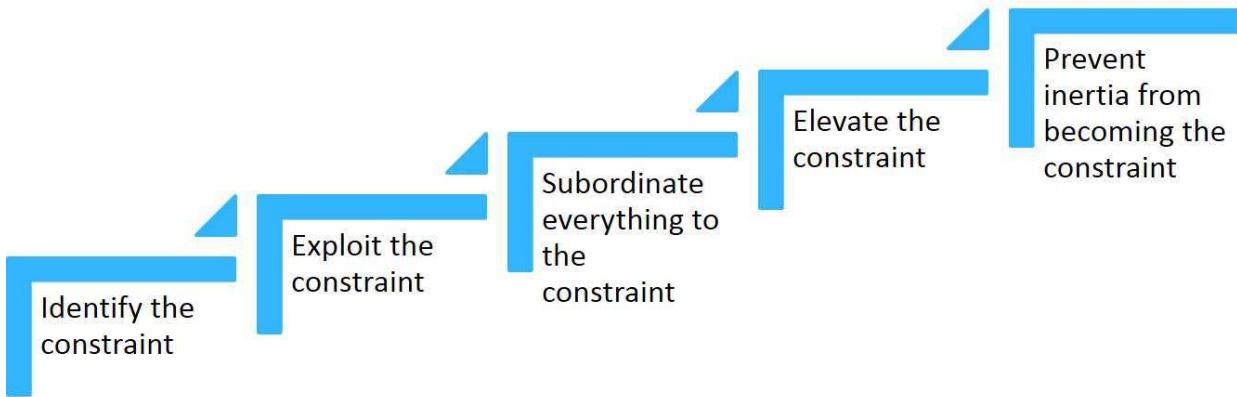
Source: Westrum, *A Typology of Organizational Cultures*

SD 4 – Sample Kanban Board



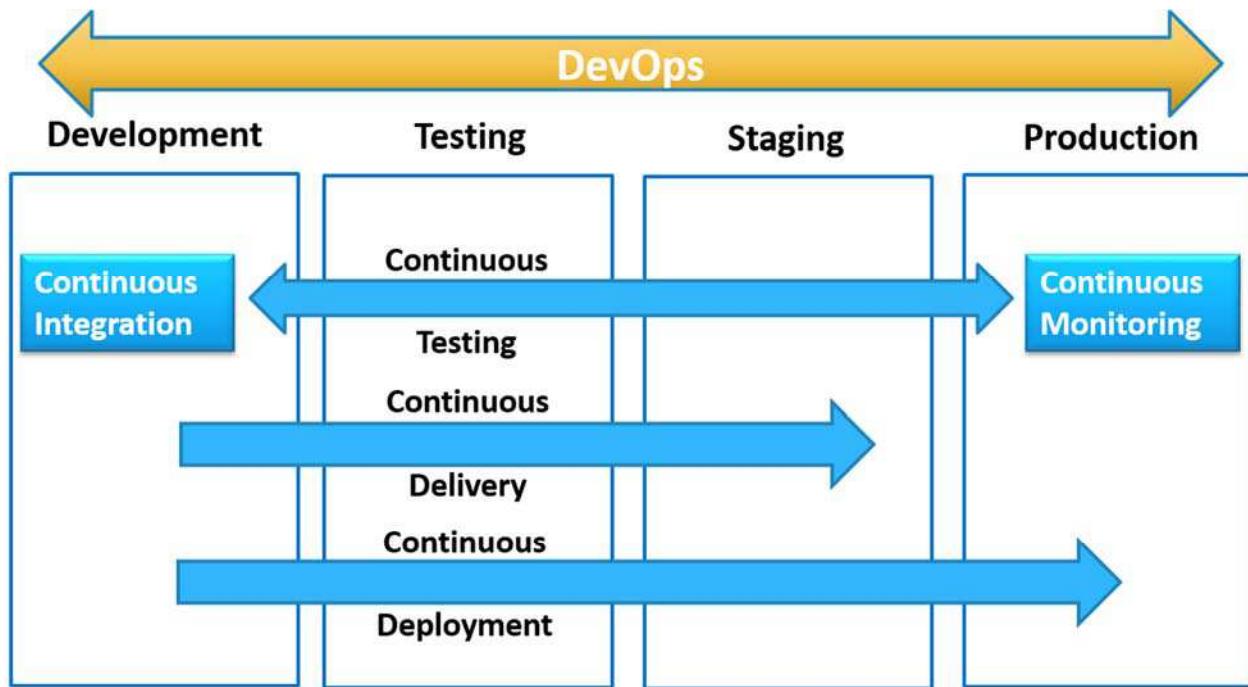
Excerpted from the presentation *IT at Ludicrous Speeds: Rugged DevOps and More*. Reprinted with permission from Gene Kim.

SD 5 – Theory of Constraints - Five Focusing Steps



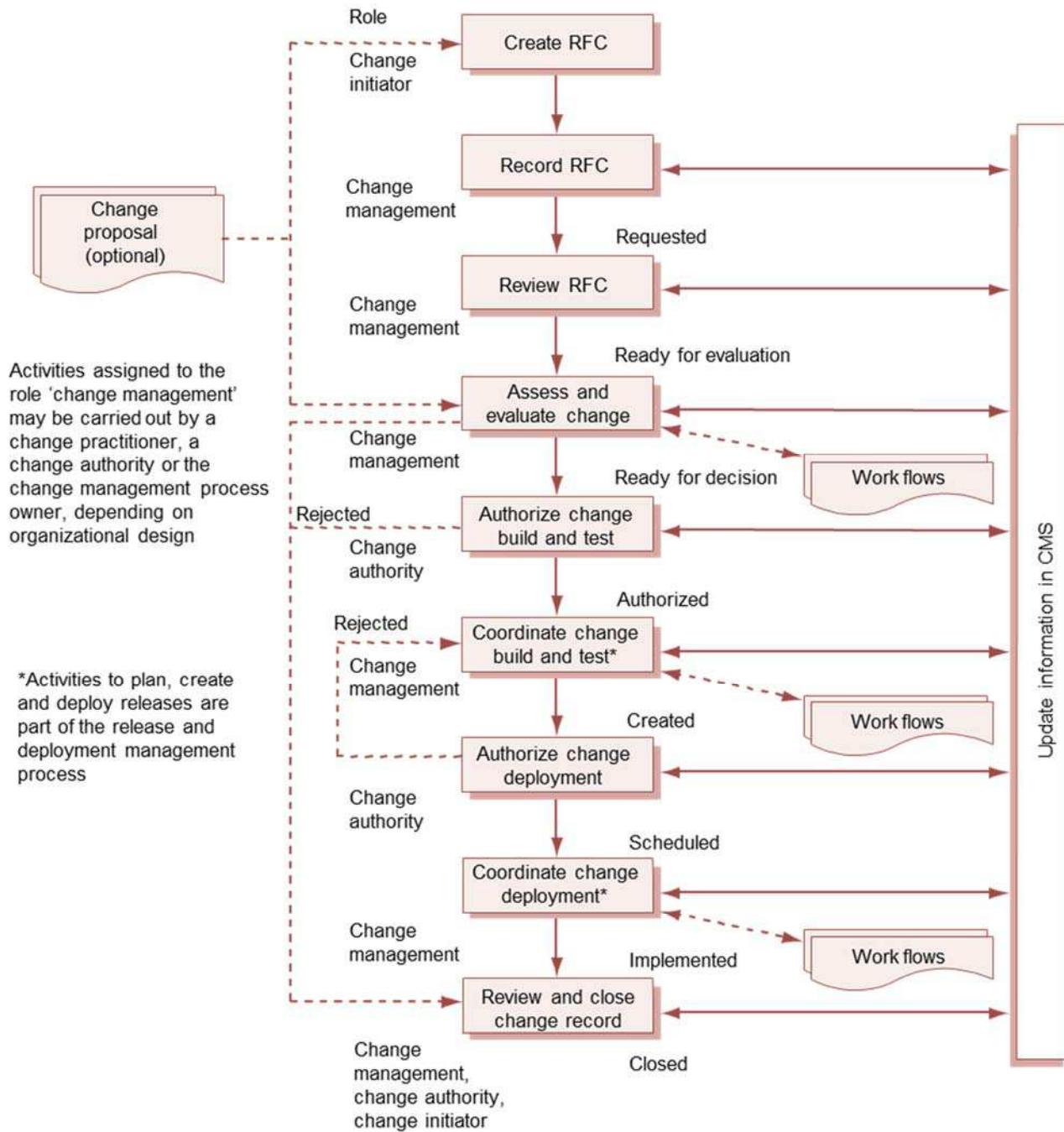
Source: Goldratt, *Theory of Constraints*

SD 6 – Continuous Delivery



*Continuous delivery requires collaboration
between Dev and Ops.*

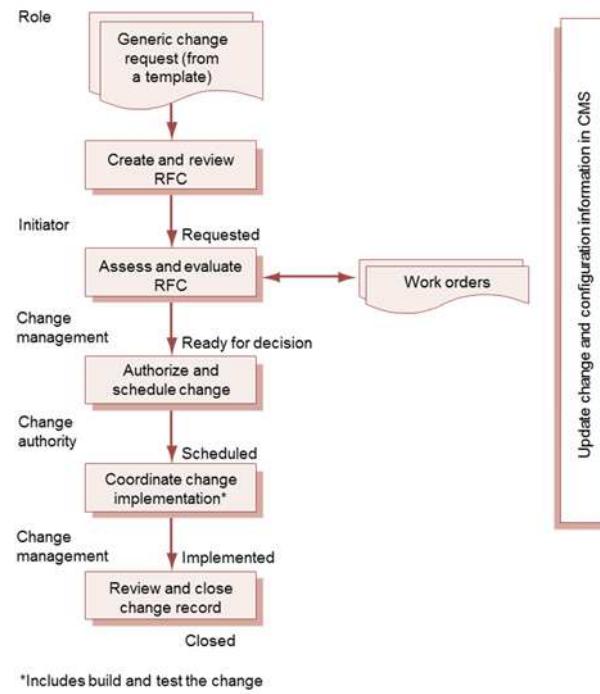
SD 7 - Normal Change Management Process Flow



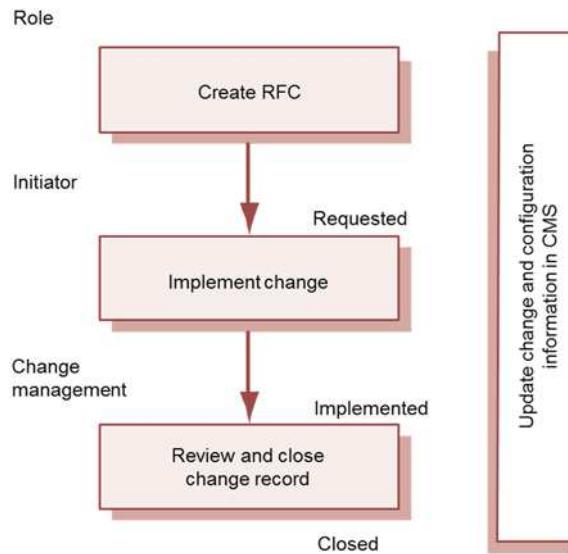
Based on AXELOS ITIL® material. Reproduced under license from AXELOS. (Figure 4.2 – Example of a process flow for a normal change - *ITIL® Service Transition 2011 edition*)

SD 8 - Sample Standard Change Management Process Flows

Standard Deployment Request

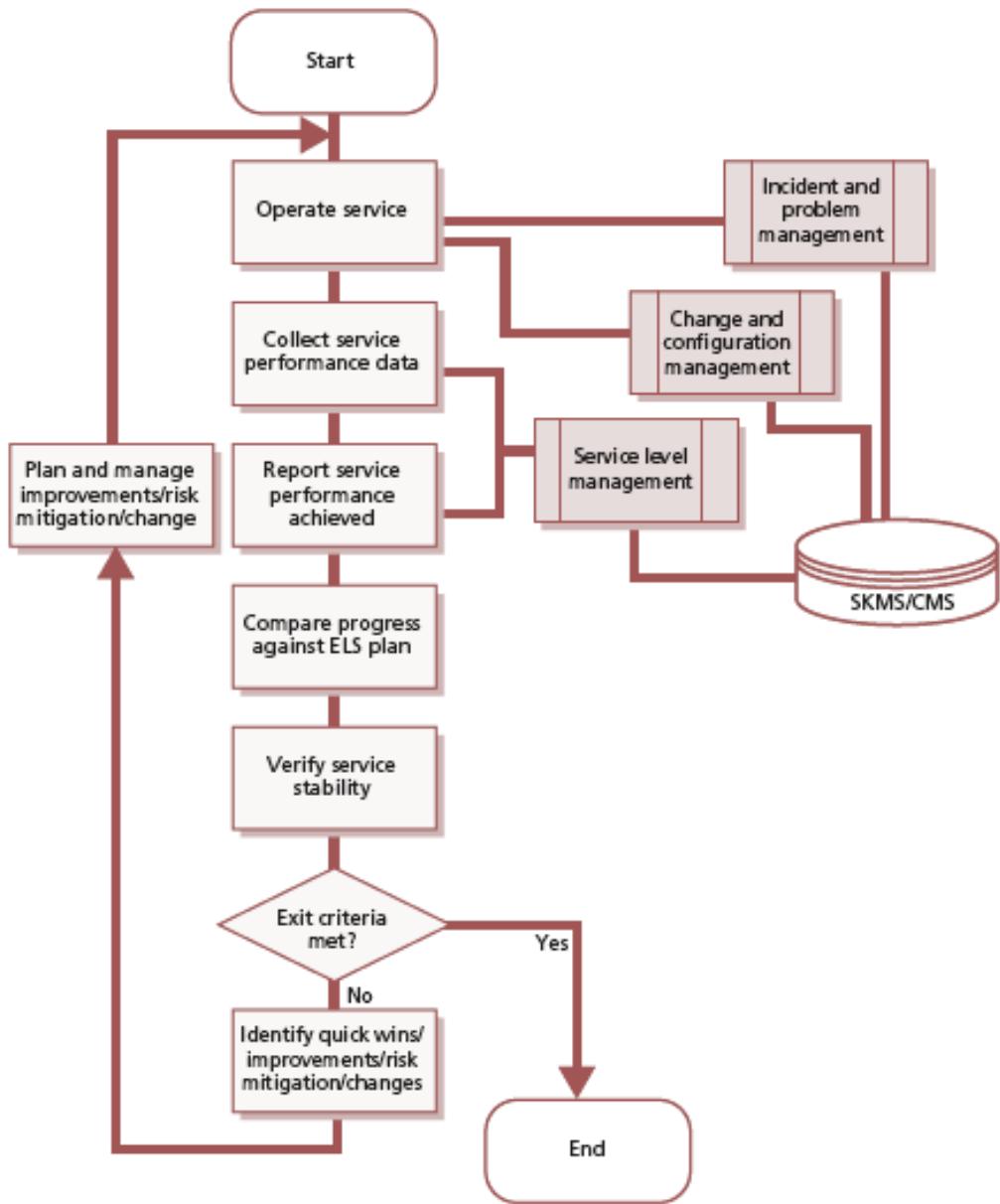


Standard Operational Change



Based on AXELOS ITIL® material. Reproduced under license from AXELOS. (Figure 4.3 – Example of a process flow for a standard deployment request – Figure 4.4 – Example of a process flow for a standard operational request - ITIL® Service Transition 2011 edition)

SD 9 – Early Life Support Activities



Based on AXELOS ITIL® material. Reproduced under license from AXELOS. (Figure 4.26 – Example of early life support activities – *ITIL® Service Transition 2011 edition*)

SD 10 – Example DevOps Tool Chain



DevOps Foundation

Practical Assignments

Assignment One: Overcoming Perceptions (15 minutes)

Assignment Goal

Capture perceptions.

Assignment

1. Provide one word or a short phrase that describes:
 - a. Dev's perceptions of Ops
 - b. Ops' perceptions of Dev
2. List two primary factors that have contributed to the current relationship between Dev and Ops in your organization.
3. What is one change that would enable the relationship between Dev and Ops to improve?

Assignment Two: Enterprise DevOps (15 minutes)

Assignment Goal

Discuss the challenges and possibilities associated with enterprise DevOps.

Assignment

This assignment uses recent Wall Street Journal and Forbes articles that provide opposing perspectives on the challenges and possibilities of enterprise DevOps.

The articles can be found here:

<http://blogs.wsj.com/cio/2014/05/13/devops-is-great-for-startups-but-for-enterprises-it-wont-work-yet/>

<http://blogs.wsj.com/cio/2014/05/22/enterprise-devops-adoption-isnt-mandatory-but-neither-is-survival/>

1. Review the articles
2. Draw your own conclusion
 - a. How long will it be until enterprises are forced to accept that they must accelerate their experiments with DevOps?
 - b. What is needed to make enterprise DevOps a reality?

Assignment Three: Identifying Waste (30 minutes)

Assignment Goal

Identify sources of waste.

Assignment

Lean recognizes seven sources of waste. Provide examples of waste that could be eliminated to improve the flow of work between Dev and Ops.

Lean	DevOps
Defects	
Overproduction	
Inventory	
Over-processing	
Motion	
Transportation	
Waiting	

Assignment Four: Improving Collaboration Skills (15 minutes)

Assignment Goal

Identify characteristics (behaviors and attitudes) of good and poor collaborators.

Assignment

1. List a characteristic of a poor collaborator.
2. List positive behaviors or attitudes that would enable more effective collaboration

Poor Collaborators	Good Collaborators

Assignment Five: Exploiting Constraints (30 minutes)

Assignment Goal

Identify ways to exploit constraints affecting the flow of work between Dev and Ops.

Assignment

1. Specify a constraint affecting your organization, or use one of the common constraints specified in your learner manual.
2. Identify ways to exploit that constraint using existing resources.

Constraint	Quick Improvements to Exploit Constraint

Assignment Six: Integrating ITSM and DevOps (15 minutes)

Assignment Goal

Identify needed ITSM process improvements.

Assignment

1. In addition to those provided in your learner manual, list at least one other ITSM process that integrates closely with DevOps.
2. Describe practices that may need to be refined for this process to integrate more effectively with DevOps?

Assignment Seven: Overcoming Challenges (30 minutes)

Assignment Goal

The goals of this assignment are to identify ways to use OSA principles and concepts to overcome challenges and mitigate risks and to identify critical success factors.

Scenario

Assignment One provided the opportunity to identify challenges and risks that pose a barrier to successful implementation and improvement of OSA processes, models, methods and techniques.

Assignment

This is a whole-class assignment where each member of the class shares ideas and action items he or she is taking away from class.

For each member of the class:

1. Refer to the challenges and risks you identified in Assignment One
2. What, if any, additional challenges or risks have you identified that your organization (or a client's organization) is currently facing or may face
3. Record and share at least one short-term action item you are taking away from class
4. Record and share at least one longer-term action item you are taking away from class
5. What, if any, additional critical success factors have you identified that will help remove barriers to successful implementation and improvement of OSA processes, models, methods and techniques
6. What, if any, conclusions can you reach as a class about overcoming the challenges and mitigating the risks associated with implementing and improving OSA processes

DevOps Foundation

Exam Requirements

DevOps Foundation Certificate



DevOps Foundation is a freestanding certification. The purpose of this course and its associated exam is to impart, test and validate knowledge of DevOps basic vocabulary, principles and practices. The vocabulary terms, principles and practices are documented in the course learner manual. DevOps Foundation is intended to provide individuals an understanding of basic DevOps concepts and how DevOps may be used to improve communication, collaboration and integration between software developers and IT operations professionals.

Eligibility for Examination

To be eligible for the exam leading to certification candidates must fill the following requirements:

- Complete at least 16 contact hours (instruction and labs) as part of a formal, approved training course
- It is recommended that students complete at a minimum 6 hours of personal study by reviewing the vocabulary list and pertinent areas of the course learner manual and by completing the sample exam

Level of Difficulty

The DevOps Foundation certification uses the Bloom Taxonomy of Educational Objectives in the construction of both the content and the examination.

- The DevOps Foundation exam contains Bloom 1 questions that test learners' **knowledge** of DevOps vocabulary terms (see list below) and concepts
- The exam also contains Bloom 2 questions that test learner's **comprehension** of these concepts

Format of the Examination

Candidates must achieve a passing score to gain the DevOps Foundation Certificate.

Exam Type	40 multiple choice questions
Duration	60 minutes for candidates in their respective language 75 minutes if English is not a candidate's native tongue
Prerequisites	DevOps Foundation training
Supervised	Yes
Open Book	No
Passing Score	65%
Delivery	Online

Exam Topic Areas and Question Weighting

The DevOps Foundation exam requires knowledge of the topic areas specified below.

Topic Area	Description	Max Questions
DO FD – 1 Introduction to DevOps	Purpose, objectives and business value of DevOps	4
DO FD – 2 DevOps Principles	Key guiding principles of DevOps	3
DO FD – 3 DevOps and Other Frameworks	The relationship between relevant frameworks and standards and DevOps	6
DO FD – 4 DevOps Culture	Characteristics of a DevOps culture and of culture change	4
DO FD – 5 DevOps Roles and Organizational Considerations	Responsibilities of key roles and considerations relative to organizational structure	2
DO FD – 6 DevOps Practices	Common DevOps practices and related processes	10
DO FD – 7 DevOps and Automation	Common technologies and their role in automation	7
DO FD – 8 Adopting DevOps	Adoption challenges, risks, critical success factors and key performance measures	4

Terminology List

After studying this course, the candidate is expected to understand the meanings of the following terms in the context of DevOps.

- Agile Manifesto
- Agile software development
- CAMS (Culture, automation, measurement, sharing)
- Change failure rate
- Change fatigue
- Change lead time
- Collaboration
- Constraint
- Continuous integration
- Continuous delivery
- Customer focus (outside-in)
- Cycle time
- Daily Scrum meeting (standup)
- Deployment frequency
- DevOps
- Flow
- Impediment
- IT service management
- Kanban
- Kanban board
- Lean production
- Lean six sigma
- Lean IT
- Mean time to detect incidents
- Mean time to recover (MTTR)
- Mean time to restore service (MTRS)
- Model
- Muda
- Organizational culture
- Product backlog
- Scrum
- Scrum team
- Sprint
- The Three Ways
- Theory of Constraints
- Tool chain
- Value stream mapping
- Waste
- Waterfall

DevOps Foundation

Study Aids

DevOps Key Terms: Matching Exercise

In the blank before each term in Column A, write the letter corresponding to the definition from Column B.

Column A	Column B
_____ Agile Manifesto	A. Cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.
_____ IT Infrastructure Library (ITIL)	B. Software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
_____ DevOps	C. Production philosophy that focuses on reducing waste to improve overall customer value.
_____ IT Service Management	D. Formal proclamation of values and principles to guide an iterative and people-centric approach to software development.
_____ Agile Software Development	E. Implementation and management of quality IT services that meet the needs of the business. (ITIL definition)
_____ Scrum	F. Iterative and incremental Agile Framework within which complex products can be developed.
_____ Lean Production	G. Set of best practice publications for IT service management. Published in a series of five core books representing the stages of the IT service lifecycle.

The Three Ways

Specify the correct order of The Three Ways by entering 1-3 in the left column.

Three Ways Order	The Three Ways
	Create a culture that fosters experimentation and learning
	Understand and increase the flow of work
	Create short feedback loops that enable continuous improvement

Agile Manifesto

Using arrows (as illustrated), align the values on the left to the corresponding values on the right.

Value	Over...	Value
Individuals and interactions		Comprehensive documentation
Working software		Following a plan
Customer collaboration		Processes and tools
Responding to change		Contract negotiations

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

DevOps Practices: Matching Exercise

In the blank before each term in Column A, write the letter corresponding to the definition from Column B.

Column A	Column B
_____ Kanban	A. Methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.
_____ Value Stream Mapping	B. Development practice that requires developers to integrate code into a shared repository on a daily basis.
_____ Theory of Constraints	C. Method for managing knowledge work with an emphasis on just-in-time delivery while not overloading the team members.
_____ Continuous Integration	D. Lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer.
_____ Continuous Delivery	E. Set of practices that enable every change that passes automated tests to be automatically deployed to production.
_____ Continuous Deployment	F. Set of management tools that views any manageable system as being limited in achieving more of its goals by a very small number of bottlenecks.

Theory of Constraints

Specify the correct order of the five focusing steps by entering 1-5 in the left column.

Focusing Steps Order	Five Focusing Steps
	Elevate the constraint
	Identify constraint
	Prevent inertia from becoming the constraint
	Exploit the constraint
	Subordinate everything to the constraint

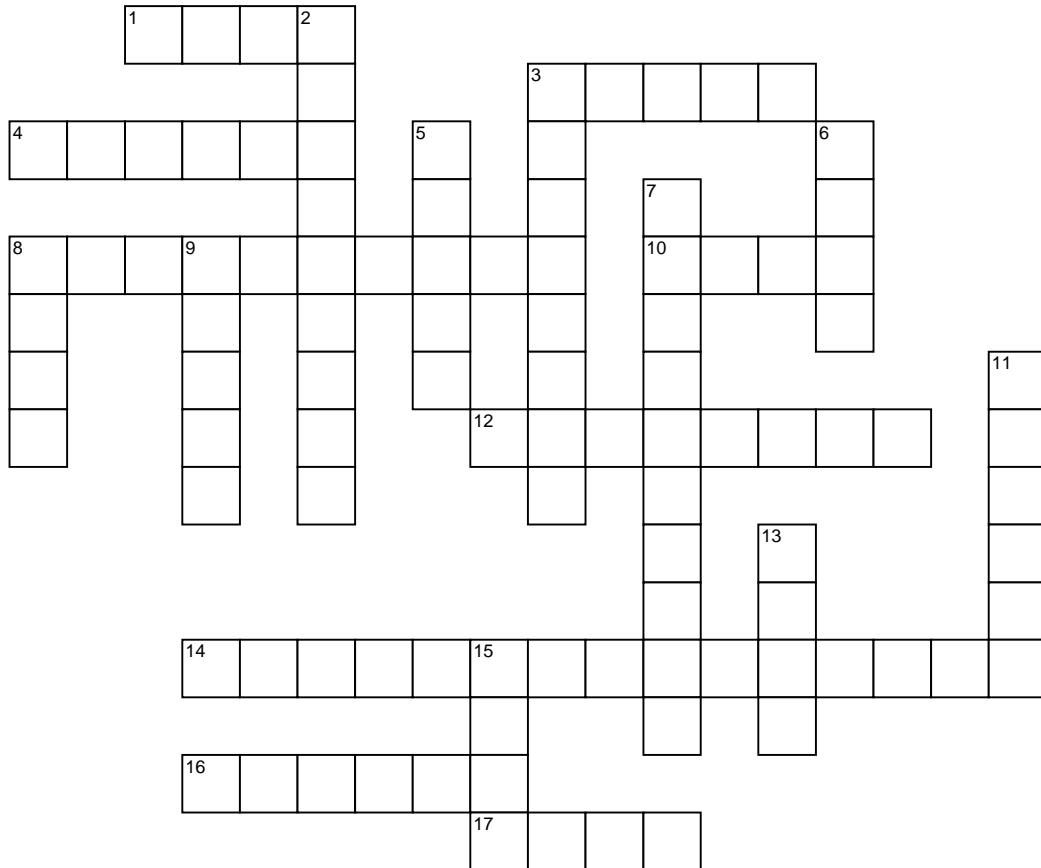
Improvement Kata: Fill in the blanks

Correctly complete each of the following statements.

Improvement Kata
1. Understand the _____
2. Grasp the _____ condition
3. Establish the next _____ condition
4. _____ toward the target condition

DevOps

Vocabulary Terms and Acronyms

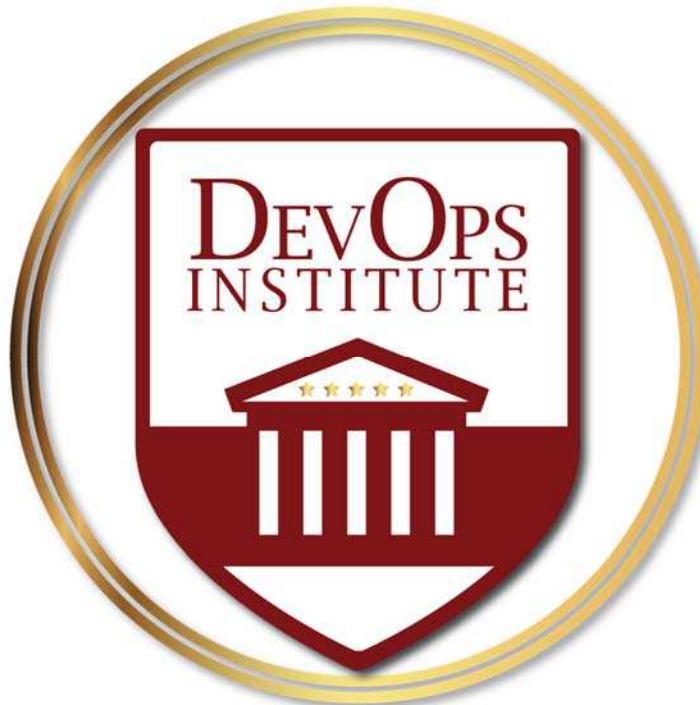


ACROSS

- 1 The First Way
- 3 Time from start of work to ready for delivery
- 4 Time-boxed iteration of work
- 8 Limiting or blocking factor in a value stream
- 10 Waste
- 12 The Second Way
- 14 The Third Way
- 16 Cultural and professional movement
- 17 Time from component failure to repair

DOWN

- 2 Linear, sequential project approach
- 3 Focus of 'outside-in' approach
- 5 Fast moving, flexible
- 6 Focuses on reducing waste and improving flow
- 7 Anything that prevents work (Scrum)
- 8 DevOps values
- 9 Agile framework
- 11 Pull system of work
- 13 Time from request to delivery
- 15 Management of quality IT services



DevOps Foundation Sample Exam

Sample Paper Version 1.0

Answer Key

© DevOps Institute
All Rights Reserved

Question	Answer	Question	Answer
1	B	21	C
2	C	22	B
3	D	23	B
4	D	24	D
5	B	25	C
6	A	26	C
7	B	27	A
8	A	28	B
9	A	29	B
10	A	30	A
11	B	31	D
12	D	32	C
13	A	33	A
14	D	34	B
15	B	35	A
16	C	36	B
17	A	37	A
18	D	38	B
19	A	39	A
20	D	40	C

DevOps

List of Acronyms

Acronym	Definition
BCM	Business Continuity Management
BRM	Business Relationship Management
CAB	Change Advisory Board
CAMS	Culture, Automation, Measurement, Sharing
CALMS	Culture, Automation, Lean, Metrics, Sharing
CI	Configuration Item
CMDB	Configuration Management Database
CMS	Configuration Management System
CSF	Critical Success Factor
CSI	Continual Service Improvement
DIKW	Data-to-Information-to-Knowledge-to-Wisdom
DML	Definitive Media Library
DoD	Definition of Done
ECAB	Emergency Change Advisory Board
ELS	Early Life Support
ISM	Information Security Management
ISMS	Information Security Management System
ISP	Information Security Policy
ITIL	IT Infrastructure Library
ITSCM	IT Service Continuity Management
ITSM	IT Service Management
KEDB	Known Error Database
KPI	Key Performance Indicator
MTBF	Mean Time Between Failures
MTBSI	Mean Time Between Service Incidents
MTTD	Mean Time to Detect
MTTR	Mean Time to Repair
MTRS	Mean Time to Restore Service
OLA	Operational Level Agreement
PBAs	Patterns of Business Activity
PDCA	Plan-Do-Check-Act
PIR	Post-implementation Review
RACI	Responsible-Accountable-Consulted-Informed
RCA	Root Cause Analysis
RFC	Request for Change
ROI	Return on Investment
SAC	Service Acceptance Criteria
SACM	Service Asset and Configuration Management
SDP	Service Design Package
SIP	Service Improvement Plan
SKMS	Service Knowledge Management System
SLA	Service Level Agreement
SLM	Service Level Management
SLR	Service Level Requirement
SPM	Service Portfolio Management
UC	Underpinning Contract
VBF	Vital Business Function
VOC	Voice of the Customer
WIP	Work in Progress