

SAFe® DevOps

Improving time-to-market with the Scaled Agile Framework®

SAFe® Authorized Course Attending this course gives students access to the SAFe® DevOps Practitioner exam and related preparation materials.



V4.5.1.1

SCALED AGILE®

© Scaled Agile, Inc.

Welcome to the course!

Make the Most of Your Learning



Access the SAFe Community Platform

Manage your member profile, join communities of practice, and access the member directory



Prepare Yourself

Prepare for certification with your learning plan: access your course workbook, study materials, and practice test prior to taking your certification exam



Become a SAFe Certified Professional

Get certified to achieve recognition of your skills and open the door to new career opportunities



Showcase SAFe Credentials

Use your digital badge to view global job insights, track market labor data, and see where your skills are in demand

community.scaledagile.com

SAFe® PROVIDED BY **SCALED AGILE®**

Logistics

- ▶ Class times
- ▶ Breaks
- ▶ Lunch
- ▶ Restrooms
- ▶ Working agreements



SCALED AGILE® © Scaled Agile, Inc.

1-3

Course goals

At the end of this course, you should be able to:

- ▶ Describe what DevOps is and why it's important
- ▶ Describe the importance of continuous testing
- ▶ Describe the value of continuous security
- ▶ Map your delivery pipeline
- ▶ Measure flow through the delivery pipeline
- ▶ Identify gaps and delays in flow
- ▶ Improve the process of exploring customer needs
- ▶ Improve the process of developing, building, and integrating continuously
- ▶ Improve the process of deployment to staging and production environments continuously
- ▶ Improve the release process
- ▶ Build an action plan for a SAFe DevOps transformation



SCALED AGILE® © Scaled Agile, Inc.

1-4

Course map

- ▶ Lesson 1: Introducing DevOps
- ▶ Lesson 2: Mapping Your Pipeline
- ▶ Lesson 3: Gaining Alignment with Continuous Exploration
- ▶ Lesson 4: Building Quality with Continuous Integration
- ▶ Lesson 5: Reducing Time-to-Market with Continuous Deployment
- ▶ Lesson 6: Delivering Business Value with Release on Demand
- ▶ Lesson 7: Taking Action
- ▶ Lesson 8: About the Exam



SCALED AGILE® © Scaled Agile, Inc.

1-5

Course interactivity

During this course, you will be engaged with:



Discussions



Practice Activities



Scenario-Based Learning



Problem-Solving Activities



Video Resources

SCALED AGILE® © Scaled Agile, Inc.

1-6

SAFe® DevOps

Lesson 1: Introducing DevOps



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

SCALED AGILE® © Scaled Agile, Inc.

1-7

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Recognize the problem to be solved
- ▶ Explain DevOps and its benefits
- ▶ Describe continuous security and testing
- ▶ Describe core DevOps principles



SCALED AGILE® © Scaled Agile, Inc.

1-8

1.1 Recognize the problem to be solved

SCALED AGILE® © Scaled Agile, Inc.

1-9



Digital disruption
is affecting every
industry across
the globe.

SCALED AGILE® © Scaled Agile, Inc.



“Every business is a software business now. Agility isn't an option, or a thing just for teams, it is a business imperative. But we struggle building big systems ... ”

—Dean Leffingwell
Creator of SAFe

SCALED AGILE® © Scaled Agile, Inc.



Chapter 1: The Beginning

- ▶ Step 1: Read ‘Chapter 1: The Beginning.’
- ▶ Step 2: Pair with someone and introduce yourself.
- ▶ Step 3: Summarize the challenges of the current situation at DynotransX. What are some of the challenges you face at your company?
- ▶ Step 4: Do you agree with any of the solutions offered by Travis and Tracy?



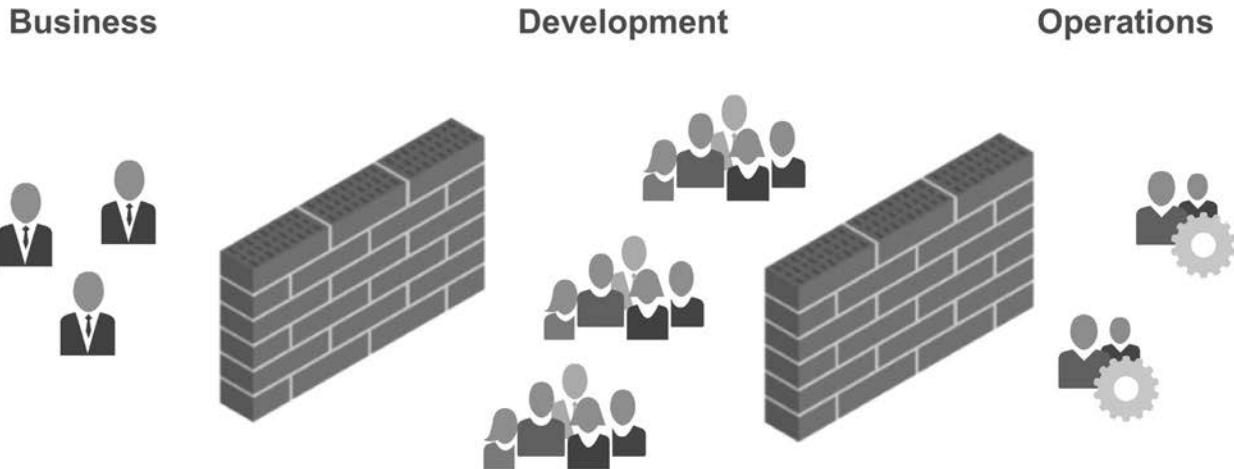
PREPARE SHARE



1-12

SCALED AGILE® © Scaled Agile, Inc.

Traditionally there are walls of confusion

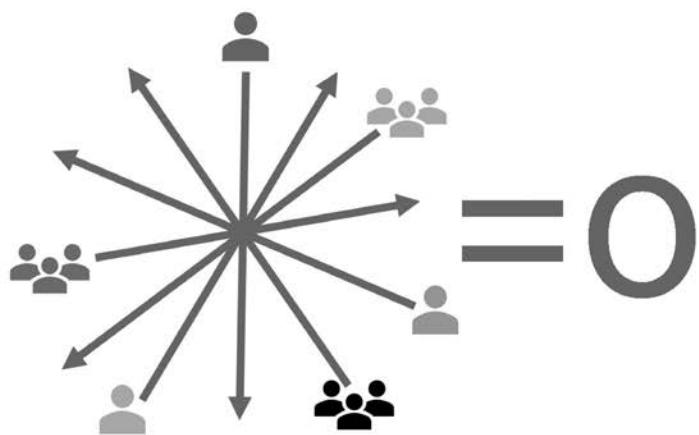


SCALED AGILE® © Scaled Agile, Inc.

1-13

Lack of alignment impedes progress

- ▶ Different groups in the organization have different goals and direction
- ▶ The lack of alignment means their different efforts cancel each other out
- ▶ This creates a feeling of constant work with little or no progress



Building alignment is a critical goal of DevOps

SCALED AGILE® © Scaled Agile, Inc.

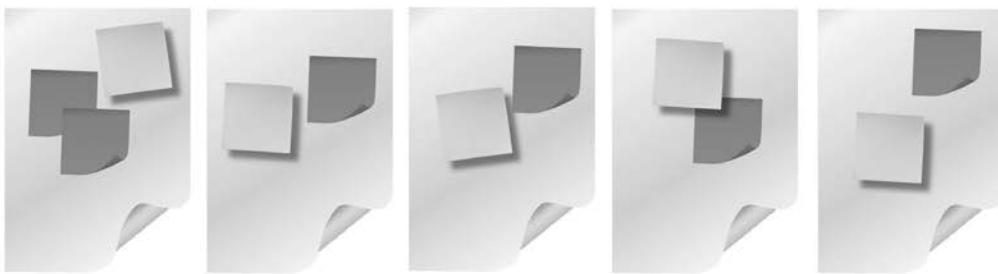
1-14

1.2 Explain DevOps and its benefits



Activity: What does DevOps mean to you?

- ▶ Step 1: Think about single words or short phrases that define what DevOps means to you.
- ▶ Step 2: Write each word or phrase (individually or as a team) on a separate sticky note. Hand them to your trainer when you're done.



PREPARE



1-16

DevOps defined (-ish)

"DevOps is a software engineering culture and *practice* that aims at unifying software development (Dev) and software operation (Ops)."

The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction from integration, testing, releasing to deployment and infrastructure management.

DevOps aims at shorter development cycles, increased deployment frequency, more dependable releases, *in close alignment with business objectives.*"

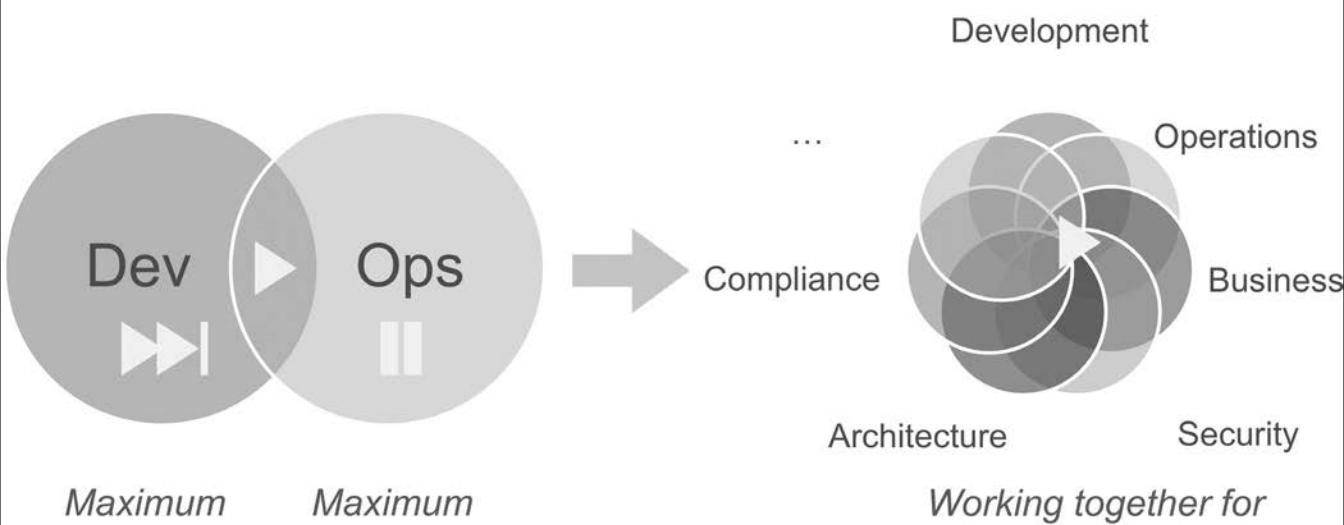


— Wikipedia

SCALED AGILE® © Scaled Agile, Inc.

1-17

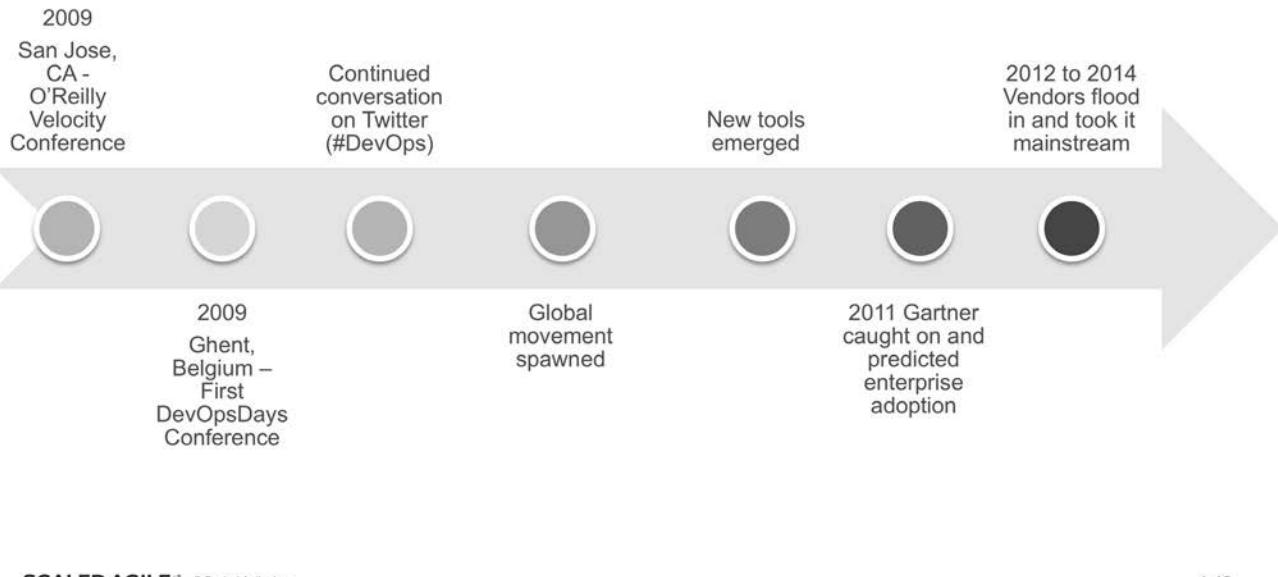
Who is DevOps?



SCALED AGILE® © Scaled Agile, Inc.

1-18

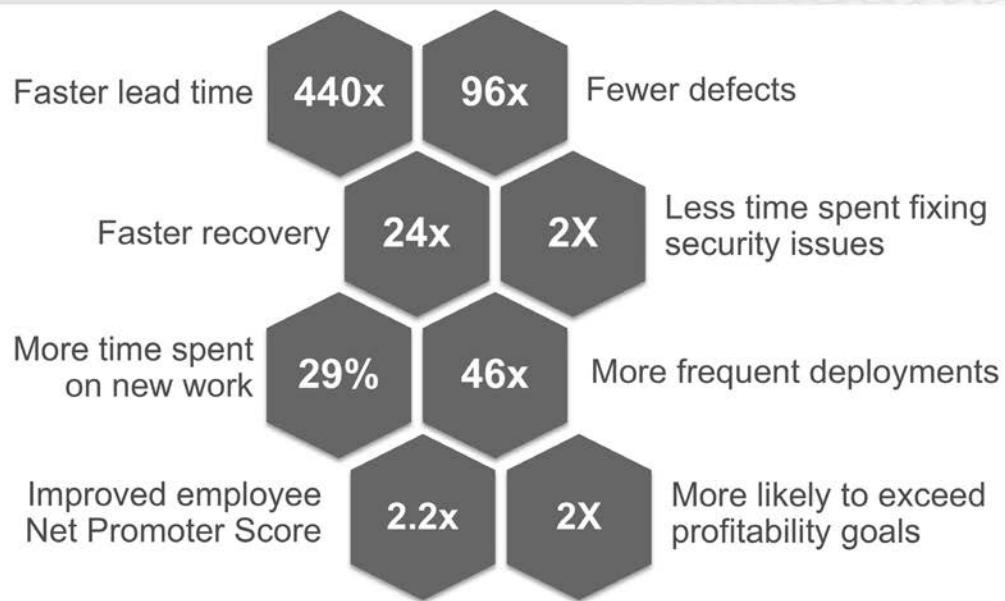
The history of DevOps



SCALED AGILE® © Scaled Agile, Inc.

1-19

DevOps benefits

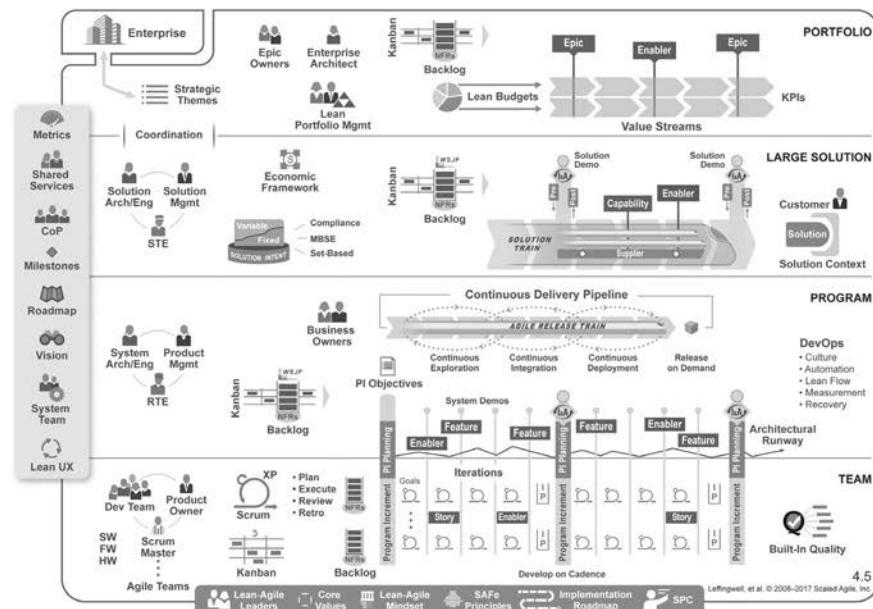


SCALED AGILE® © Scaled Agile, Inc.

Source: <https://puppet.com/resources/whitepaper/state-of-devops-report>

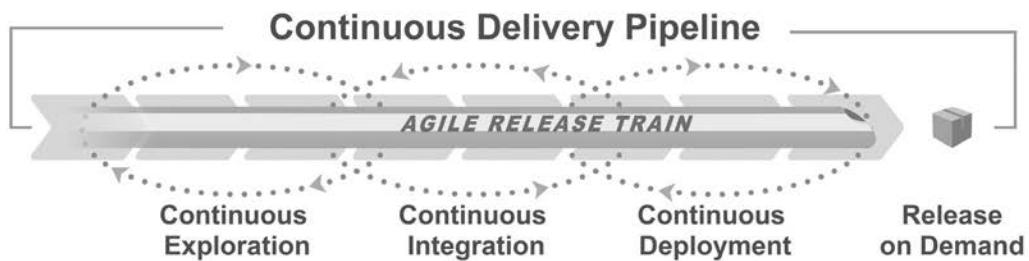
1-20

SAFe helps bridge the gap between Agile and DevOps



1-21

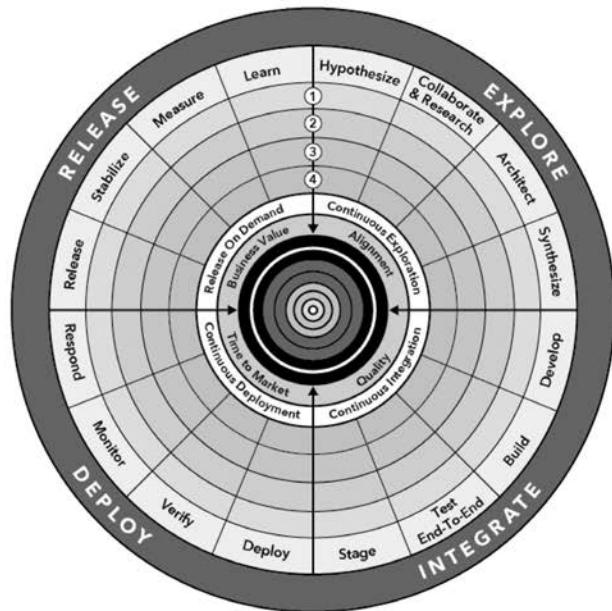
The Continuous Delivery Pipeline



The DevOps health radar

“Every team can complete a full revolution in less than 24 hours.”

– Sr. Engineering Lead, Amazon



SCALED AGILE® © Scaled Agile, Inc.

1-23

But wait, ‘Our problems are different’

We aren't building a website.

We aren't hosted in the cloud.

Our customers don't want continuous delivery.

Our technology isn't based on micro services.

We don't have just the one platform.

SCALED AGILE® © Scaled Agile, Inc.

1-24

Systems thinking

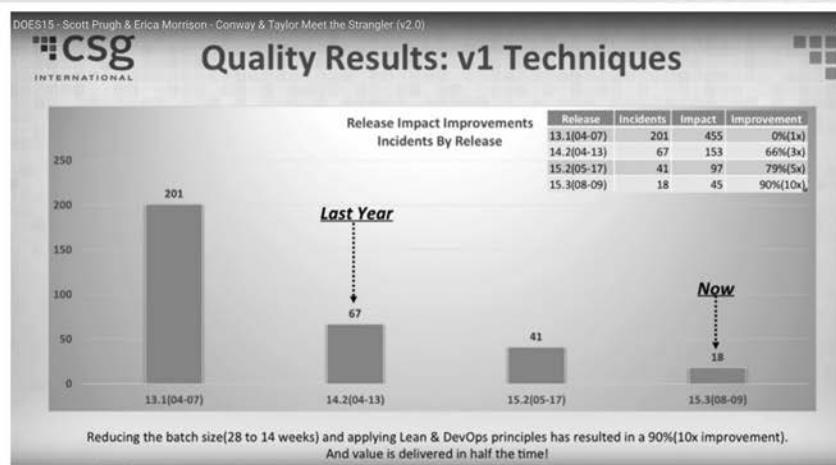


A common disease that afflicts management the world over is the impression that, “Our problems are different.”

They are different to be sure, but the principles that will help to improve quality of product and service are universal in nature.

—W. Edwards Deming

DevOps in a legacy environment – 10x improvement

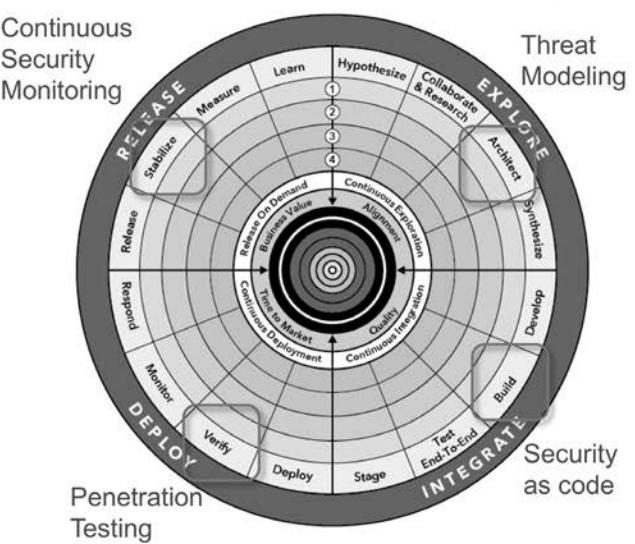


Scott Prugh: DevOps in Legacy Environments
<https://youtu.be/f4et0EGvKXA>
25:38

1.3 Describe continuous security and testing

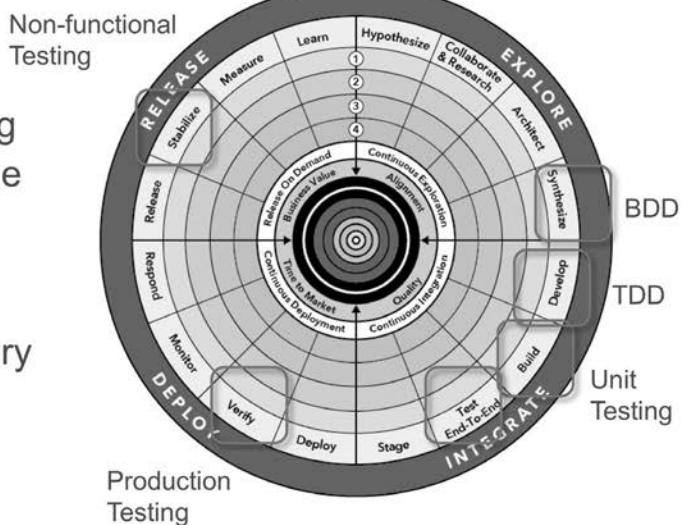
The role of continuous security

- ▶ Information security is a key concept in SAFe DevOps
- ▶ DevSecOps concepts present key principles for value delivery
- ▶ Security affects every dimension of the continuous delivery cycle
- ▶ Information security should be part of every DevOps transformation



The role of continuous testing

- ▶ Testing is an ongoing activity
- ▶ We build quality in by addressing testing and quality throughout the continuous delivery cycle
- ▶ Automated testing and quality assurance should be part of every DevOps transformation



SCALED AGILE® © Scaled Agile, Inc.

1-29

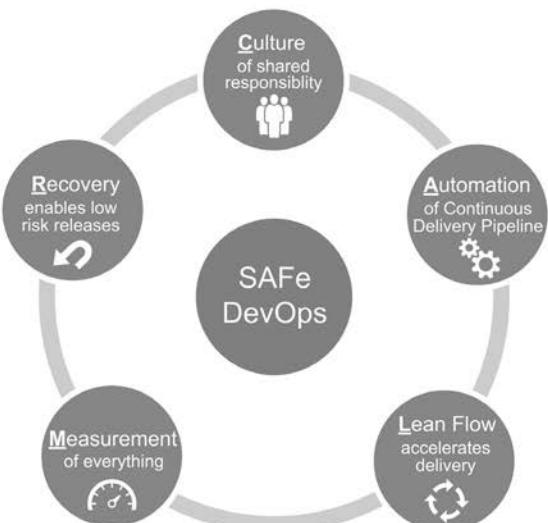
1.4 Describe core DevOps principles

SCALED AGILE® © Scaled Agile, Inc.

1-30

A CALMR approach to DevOps

- ▶ **C**ulture Establish a culture of shared responsibility for development, deployment, and operations.
- ▶ **A**utomation Automate the Continuous Delivery Pipeline.
- ▶ **L**ean flow Keep batch sizes small, limit WIP, and provide extreme visibility.
- ▶ **M**easurement Measure the flow through the pipeline. Implement full-stack telemetry.
- ▶ **R**ecover Architect and enable low-risk releases. Establish fast recovery, fast reversion, and fast fix-forward.



SCALED AGILE® © Scaled Agile, Inc.

1-31

DevOps is a Cultural shift

- ▶ Adopt a culture of shared responsibility for development and deployment
- ▶ DevOps requires a tolerance for failure and rapid recovery, and rewards risk taking
- ▶ Sharing discoveries, practices, tools, and learning across silos is encouraged

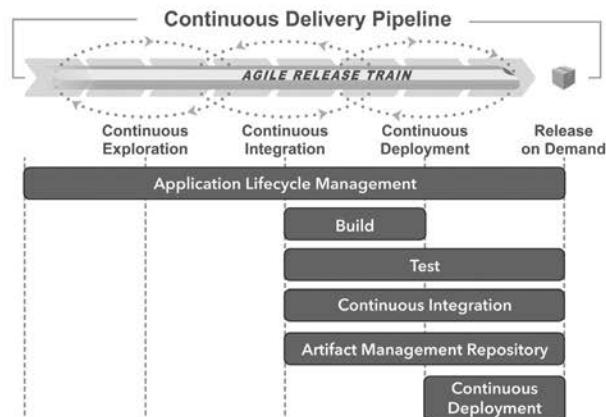


SCALED AGILE® © Scaled Agile, Inc.

1-32

Automate the deployment process

- ▶ Automation is a key concept in DevOps
- ▶ Automate as much as you can from the Continuous Delivery Pipeline – fewer manual steps improves the flow of value
- ▶ Build a comprehensive toolchain to help teams release value more frequently
- ▶ Focus on automating healthy processes. If the underlying process is broken, fix it before automating it



SCALED AGILE® © Scaled Agile, Inc.

1-33

Focus on continuous Lean flow of value

- ▶ Identify bottlenecks to the flow of value
- ▶ Decrease the batch sizes of the work
- ▶ Manage and reduce queue lengths



SCALED AGILE® © Scaled Agile, Inc.

1-34

Measure everything

- ▶ Collect data on business, application, infrastructure, and client layers
- ▶ Collect data about the deployment pipeline itself
- ▶ Maintain different telemetry for different stakeholders
- ▶ Broadcast measurements
- ▶ Continuously improve telemetry during and after problem solving



SCALED AGILE® © Scaled Agile, Inc.

1-35

Architect for release-ability and Recovery

- ▶ Adopt a stop-the-line mentality
- ▶ Plan for and rehearse failures
- ▶ Build the environment for both roll-back and fix-forward



Source: *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* Kim, Gene; Humble, Jez; Debois, Patrick; Willis, John

SCALED AGILE® © Scaled Agile, Inc.

1-36



Activity: CALMR

- ▶ Step 1: Revisit the CALMR board. Which CALMR concepts are strongly represented? Which ones are under represented? Why?



SUMMARIZE



1-37

SCALED AGILE® © Scaled Agile, Inc.

Lesson review

In this lesson you learned to:

- ▶ Recognize the problem to be solved
- ▶ Explain DevOps and its benefits
- ▶ Describe continuous security and testing
- ▶ Describe core DevOps principles



SCALED AGILE® © Scaled Agile, Inc.

1-38

SAFe® DevOps

Lesson 2: Mapping Your Pipeline

SCALED AGILE® © Scaled Agile, Inc.



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

2-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Explain the purpose of mapping the Value Stream
- ▶ Visualize the current state of the delivery pipeline using key metrics



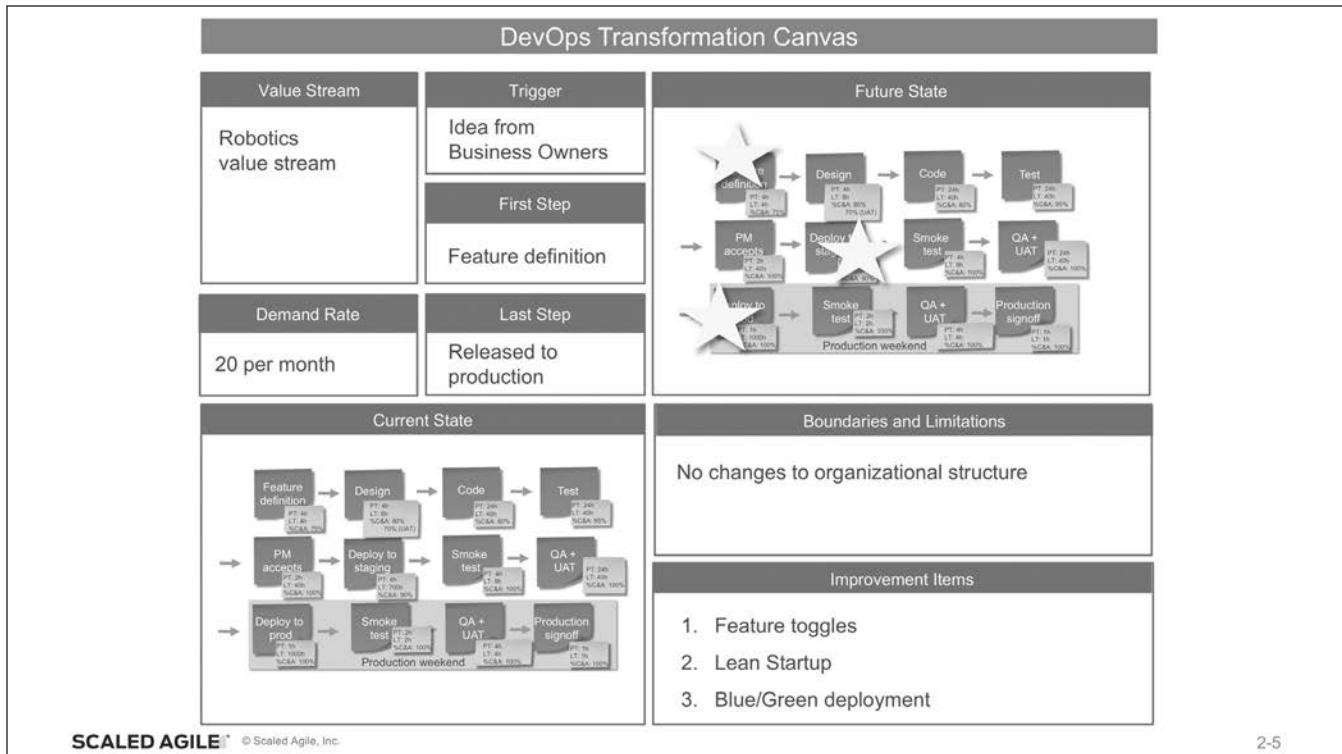
SCALED AGILE® © Scaled Agile, Inc.

2-2

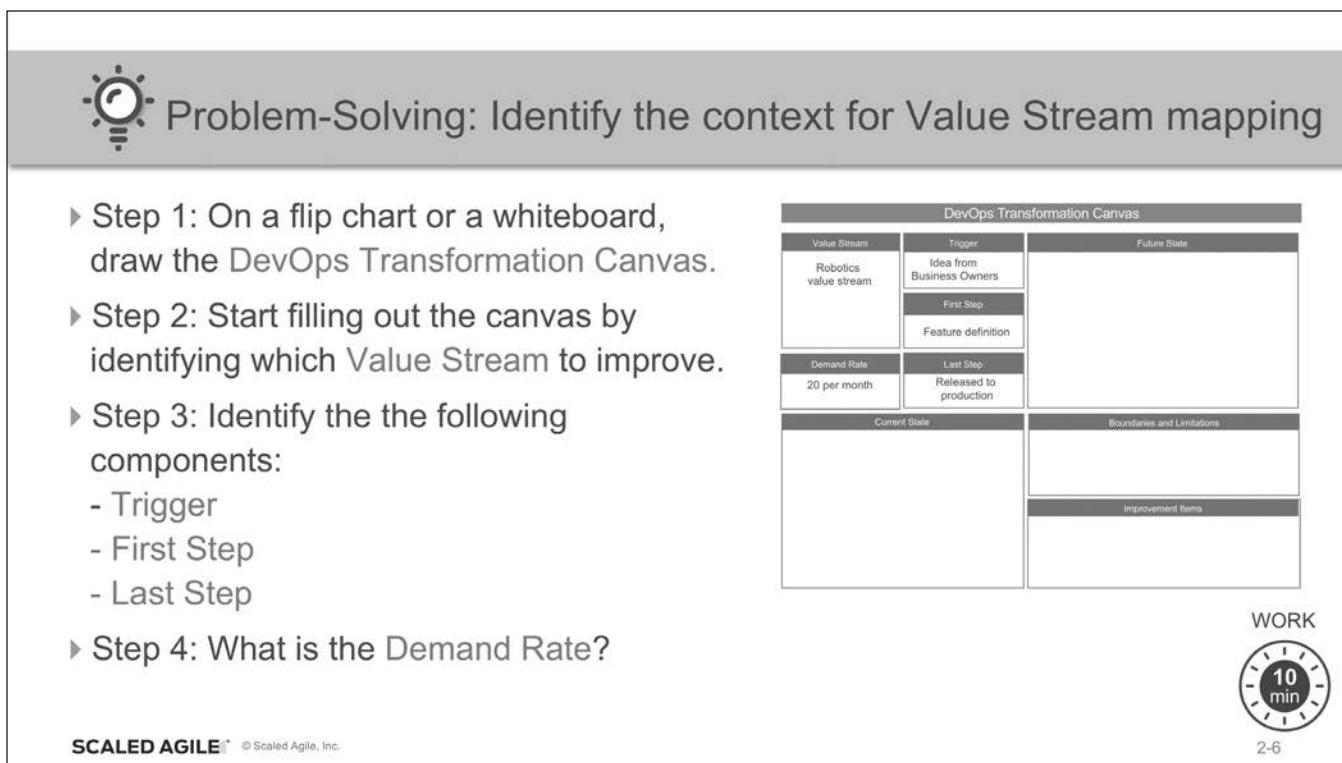
2.1 Explain the purpose of mapping the Value Stream

Why map the Value Stream?

- ▶ Understand how value flows through the enterprise and the various functions of the business
- ▶ Gain insight into organizational efficiency
- ▶ Identify bottlenecks to the flow of value
- ▶ Understand how we can improve the flow of value



2-5

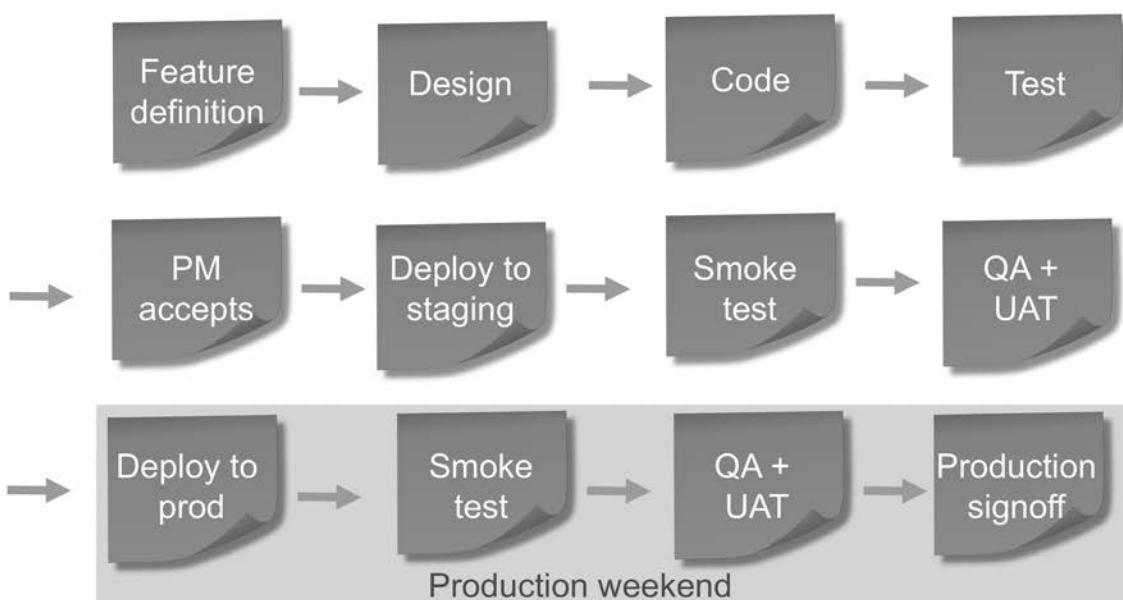


2.2 Visualize the current state of the delivery pipeline using key metrics

SCALED AGILE® © Scaled Agile, Inc.

2-7

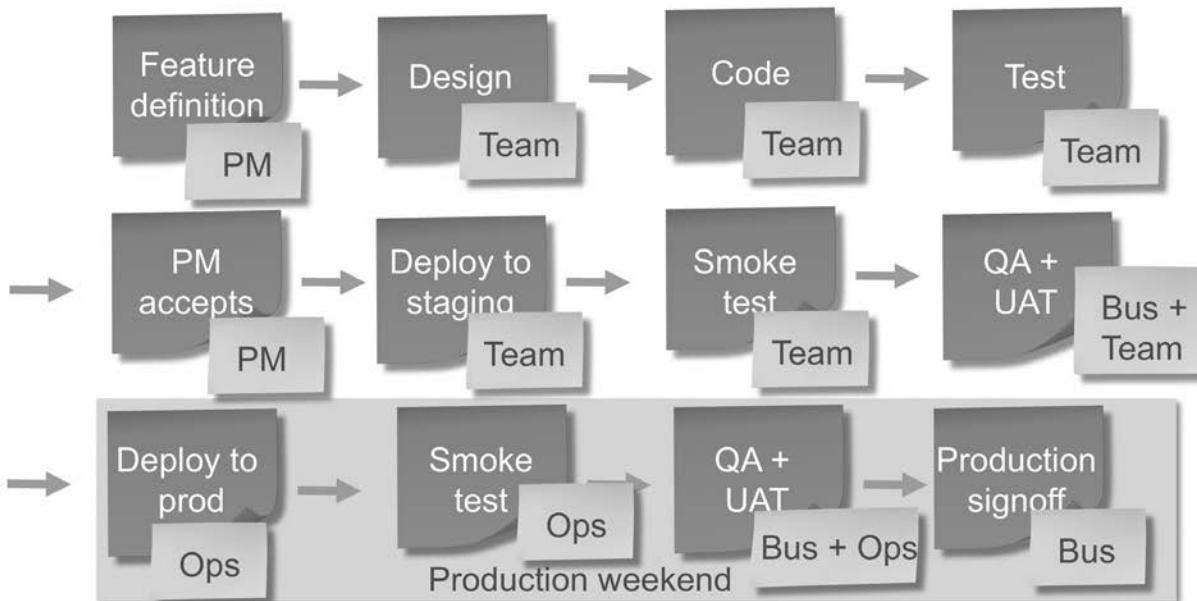
Value Stream mapping – identify the steps – customer example



SCALED AGILE® © Scaled Agile, Inc.

2-8

Value Stream mapping – identify the people



SCALED AGILE® © Scaled Agile, Inc.

2-9



Problem-Solving: Visualize the delivery pipeline

- ▶ Step 1: On the current state section of the canvas, visualize your current delivery pipeline using sticky notes. Start at the first step and finish at the last step identified in the previous exercise.
- ▶ Step 2: Identify the groups participating in every step.



SCALED AGILE® © Scaled Agile, Inc.

2-10



Problem-Solving: Gallery walk and adjust

- ▶ Step 1: Walk the walls and review other peoples' maps. Each team should have one person at their map to answer questions. (Also, rotate people so that everyone gets to walk the gallery).
- ▶ Step 2: Identify things from other maps that you might have missed at your map.
- ▶ Step 3: As a group, adjust your current state map.

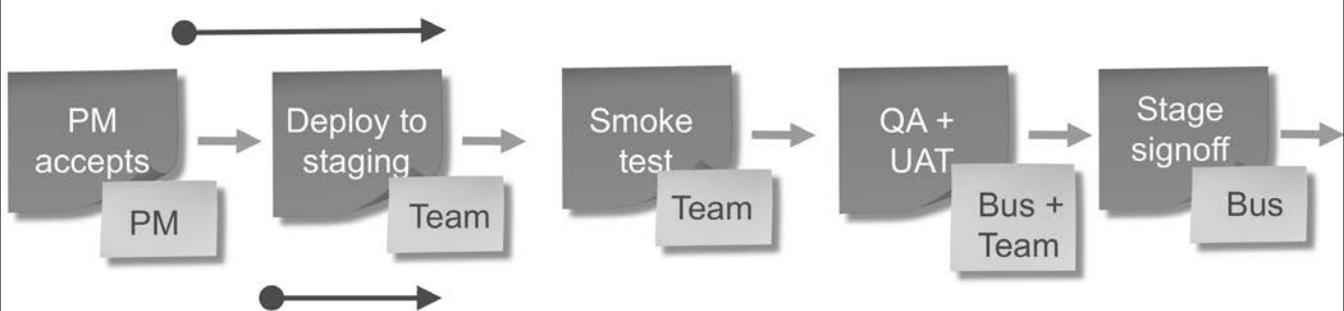


SCALED AGILE® © Scaled Agile, Inc.

2-11

Value Stream measurements

Lead time (LT) – Time from when work was ready after the previous station to completion

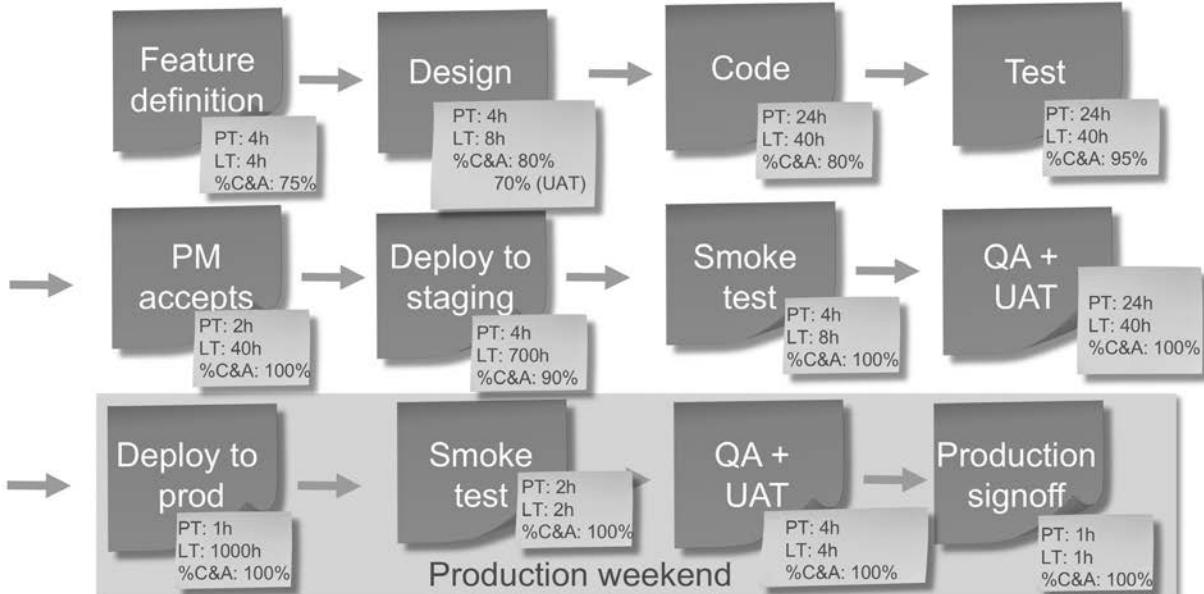


Percent complete and accurate (%C&A) – Percent of work that the next station could process as-is

SCALED AGILE® © Scaled Agile, Inc.

2-12

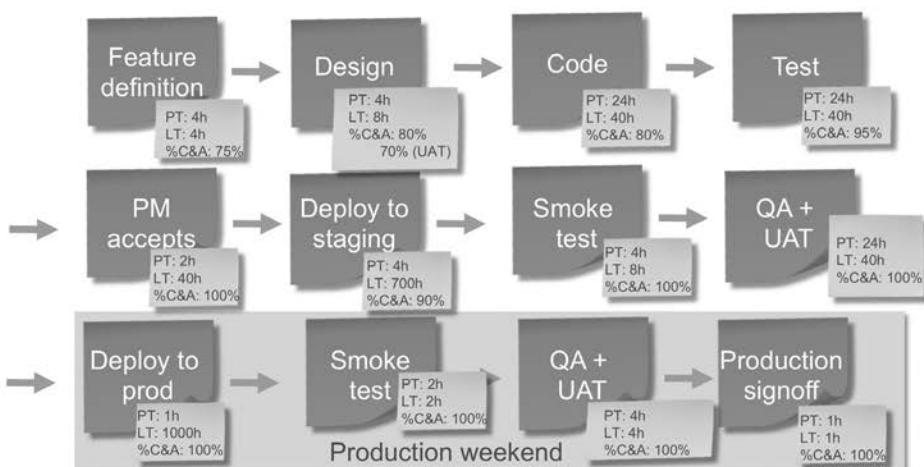
Value Stream mapping – measure the steps



SCALED AGILE® © Scaled Agile, Inc.

2-13

Value Stream map – calculate the total metrics



Total PT = 98 hours
Total LT = 1887 hours
Activity ratio = 5%
Rolled %C&A = 36%

Activity ratio = PT/LT

Rolled %C&A = %C&A₁ * %C&A₂ * %C&A_n * 100

SCALED AGILE® © Scaled Agile, Inc.

2-14



Problem-Solving: Measure the delivery pipeline

- ▶ Step 1: Measure the lead time, process time, and % complete and accurate for each step.
(HINT: Ask people responsible for the next step about the % complete and accurate of the current step.)
- ▶ Step 2: Calculate the total lead time, total process time, activity ratio, and rolled % complete and accurate.



SCALED AGILE® © Scaled Agile, Inc.

2-15



Readout: Define and measure the delivery pipeline

- ▶ Step 1: As a team, present your board and address the following questions:
 - When does the process start (trigger and first step) and end (last step)?
 - What ‘things’ (units) flow through the system?
 - What activities make up the workflow?
 - What is the total lead time, total process time?
 - What is the activity ratio and the rolled % complete and accurate?
 - Where are the biggest bottlenecks?



SCALED AGILE® © Scaled Agile, Inc.

2-16

Lesson review

In this lesson you learned to:

- ▶ Explain the purpose of mapping the Value Stream
- ▶ Visualize the current state of the delivery pipeline using key metrics



SAFe® DevOps

Lesson 3: Gaining Alignment with Continuous Exploration



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

SCALED AGILE® © Scaled Agile, Inc.

3-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Create solution hypotheses
- ▶ Collaborate and research customer needs
- ▶ Architect the solution for continuous delivery
- ▶ Synthesize the Vision, the Roadmap, and the Program Backlog



SCALED AGILE® © Scaled Agile, Inc.

3-2



Chapter 2: A brilliant idea

- ▶ Step 1: Read 'Chapter 2: A brilliant idea.'
- ▶ Step 2: Form pairs and assume the roles of Ben (The Business Owner) and Travis (The Lead Developer).
- ▶ Step 3: From the perspective of the role you play, defend or challenge the outlined assumptions.



PREPARE SHARE



3-3

SCALED AGILE® © Scaled Agile, Inc.

Explore dimension

Business objective (The 'Why')

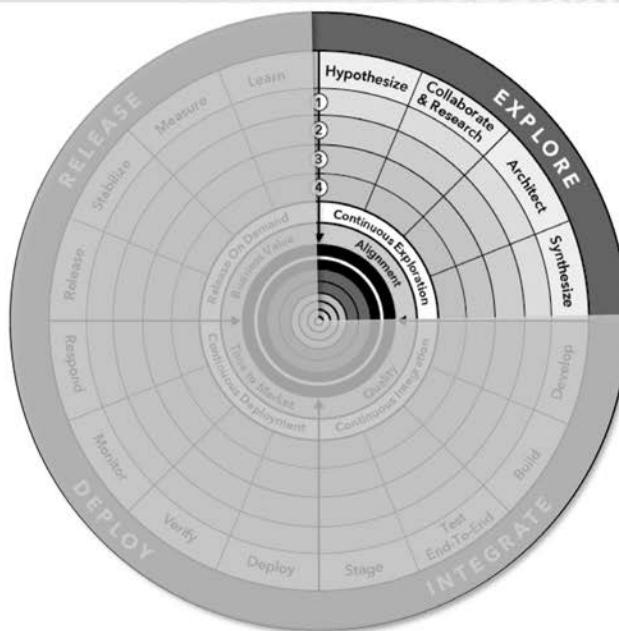
- ▶ Alignment

IT objective (The 'How')

- ▶ Continuous Exploration (CE)

Sub-dimensions (The 'What')

- ▶ Hypothesize
- ▶ Collaborate & Research
- ▶ Architect
- ▶ Synthesize



SCALED AGILE® © Scaled Agile, Inc.

3-4

3.1 Create solution hypotheses

Hypothesize



Hypothesize

Collaborate & Research

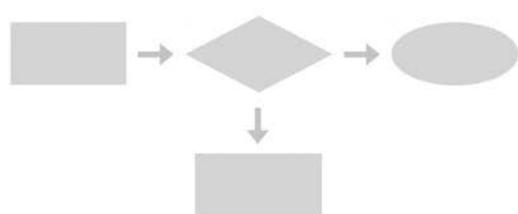
Architect

Synthesize

Purpose: Define the hypothesis to be validated through the Continuous Delivery Pipeline

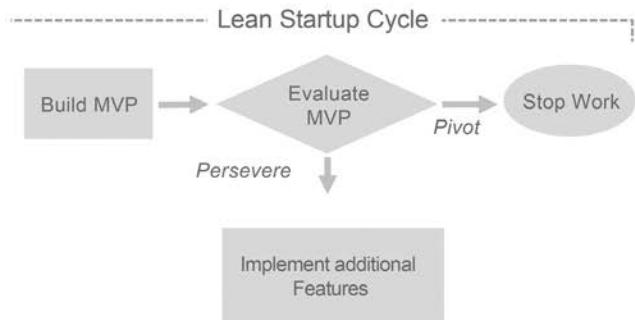
Competencies:

- Lean Startup
- Innovation accounting



Lean Startup

- ▶ The Lean Startup cycle focuses on identifying the viability of ideas
- ▶ It follows the Plan-Do-Check-Adjust cycle (PDCA)
- ▶ Focus on the Minimal Viable Product (MVP) that can prove or disprove the hypothesis
- ▶ Measure, and then persevere or pivot without guilt or mercy
- ▶ Most err in persevering too long



SCALED AGILE® © Scaled Agile, Inc.

3-7

Innovation accounting

- ▶ New products and new features are hard to measure by traditional accounting standards
- ▶ When defining an MVP it is important to use metrics that will validate its success or failure
- ▶ It is important to focus on metrics that demonstrate real customer engagement and not on vanity metrics



SCALED AGILE® © Scaled Agile, Inc.

3-8



Self assess: Hypothesize sub-dimension

► Individually assess where you are:

1. **CRAWL** — Requirements are not written as hypotheses.
2. **WALK** — Some requirements are written as hypotheses.
3. **RUN** — All requirements are written as hypotheses, and some have concrete measures and/or a Minimal Viable Product.
4. **FLY** — All requirements are written as hypotheses with concrete measures and a Minimal Viable Product.

► Place a dot on the DevOps Health Radar.



SCALED AGILE® © Scaled Agile, Inc.

3-9

3.2 Collaborate and research customer needs

SCALED AGILE® © Scaled Agile, Inc.

3-10

Collaborate & Research



Hypothesize

Collaborate & Research

Architect

Synthesize

Purpose: Work with multiple stakeholders to understand the needs

Competencies:

- Lean UX
- Research

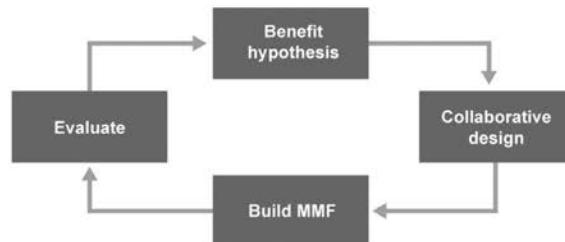


SCALED AGILE® © Scaled Agile, Inc.

3-11

Lean UX

- ▶ Lean User Experience (Lean UX) is a mindset, culture, and a process that implements functionality in minimum viable increments and determines success by measuring results against a benefit hypothesis
- ▶ Features must be broken into Minimal Marketable Features – the minimum functionality that the teams can build to learn whether the benefit hypothesis is valid or not



SCALED AGILE® © Scaled Agile, Inc.

3-12

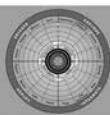
Research

- ▶ **Customer visits** – There is no substitute for observing users in their specific Solution Context.
- ▶ **Gemba walks** – A Gemba walk ('Gemba' is the place where the work is performed) can be used by developers to observe how internal stakeholders execute the steps and specific activities in their operational value streams.
- ▶ **Elicitation** – There are a variety of structured elicitation techniques that can be used, such as interviews, surveys, competitive analysis, requirements workshops, and use-case modeling.
- ▶ **Trade studies** – Teams engage in trade studies to determine the most practical characteristics of a solution.



SCALED AGILE® © Scaled Agile, Inc.

3-13



Self assess: Collaborate and research sub-dimension

- ▶ Individually assess where you are:
 1. **CRAWL** — Product Management creates requirements as a big batch by themselves.
 2. **WALK** — Product Management uses multiple research approaches and collaborates with teams and business stakeholders to create requirements.
 3. **RUN** — Product Management uses multiple research approaches and collaborates with teams and business stakeholders to create requirements with benefit hypotheses.
 4. **FLY** — Product Management uses multiple research approaches and collaborates with teams and business stakeholders to create requirements and identify Minimal Marketable Features with benefit hypotheses.
- ▶ Place a dot on the DevOps Health Radar.



SCALED AGILE® © Scaled Agile, Inc.

3-14

3.3 Architect the Solution for continuous delivery

SCALED AGILE® © Scaled Agile, Inc.

3-15

Architect



Hypothesize Collaborate & Research Architect Synthesize

Purpose: Architect for continuous delivery and DevOps

Competencies:

- Architect for testability
- Separate deploy and release
- Decouple release elements
- Architect for operations
- Threat modeling

- #1 - Balance emergence and intentionality
- #2 - Design for congruence; communicate for clarity
- #3 - Empower local ownership
- #4 - Architect for flow
- #5 - Pursue minimally sufficient

SCALED AGILE® © Scaled Agile, Inc.

3-16

Architect for testability

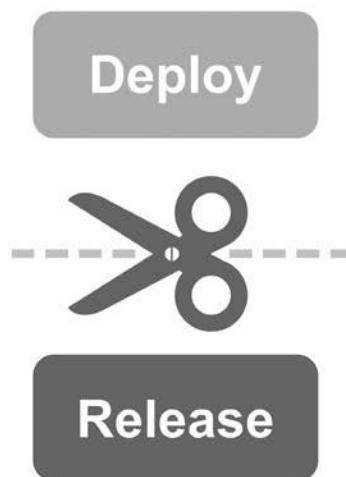
- ▶ Systems that can't readily be tested can't readily be changed
- ▶ In a system that is designed for testability, all jobs require less time
- ▶ Common DevOps patterns:
 - Loose coupling
 - APIs
 - Microservices
 - Containerization
 - Stranglers



Test

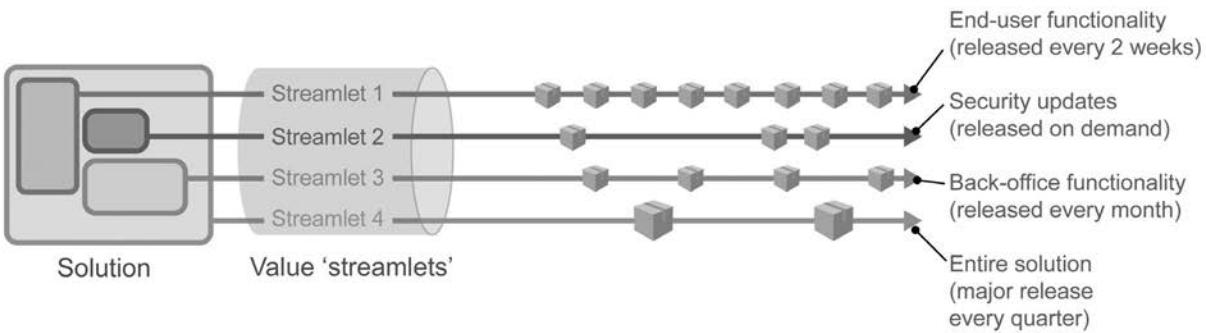
Separate deploy from release

- ▶ Separate deploy to production from release
- ▶ Hide all new functionality under feature toggles
- ▶ Enable the ability to deploy and verify in production and release on demand



Decouple release elements

- ▶ Different parts of the solution require different release strategies
- ▶ Architect the solution to enable the various strategies and to shift them over time based on business demand



SCALED AGILE® © Scaled Agile, Inc.

3-19

Architect for operations

- ▶ Take the operational needs into account
- ▶ Build telemetry and logging capabilities into every application and into the solution as a whole
- ▶ Allow services to be downgraded or even removed in times of high loads or in response to incidents
- ▶ Build capabilities for fast recovery and for fix-forward



SCALED AGILE® © Scaled Agile, Inc.

3-20

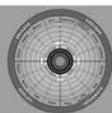
Threat modeling

- ▶ InfoSec consideration should start early
- ▶ Identify potential security threats
- ▶ Build security considerations into architecture
- ▶ Identify threats to proposed architecture, infrastructure, and applications



SCALED AGILE® © Scaled Agile, Inc.

3-21



Self assess: Architect sub-dimension

- ▶ Individually assess where you are:
 - 1. CRAWL** — Architecture doesn't account for continuous delivery and DevOps.
 - 2. WALK** — Architecture is built for testability.
 - 3. RUN** — Architecture is built for testability, separation of deployment and release, and telemetry.
 - 4. FLY** — Architecture is built for testability, separation of deployment and release, telemetry, and operations.
- ▶ Place a dot on the DevOps Health Radar



SCALED AGILE® © Scaled Agile, Inc.

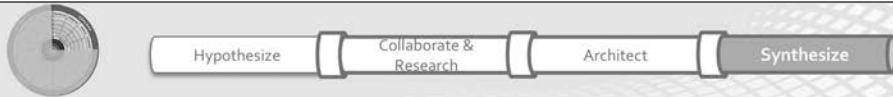
3-22

3.4 Synthesize the Vision, the Roadmap, and the Program Backlog

SCALED AGILE® © Scaled Agile, Inc.

3-23

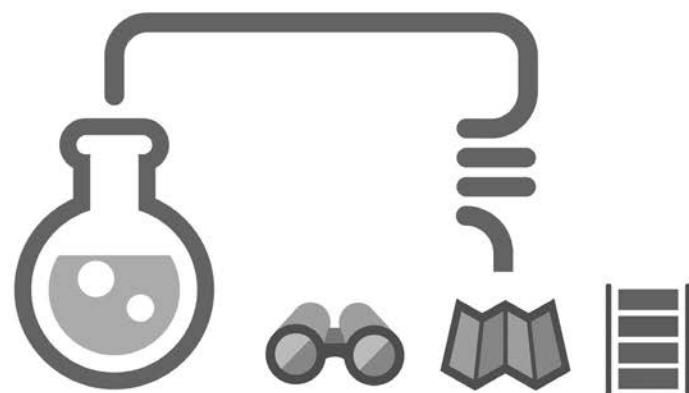
Synthesize



Purpose: Synthesize the hypotheses, research, collaboration, and architecture into a vision, a roadmap, and backlog

Competencies:

- Feature writing
- Behavior-driven development
- Economic prioritization

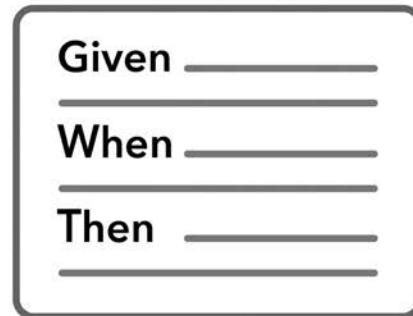


SCALED AGILE® © Scaled Agile, Inc.

3-24

Behavior-Driven Development (BDD)

- ▶ Behavior-driven development is a test-first approach to writing requirements
- ▶ Define the test as the acceptance criteria for a feature
- ▶ A common format is Given-When-Then
- ▶ Permits building executable specifications

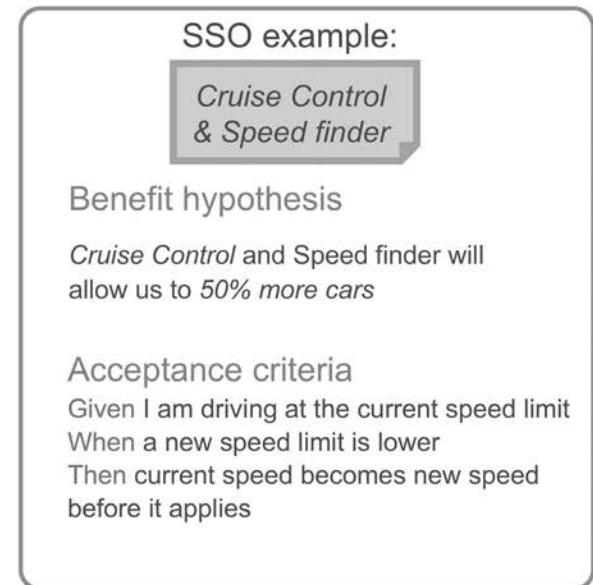


SCALED AGILE® © Scaled Agile, Inc.

3-25

Feature writing

- ▶ 'Feature' is an industry-standard term familiar to marketing and Product Management
- ▶ Benefit hypothesis justifies Feature implementation cost, and provides business perspective when making scope decisions
- ▶ Acceptance criteria are typically defined during Program Backlog refinement
- ▶ Reflects functional and Nonfunctional Requirements (NFRs)
- ▶ Fits in one Program Increment



SCALED AGILE® © Scaled Agile, Inc.

3-26

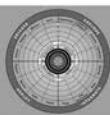
Economic prioritization

- ▶ In a flow system, job sequencing is the key to economic outcomes.
- ▶ Give preference to jobs with *shorter Duration* and *higher Cost of Delay (CoD)*, using *Weighted Shortest Job First (WSJF)*
- ▶ WSJF provides a way of understanding the cost of delay and focusing on items that provide the best cost of delay reduction in the shortest time

$$\text{WSJF} = \frac{\text{Cost of Delay}}{\text{Job Duration} \quad (\text{Job size})}$$

SCALED AGILE® © Scaled Agile, Inc.

3-27



Self assess: Synthesize sub-dimension

- ▶ Individually assess where you are:
 1. CRAWL — The program has a backlog of features.
 2. WALK — The program has a prioritized backlog of features with a benefit hypothesis.
 3. RUN — The program has a backlog of features with a benefit hypothesis, prioritized using WSJF.
 4. FLY — The program has a backlog of features with a benefit hypothesis and BDD acceptance criteria, prioritized using WSJF.
- ▶ Place a dot on the DevOps Health Radar



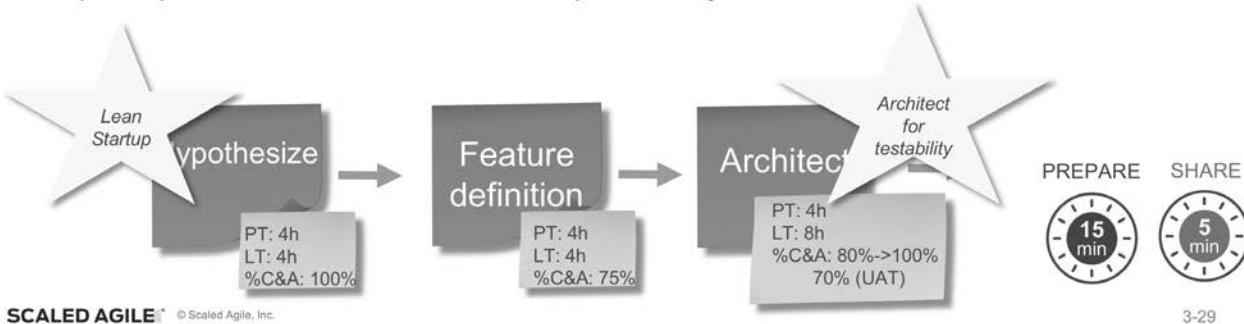
3-28

SCALED AGILE® © Scaled Agile, Inc.



Activity: Thinking about the future of exploration

- ▶ Step 1: Review the competencies we just covered, and discuss which can you use to improve your current-state map.
- ▶ Step 2: Identify improvement opportunities (this might change steps, add steps, or eliminate steps). Then, identify how they will change the Lead Time, Process Time, and % Complete and Accurate.
- ▶ Step 3: Update these on the future-state portion of your transformation canvas.



SCALED AGILE® © Scaled Agile, Inc.

3-29

Lesson review

In this lesson you learned how to:

- ▶ Create solution hypotheses
- ▶ Collaborate and research customer needs
- ▶ Architect the solution for continuous delivery
- ▶ Synthesize the Vision, the Roadmap, and the Program Backlog



SCALED AGILE® © Scaled Agile, Inc.

3-30

SAFe® DevOps

Lesson 4: Building Quality with Continuous Integration

SCALED AGILE® © Scaled Agile, Inc.



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

4-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Develop the solution
- ▶ Build continuously
- ▶ Test end-to-end
- ▶ Validate on a staging environment



SCALED AGILE® © Scaled Agile, Inc.

4-2



Chapter 3: Continuous Integration problems

- ▶ Step 1: Read ‘Chapter 3: Continuous Integration problems.’
- ▶ Step 2: In your group, discuss common challenges with Continuous Integration (CI).
- ▶ Step 3: In one statement, summarize the biggest challenge with CI in your own context.
- ▶ Step 4: Do you share similar CI challenges with other members in your group?



4-3

SCALED AGILE® © Scaled Agile, Inc.

Integrate dimension

Business objective (The ‘Why’)

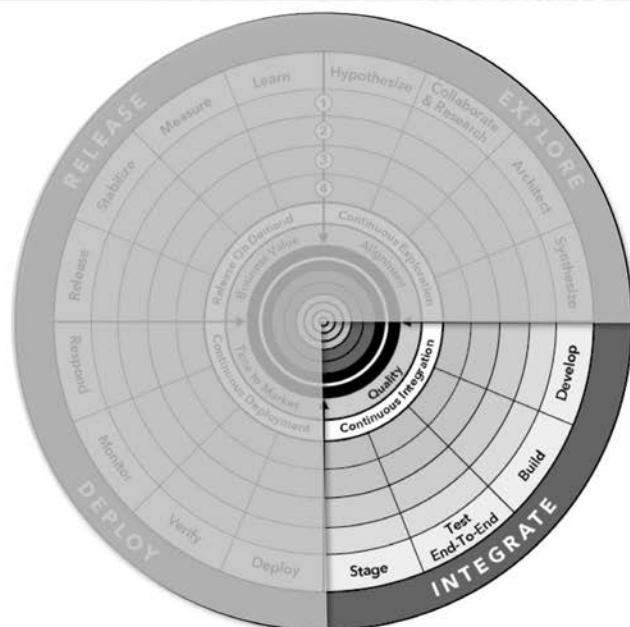
- ▶ Quality

IT objective (The ‘How’)

- ▶ Continuous Integration (CI)

Sub-dimensions (The ‘What’)

- ▶ Develop
- ▶ Build
- ▶ Test end-to-end
- ▶ Stage



SCALED AGILE® © Scaled Agile, Inc.

4-4

4.1 Develop the Solution

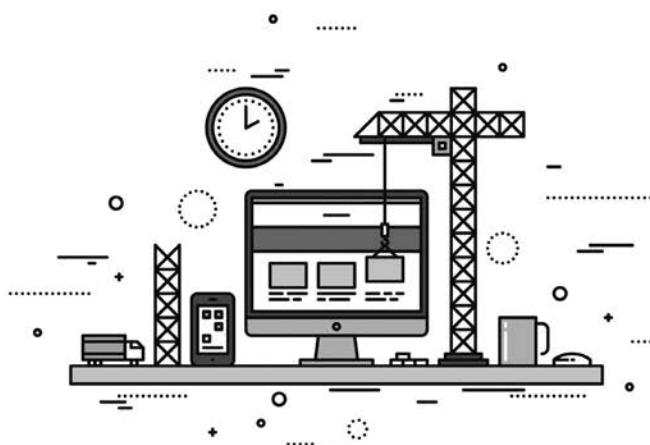
Develop



Purpose: Implement a Story or a part of a Story and commit the code

Competencies:

- ▶ Break features into stories
- ▶ Test-Driven Development
- ▶ Version control
- ▶ Engineering practices
- ▶ Pair work
- ▶ Application telemetry
- ▶ Threat modeling (covered in Architect)

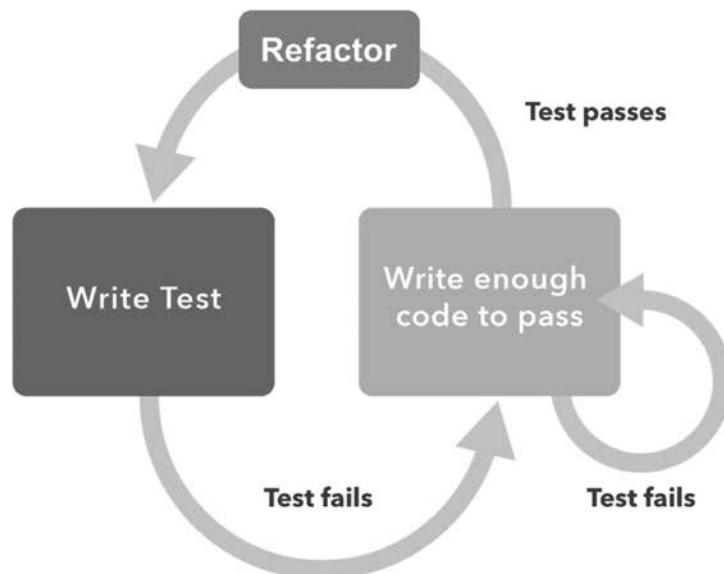


Break Features into Stories

Techniques for splitting Features and Stories to fit within their boundaries (PI and Iteration, respectively).

- 1. Workflow steps
- 2. Business rule variations
- 3. Major effort
- 4. Simple/complex
- 5. Variations in data
- 6. Data methods
- 7. Defer system qualities
- 8. Operations
- 9. Use-case scenarios
- 10. Break out a spike

Test-Driven Development (TDD)



Version control

- ▶ Maintain all assets under version control
- ▶ From requirements, to code, to configuration, to tests, and test data
- ▶ Establish clear check-in and check-out procedures
- ▶ Version control improves traceability for automating compliance



SCALED AGILE® © Scaled Agile, Inc.

4-9

Agile Software Engineering

Focus on engineering practices

- ▶ Test-first
- ▶ Collective ownership
- ▶ Code quality
- ▶ ...



Built-In Quality

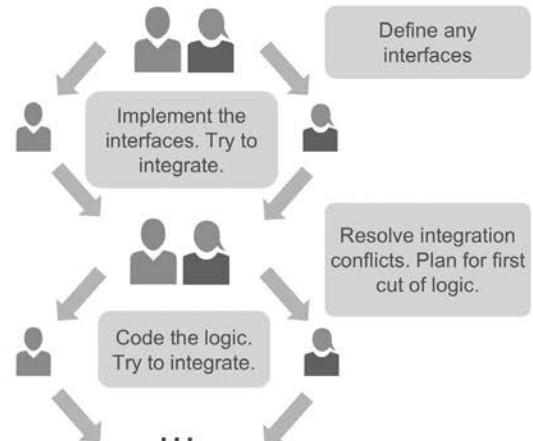
SCALED AGILE® © Scaled Agile, Inc.

4-10

Pair work

Pair work improves system quality, design decisions, knowledge sharing, and team velocity.

- ▶ Pair work is ...
 - A collaborative effort of any two team members: dev/dev, dev/PO, dev/tester, etc.
 - Broader and less constraining than pair programming
- ▶ Team members spend 20% to 80% time pairing
- ▶ Spontaneous pairing, and purposeful rotation over time



Example user Story implementation flow

SCALED AGILE® © Scaled Agile, Inc.

4-11

Application telemetry

Application telemetry enables faster identification of problems from production incidents

- ▶ Telemetry should cover all levels of the code — from methods, to components, to services, to the entire application
- ▶ Application design must take into account operational health telemetry
- ▶ Features must include the ability to measure the benefit hypothesis against both leading and trailing indicators

SCALED AGILE® © Scaled Agile, Inc.

4-12



Self assess: Develop sub-dimension

► Individually assess where you are:

1. **CRAWL** — Individual and team branches persist for longer than an iteration.
2. **WALK** — Code is checked in once an iteration. Unit tests are written some of the time.
3. **RUN** — All code is checked in daily. Unit tests are written after the code.
4. **FLY** — All code is written with TDD and checked to source control multiple times per day. Agile software engineering practices are followed.

► Place a dot on the DevOps Health Radar



4-13

SCALED AGILE® © Scaled Agile, Inc.

4.2 Build continuously

SCALED AGILE® © Scaled Agile, Inc.

4-14

Build



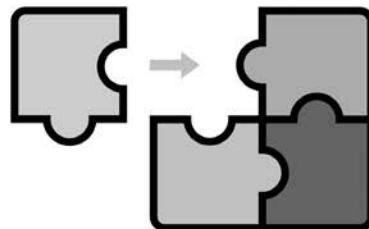
Develop

Build

Test End-To-End

Stage

Purpose: Compile source files into deployable binaries, verify that code functions as the developer(s) intended & merge dev branches to trunk



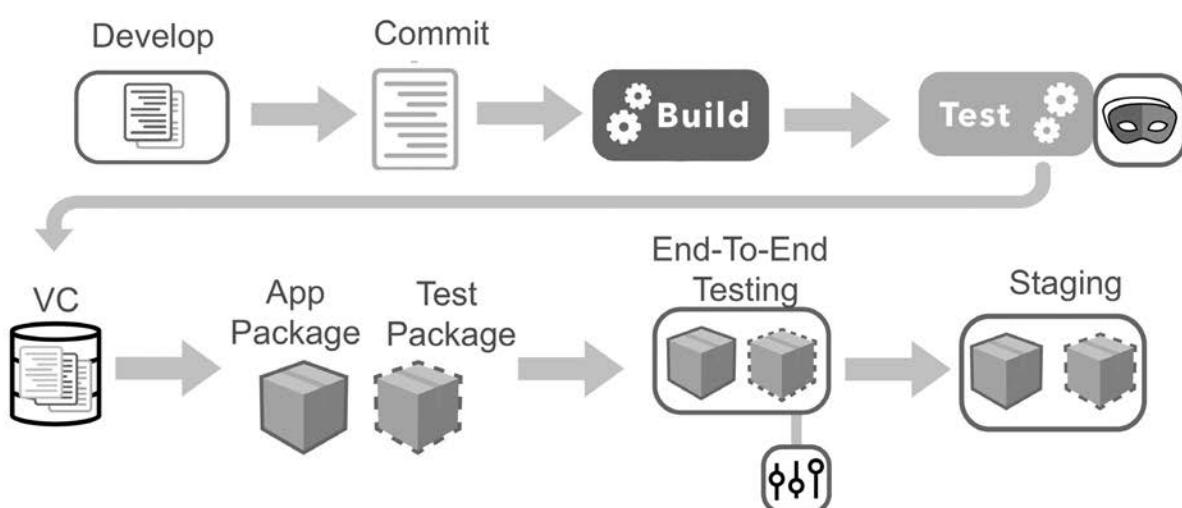
Competencies:

- Continuous code integration
- Build and test automation
- Trunk-based development
- Gated commit
- Application security

SCALED AGILE® © Scaled Agile, Inc.

4-15

Continuous code integration

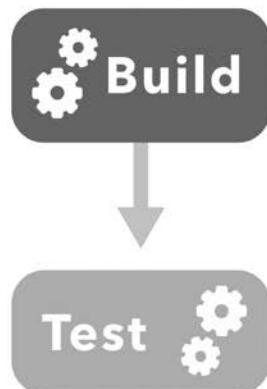


SCALED AGILE® © Scaled Agile, Inc.

4-16

Build and Test automation

- ▶ Initiate a build often, preferably on every commit
- ▶ Run unit tests as part of the build
- ▶ Run static code analysis as part of the build
- ▶ Visualize and monitor the build and test process
- ▶ Report failures immediately
- ▶ Broken builds are the highest priority

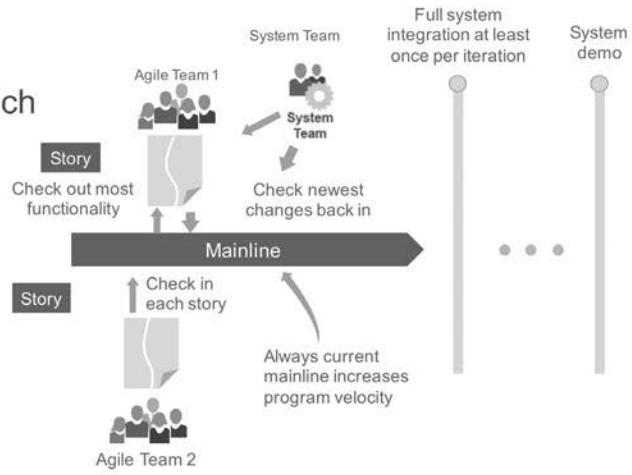


SCALED AGILE® © Scaled Agile, Inc.

4-17

Trunk-based development

- ▶ Single trunk for all teams
- ▶ Commit is always to the single branch
- ▶ Avoid long-lived branches

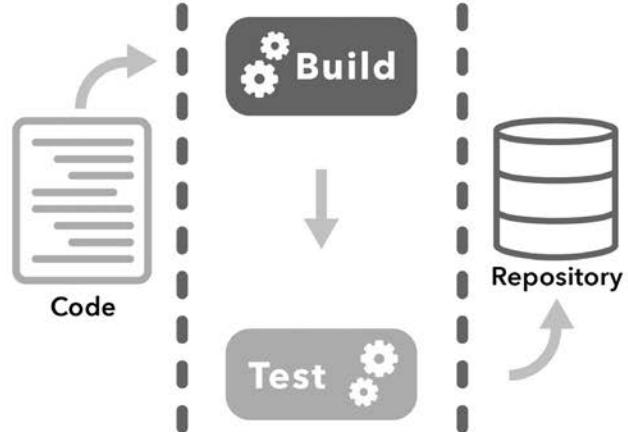


SCALED AGILE® © Scaled Agile, Inc.

4-18

Gated commit

- ▶ Gated commits ensure that broken code doesn't block the rest of the developers or the pipeline
- ▶ Only commits that have made it through the build and test are merged into trunk
- ▶ Alert the person who committed on rejection



SCALED AGILE® © Scaled Agile, Inc.

4-19

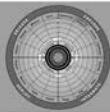
Application security

- ▶ Apply tools to automatically identify security vulnerabilities in the code during the build process
- ▶ Assess open source libraries continuously for known vulnerabilities to identify risks during development or build processes



SCALED AGILE® © Scaled Agile, Inc.

4-20



Self assess: Build sub-dimension

- ▶ Individually assess where you are:

- 1. CRAWL** — Builds are run once every few days or less.
- 2. WALK** — Builds are run automatically once a day and include unit tests. Broken builds are highest priority.
- 3. RUN** — Builds are run automatically after code commit and include unit tests. Broken builds are highest priority.
- 4. FLY** — Builds are run automatically after code commit and run unit test. Static code analysis and security testing. Failed builds are rejected.

- ▶ Place a dot on the DevOps Health Radar.



4-21

SCALED AGILE® © Scaled Agile, Inc.

SCALED AGILE® © Scaled Agile, Inc.

4-22

4.3 Test end-to-end



Video resource: Bing Continuous Delivery (2 of 4) - Microsoft Engineering Stories



DURATION



<https://youtu.be/3sFT7tgyEQk>

4-23

SCALED AGILE® © Scaled Agile, Inc.

Test end-to-end

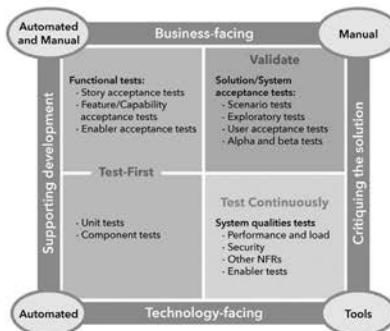


Purpose: To validate changes against acceptance criteria in an integrated, production-simulated environment



Competencies:

- Test and production environment congruity
- Test automation
- Test data management
- Service virtualization
- Nonfunctional requirements (NFRs)



SCALED AGILE® © Scaled Agile, Inc.

4-24

Test and production environment congruity

- ▶ Make sure the test environments match production as much as possible
- ▶ Maintain all configuration changes under version control
- ▶ Service virtualization helps elevate some cost considerations
- ▶ Invest in higher fidelity for more accurate testing

“At Telstra, only 50% of the source code in their development and test environments matched what was running in production.”

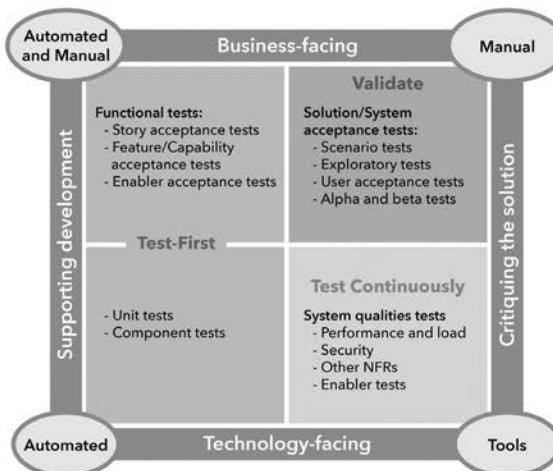
– Em Campbell-Pretty

SCALED AGILE® © Scaled Agile, Inc.

4-25

Test automation

- ▶ Many types of testing need to be run:
 - Functional testing
 - Integration testing
 - Regression testing
 - Performance testing
 - Security testing
 - Exploratory testing
 - Penetration testing
- ▶ Not all tests should be automated

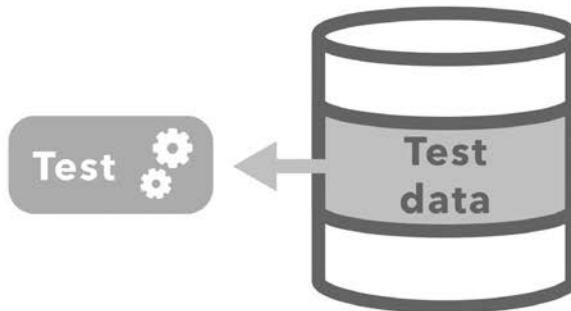


SCALED AGILE® © Scaled Agile, Inc.

4-26

Test data management

- ▶ Data for all types of tests must be managed
- ▶ Store data in a repository for consistent testing
- ▶ Emulate production data to ensure tests reflect realistic situations

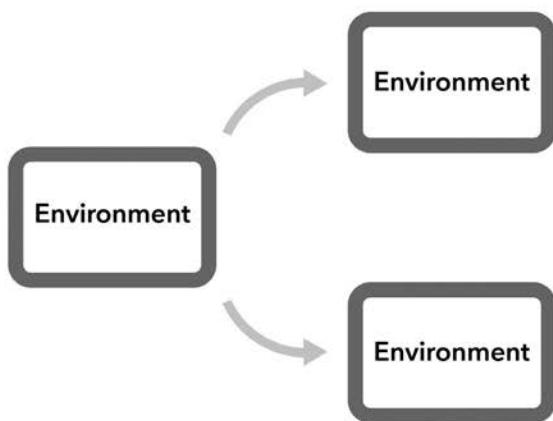


SCALED AGILE® © Scaled Agile, Inc.

4-27

Service virtualization

- ▶ Ability to spawn environments which match production to test
- ▶ Environments that support different types of testing
- ▶ Maintain environment data in source control



SCALED AGILE® © Scaled Agile, Inc.

4-28

Nonfunctional Requirements (NFRs)

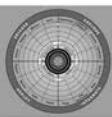
Nonfunctional Requirements are system qualities that support end-user functionality and system goals.

- ▶ Sometimes known as the ‘ilities’— reliability, usability, scalability, availability, etc.
- ▶ Nonfunctional requirements constrain all backlog items



SCALED AGILE® © Scaled Agile, Inc.

4-29



Self assess: Test end-to-end sub-dimension

- ▶ Individually assess where you are:
 1. CRAWL — Mostly manual acceptance tests are run every iteration.
 2. WALK — Mostly automated acceptance tests are run every iteration.
 3. RUN — Successful build triggers automated acceptance tests with production-like data.
 4. FLY — Successful build triggers automated acceptance tests with production-like data from a repository on automatically spawned testing environments.
- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.



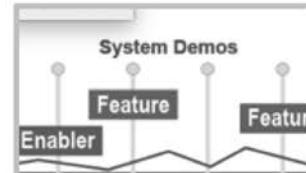
4-30

4.4 Validate on a staging environment

Stage

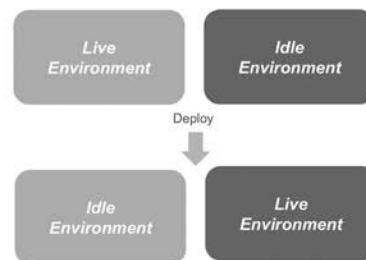


Purpose: Host fully-validated, systems in a production-grade environment, from which they can be deployed to production



Competencies:

- Maintain a staging environment
- Blue/Green deployment
- System demo

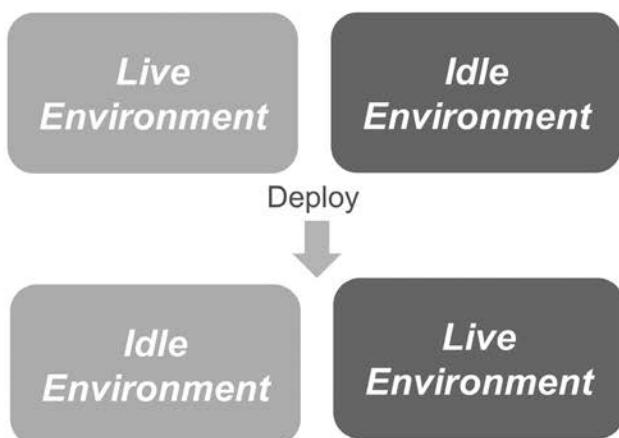


Maintain a staging environment

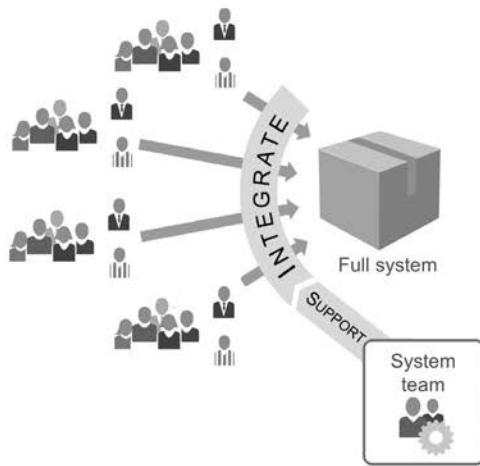
- ▶ Maintain a staging environment that matches production to prepare for moving to production
- ▶ Deploy to staging at least every iteration and run your system demos from there
- ▶ It is preferable to deploy to staging automatically after all build and end-to-end tests have passed

Blue/Green deployment

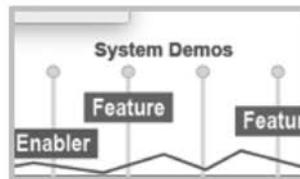
- ▶ Maintain two environments: idle and live
- ▶ Deployment is switching the two environments
- ▶ Switching can be done by redirecting the load balancer



System demo

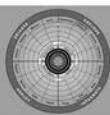


- ▶ Demo working integrated systems every two weeks
- ▶ New Features work together, and with existing functionality
- ▶ Demo from a staging environment that resembles production as much as possible
- ▶ Program stakeholders provide feedback



SCALED AGILE® © Scaled Agile, Inc.

4-35



Self assess: Stage sub-dimension

- ▶ Individually assess where you are:

- 1. CRAWL** — Changes are deployed manually to a staging environment less frequently than once every iteration.
- 2. WALK** — Changes are deployed to a staging environment every iteration.
- 3. RUN** — Whole packages are deployed to a staging environment every iteration. Product Management accepts features on staging.
- 4. FLY** — Whole packages are deployed to a staging environment following acceptance test. Product Management accepts features on staging.

- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.

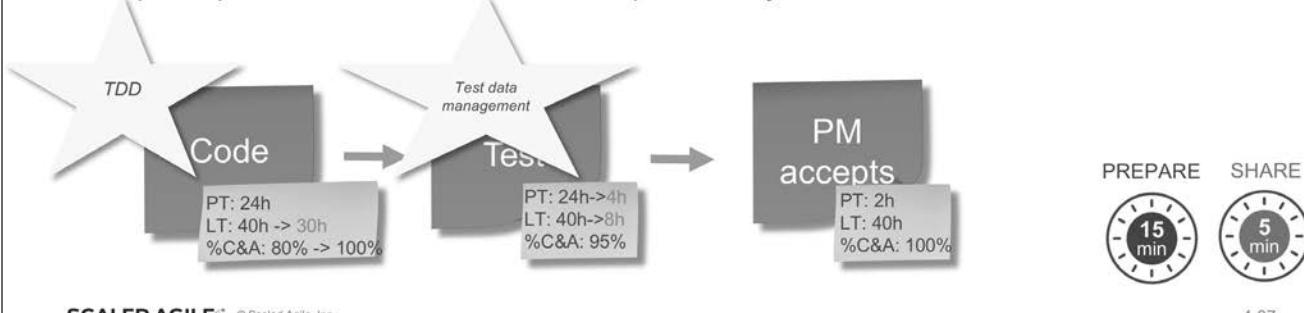


4-36



Thinking about the future of integration

- ▶ Step 1: Review the competencies we just covered, and discuss which can you use to improve your current-state map.
- ▶ Step 2: Identify improvement opportunities (this might change steps, add steps, or eliminate steps). Then, identify how they will change the Lead Time, Process Time, and % Complete and Accurate.
- ▶ Step 3: Update these on the future-state portion of your transformation canvas.



SCALED AGILE® © Scaled Agile, Inc.

4-37

Lesson review

In this lesson you learned how to:

- ▶ Develop the solution
- ▶ Build continuously
- ▶ Test end-to-end
- ▶ Validate on a staging environment



SCALED AGILE® © Scaled Agile, Inc.

4-38

SAFe® DevOps

Lesson 5: Reducing Time-to-Market with Continuous Deployment

SCALED AGILE® © Scaled Agile, Inc.



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

5-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Deploy to production
- ▶ Verify the solution
- ▶ Monitor for problems
- ▶ Respond and recover



SCALED AGILE® © Scaled Agile, Inc.

5-2



Chapter 4: Deployment gone wrong

- ▶ Step 1: Read ‘Chapter 4: Deployment gone wrong.’
- ▶ Step 2: Form pairs and assume the roles of Tracy and Travis.
- ▶ Step 3: Consider some of the immediate solutions suggested by your character. Could any of them be the right course of action?
- ▶ Step 4: Explain to your partner the reasons why the option you chose may or may not be an appropriate solution.



PREPARE SHARE



5-3

SCALED AGILE® © Scaled Agile, Inc.

Deployment dimension

Business objective (The ‘Why’)

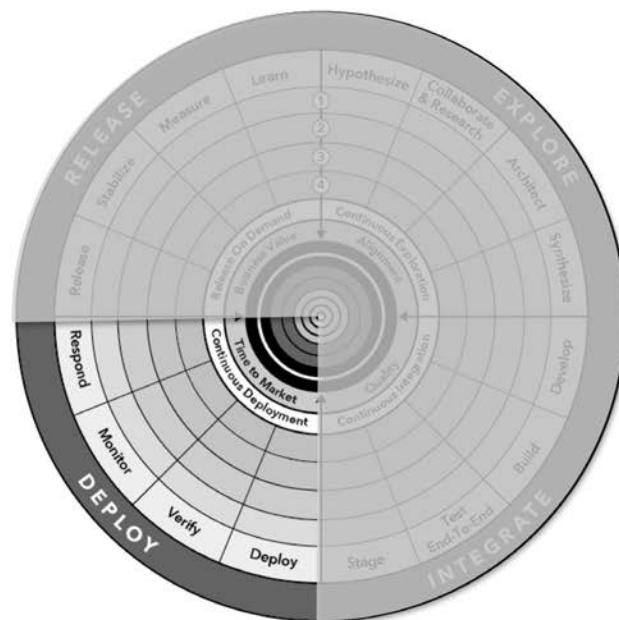
- Time to market

IT objective (The ‘How’)

- Continuous Deployment (CD)

Sub-dimensions (The ‘What’)

- Deploy
- Verify
- Monitor
- Respond

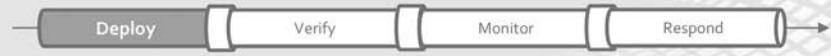


SCALED AGILE® © Scaled Agile, Inc.

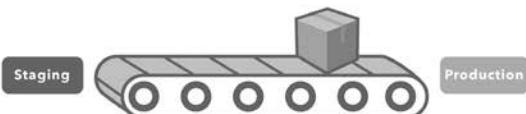
5-4

5.1 Deploy to production

Deploy

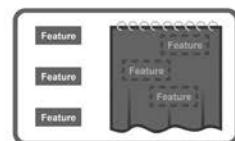


Purpose: Deploy changes into production with high frequency and low risk



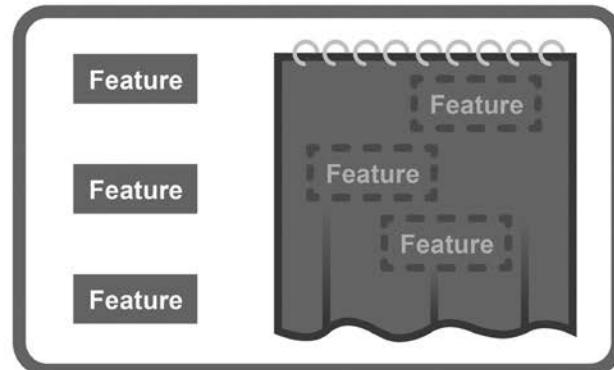
Competencies:

- Dark launches
- Feature toggles
- Deployment automation
- Selective deployment
- Self-service deploys
- Version control (covered in Build)
- Blue/Green deployment (covered in Stage)



Dark launches

- ▶ Separate deploy to production from release
- ▶ Enables testing background and foreground processes in the actual production environment before exposing new functionality to users

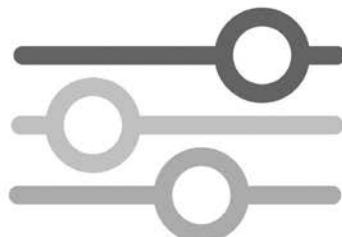


SCALED AGILE® © Scaled Agile, Inc.

5-7

Feature toggles

- ▶ Hide all new functionality under feature toggles
- ▶ Separates the concern of deployment from release
- ▶ Make sure to test 'toggle off' first



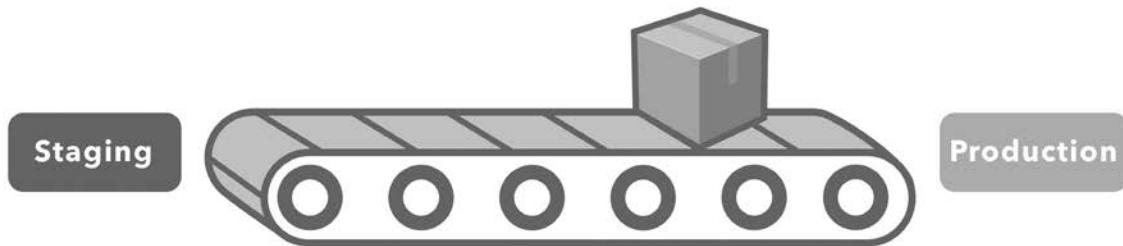
Warning: Be careful of toggle overload and testing complexity

SCALED AGILE® © Scaled Agile, Inc.

5-8

Deployment automation

- ▶ Automate deployment to production
- ▶ Eliminate manual steps from code commit to production deployment
- ▶ Store all environment and package information in version control
- ▶ Test the deployment process itself



SCALED AGILE® © Scaled Agile, Inc.

5-9

Self Service deployment

- ▶ If complete automation from code commit to deployment to production is not possible, automate the deployment of the package to production
- ▶ Enable anyone to safely deploy validated packages
- ▶ Provide simple controls to facilitate what gets deployed

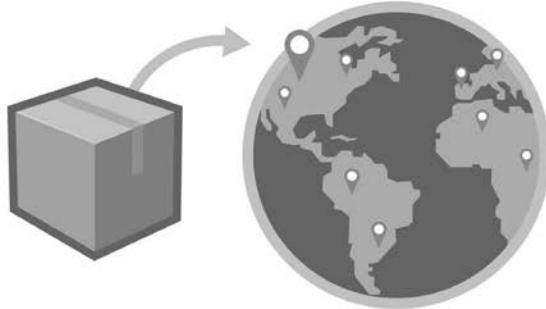


SCALED AGILE® © Scaled Agile, Inc.

5-10

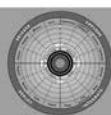
Selective deployment

- ▶ Deploy to select production environments
- ▶ This can be differentiated by data center, geography, customers
- ▶ Allows for a gradual deployment to production for incremental verification and even release



SCALED AGILE® © Scaled Agile, Inc.

5-11



Self assess: Deploy sub-dimension

- ▶ Individually assess where you are:
 - 1. CRAWL** — Changes are deployed to production by operations less frequently than every iteration.
 - 2. WALK** — Changes are deployed to production by operations every iteration.
 - 3. RUN** — Developers deploy code to production every iteration. Release is done on demand.
 - 4. FLY** — Changes are deployed to production immediately after staging. Release is done on demand.
- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.



5-12

5.2 Verify the Solution

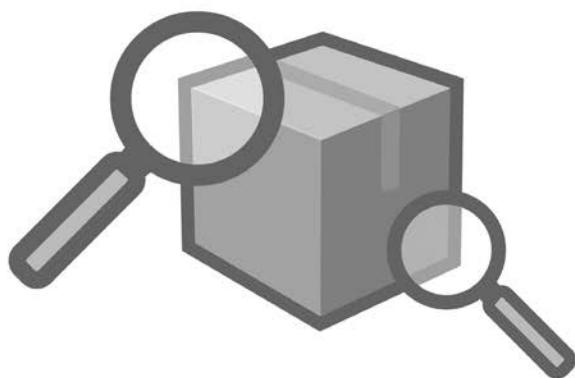
Verify



Purpose: Assure that deployment solutions behave as expected in production before they are released to end users

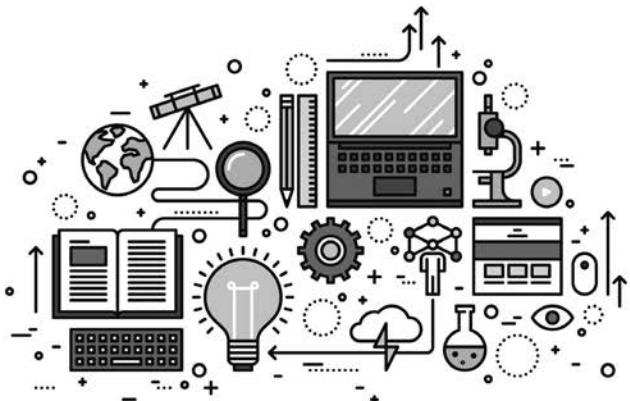
Competencies:

- Production testing
- Test automation (covered in build)
- Test data management (covered in Test End-To-end)
- Nonfunctional requirement (covered in Test End-To-End)



Production Testing

- ▶ Testing of features in the live environment
- ▶ This includes functional and non-functional testing
- ▶ Running synthetic transactions through the services to verify fitness for purpose (utility) and fitness for use (warranty)



SCALED AGILE® © Scaled Agile, Inc.

5-15



Self Assess: Verify sub-dimension

- ▶ Individually assess where you are:
 - 1. Crawl** — Changes are not verified in production.
 - 2. Walk** — Changes are verified in production only when released.
 - 3. Run** — Changes are verified in production using smoke testing.
 - 4. Fly** — Changes are verified in production using smoke testing and production testing with synthetic transactions and penetration testing.
- ▶ Place a dot on the DevOps Health Radar



5-16

SCALED AGILE® © Scaled Agile, Inc.

5.3 Monitor for problems

Monitor



Purpose: Quantitatively measure system and user behavior in real time

Competencies:

- Full-stack telemetry
- Visual displays
- Federated monitoring



Full-stack telemetry

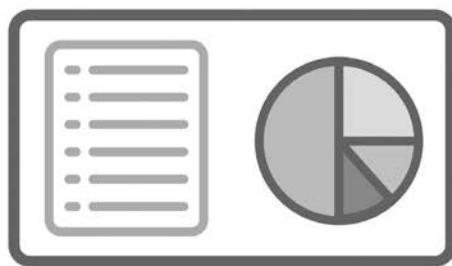
To monitor activities we need proper data

- ▶ Applications should clearly log and report meaningful activities and events
- ▶ Architect applications and infrastructure to support telemetry
- ▶ Monitor both technical data and business data



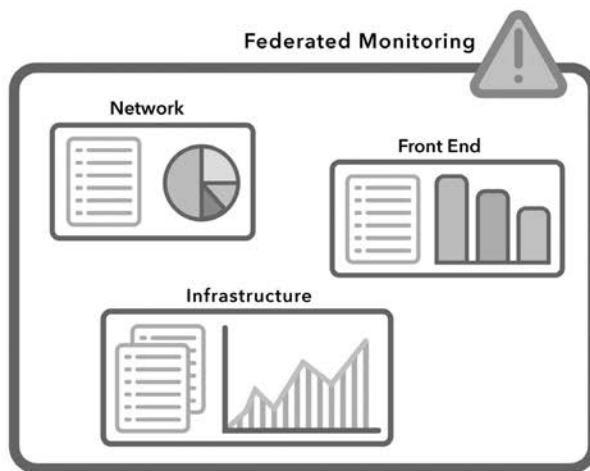
Visual displays

- ▶ Visualize telemetry to the entire organization
- ▶ Big visible information radiators should project the health of the applications and the systems at all times
- ▶ Information about key DevOps metrics should also be visible (ex. — time since last deploy, time since last outage, average lead time, etc.)



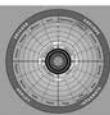
Federated monitoring

- ▶ Aggregate data from various sources into a collection point
- ▶ Build big visual information radiators to display the aggregated data
- ▶ Provide accessibility and ways to drill down into individual application and infrastructure telemetry



SCALED AGILE® © Scaled Agile, Inc.

5-21



Self assess: Monitor sub-dimension

- ▶ Individually assess where you are:
 - 1. CRAWL** — Services report and log on technical activities.
 - 2. WALK** — Services report and log on technical activities. Visual display shows status of technical activities.
 - 3. RUN** — Services report and log on business and technical activities. Visual display shows status to the entire enterprise.
 - 4. FLY** — Services report and log on business and technical activities. Visual display shows status to the entire enterprise. Federated monitoring allows holistic view and log drilldown.
- ▶ Place a dot on the DevOps Health Radar.



SCALED AGILE® © Scaled Agile, Inc.

5-22

5.4 Respond and recover

Respond



Purpose: Proactively detect and resolve production issues before they cause business disruption

Competencies:

- Proactive detection
- Cross-team collaboration
- Session replay
- Rollback and fix forward
- Immutable infrastructure
- Version control (covered in Build)





Activity: Preparing for outages

Step 1: Read the blog post in your Workbook

Step 2: Discuss in your group:

- ▶ How did Netflix prepare for the AWS outage?
- ▶ What are you doing in your organization to prepare?
- ▶ What can we learn from Netflix?



PREPARE SHARE



SCALED AGILE® © Scaled Agile, Inc.

5-25

Proactive detection

- ▶ Decoupling deployment from release allows problem detection before problems are exposed to customers
- ▶ Proactively look for problems, and practice disaster and recovery situations
- ▶ Self sabotage, like chaos monkey, helps build resilience
- ▶ These should be coordinated practices in high-assurance environments, along with failure modes and effect analysis (FMEA) during architecture and design



SCALED AGILE® © Scaled Agile, Inc.

5-26

Cross-team collaboration

- ▶ Dealing with production issues is everyone's responsibility
- ▶ Having a team that can develop and support is preferred
- ▶ Teams from across the value stream should collaborate on solving production issues and identifying root causes

Dev Ops



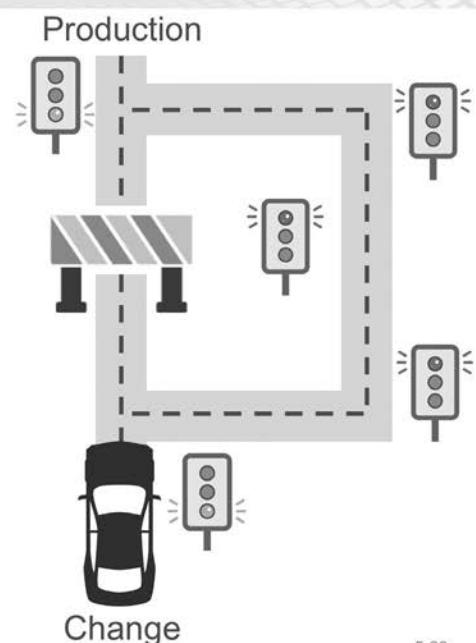
SCALED AGILE® © Scaled Agile, Inc.

5-27

Immutable infrastructure

- ▶ Making changes directly in the production environment creates configuration drifts and has inherent risks
- ▶ In an immutable infrastructure environment all changes are deployed through the Continuous Delivery Pipeline.

Make no changes directly in production



SCALED AGILE® © Scaled Agile, Inc.

5-28

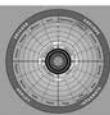
Session replay

- ▶ Record customer sessions and replay them to test problems
- ▶ Make session replay available in production, testing, and development environments
- ▶ Consider security and privacy when implementing this capability



SCALED AGILE® © Scaled Agile, Inc.

5-29



Self assess: Respond sub-dimension

- ▶ Individually assess where you are:
 1. **CRAWL** — Ops owns production issues; development involvement requires escalation.
 2. **WALK** — Shared ownership of production issues.
 3. **RUN** — Proactive detection is used to identify problems. Shared ownership of production issues.
 4. **FLY** — Proactive detection is used to identify problems. Shared ownership of production issues. Changes to production are done through the pipeline.
- ▶ Place a dot on the DevOps Health Radar.



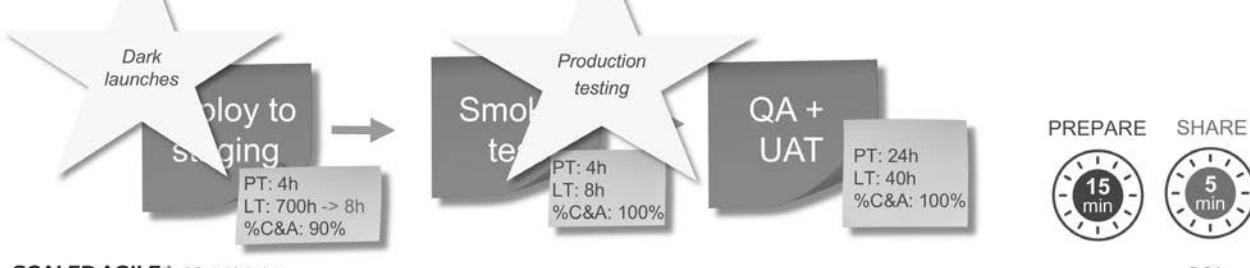
SCALED AGILE® © Scaled Agile, Inc.

5-30



Activity: Thinking about the future of deploy

- ▶ Step 1: Review the competencies we just covered, and discuss which can you use to improve your current-state map.
- ▶ Step 2: Identify improvement opportunities (this might change steps, add steps, or eliminate steps). Then, identify how they will change the Lead Time, Process Time, and % Complete and Accurate.
- ▶ Step 3: Update these on the future-state portion of your transformation canvas.



SCALED AGILE® © Scaled Agile, Inc.

5-31

Lesson review

In this lesson you learned how to:

- ▶ Deploy to production
- ▶ Verify the solution
- ▶ Monitor for problems
- ▶ Respond and recover



SCALED AGILE® © Scaled Agile, Inc.

5-32

SAFe® DevOps

Lesson 6: Delivering Business Value with Release on Demand



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

SCALED AGILE® © Scaled Agile, Inc.

6-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Release on demand
- ▶ Stabilize the solution
- ▶ Measure the business value
- ▶ Learn and react



SCALED AGILE® © Scaled Agile, Inc.

6-2



Chapter 5: What did we release?

- ▶ Step 1: Read ‘Chapter 5: What did we release?’
- ▶ Step 2: Review the clues provided by the characters.

What are some of the actions you could take (considering your own context), to prevent situations like the one at DynotransX from occurring so often.



PREPARE SHARE



6-3

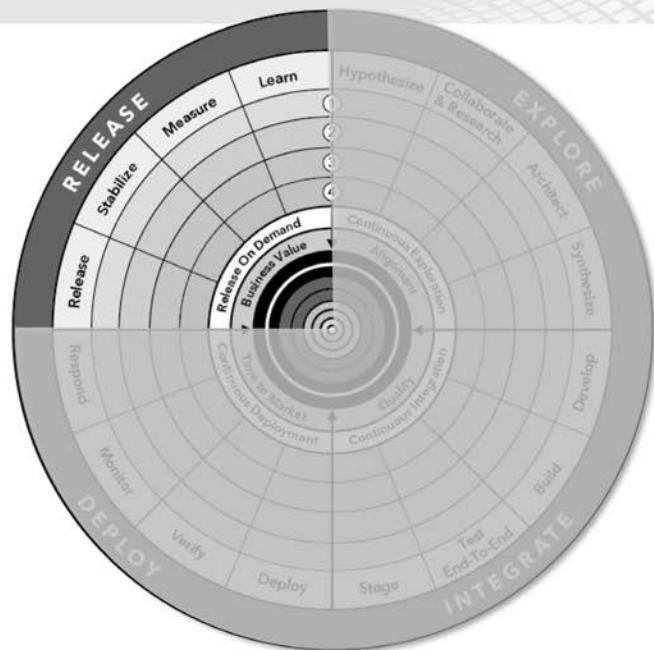
SCALED AGILE® © Scaled Agile, Inc.

Release dimension

Business objective (The ‘Why’)
 - Business value

IT objective (The ‘How’)
 - Release on Demand

Sub-dimensions (The ‘What’)
 - Release
 - Stabilize
 - Measure
 - Learn



SCALED AGILE® © Scaled Agile, Inc.

6-4

6.1 Release on demand

SCALED AGILE® © Scaled Agile, Inc.

6-5

Release



Purpose: Release value to customers all at once or incrementally

Competencies:

- Feature toggles (covered in Deploy)
- Canary releases
- Decouple release elements (covered in Architect)
- Dark launches (covered in Deploy)

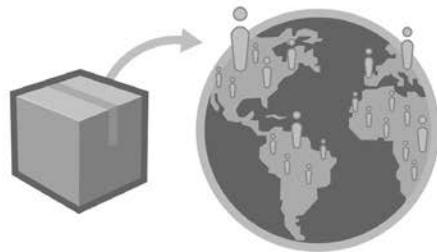


SCALED AGILE® © Scaled Agile, Inc.

6-6

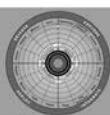
Canary releases

- ▶ Provide the ability to release value to part of the user population, be it internal or external
- ▶ Add or remove user segments based on business decisions
- ▶ Combine with selective deployments to enable incremental deployment and rollout



SCALED AGILE® © Scaled Agile, Inc.

6-7



Self assess: Release sub-dimension

- ▶ Individually assess where you are:
 - 1. CRAWL** — Release and deployment are coupled.
 - 2. WALK** — Release and deployment are coupled but occur every PI.
 - 3. RUN** — Release is decoupled from deployment. Business can decide to release features on demand. Toggles are refactored when done.
 - 4. FLY** — Release is decoupled from deployment. Business can decide to release features on demand and to selected audiences. Toggles are refactored when done.
- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.



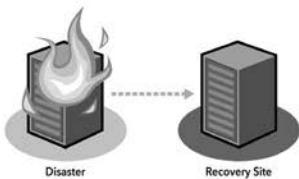
6-8

6.2 Stabilize the Solution

Stabilize



Purpose: Assure sustainably high levels of business continuity, application service levels, and data protection



Competencies:

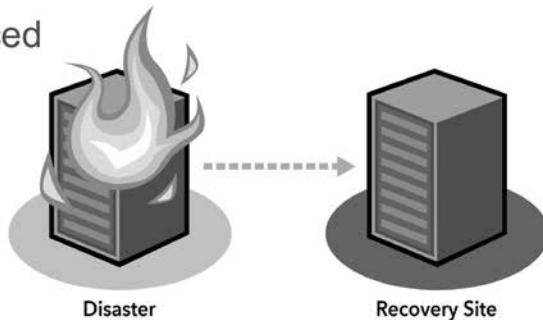
- Failover/disaster recovery
- Continuous security monitoring
- Architect for operations (covered in Architect)
- Nonfunctional requirements (covered in Test End-To-End)



Failover/disaster recovery

Failures will occur. Develop the ability to recover quickly.

- ▶ Failover mechanism allows service to resume quickly, or even avoid service interruption
- ▶ Disaster recovery must be planned, architected into the service and practiced

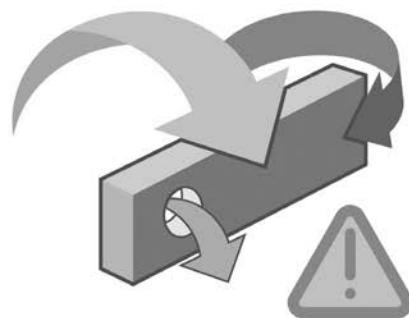


SCALED AGILE® © Scaled Agile, Inc.

6-11

Continuous security monitoring

- ▶ Detect intrusions and attacks on services and infrastructure
- ▶ Security as code and penetration testing focus on preventing known vulnerabilities from getting to production
- ▶ Test services continuously for newly discovered and reported vulnerabilities



Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring

SCALED AGILE® © Scaled Agile, Inc.

6-12



Self assess: Stabilize sub-dimension

► Individually assess where you are:

1. **CRAWL** — Features receive no special attention after release.
2. **WALK** — Features are monitored after release.
3. **RUN** — Features are monitored after release. Failover and recovery processes are in place.
4. **FLY** — Features are monitored after release. Failover and recovery processes are in place. Continuous security monitoring is done.

PREPARE



6-13

► Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.

SCALED AGILE® © Scaled Agile, Inc.

6-14

6.3 Measure the business value

Measure



Purpose: Test hypothesis of business value, and customers are delighted with delivered solution before Operations takes custody

Competencies:

- Innovation accounting (covered in Hypothesize)
- Evaluate hypothesis



SCALED AGILE® © Scaled Agile, Inc.

6-15

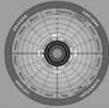
Evaluate hypothesis

- ▶ Application telemetry creates a way to evaluate the business results of a hypothesis
- ▶ Measure both leading and lagging indicators
- ▶ Build the ability to identify the correlation between business results and the hypothesis being tested



SCALED AGILE® © Scaled Agile, Inc.

6-16



Self assess: Measure sub-dimension

- ▶ Individually assess where you are:

- 1. CRAWL** — Feature effectiveness is not measured.
- 2. WALK** — Data about the business is constantly monitored.
- 3. RUN** — Data about the business is constantly monitored. Hypotheses data is gathered.
- 4. FLY** — Data about the business is constantly monitored. Correlation between deployment, releases, and business data is available. Hypotheses data is gathered.

- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.

PREPARE



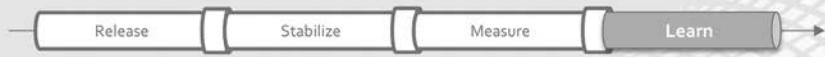
6-17

6.4 Learn and react

SCALED AGILE® © Scaled Agile, Inc.

6-18

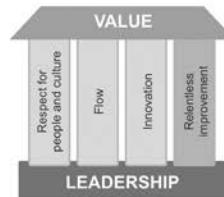
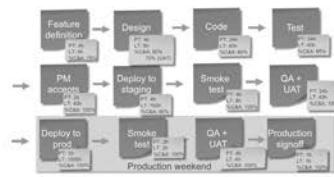
Learn



Purpose: Learn from the hypothesis on whether to pivot or persevere, as well as how to improve the flow of value

Competencies:

- Lean startup (covered in Hypothesize)
- Relentless Improvement
- Value stream mapping

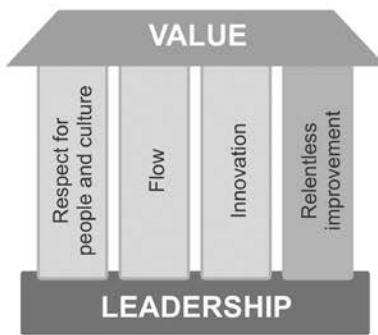


SCALED AGILE® © Scaled Agile, Inc.

6-19

Relentless improvement

- ▶ In order to improve the ability to test hypotheses, the Continuous Delivery Pipeline must be constantly maintained
- ▶ Team-level retrospectives and program and solution-level Inspect and Adapt events are crucial to improve the flow
- ▶ Focus on the root cause of bottlenecks and research incidents to identify the most important place to improve

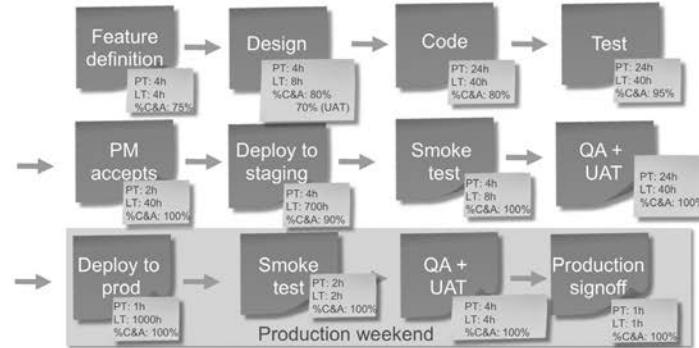


SCALED AGILE® © Scaled Agile, Inc.

6-20

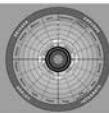
Value stream mapping

- ▶ Continuously apply Value stream mapping to reduce time to market
- ▶ Apply current state mapping and future state mapping so objectives are clear



SCALED AGILE® © Scaled Agile, Inc.

6-21



Self assess: Learn sub-dimension

- ▶ Individually assess where you are:
 - 1. CRAWL** — Hypotheses are seldom evaluated.
 - 2. WALK** — Continuous improvement of the delivery pipeline occurs every program increment.
 - 3. RUN** — Hypotheses data are evaluated. Pivot or preserve decisions are made without mercy or guilt. Continuous improvement of the delivery pipeline occurs every program increment.
 - 4. FLY** — Hypotheses data are evaluated. Pivot or preserve decisions are made without mercy or guilt. Continuous improvement of the delivery pipeline occurs every program increment. Value stream mapping is done at least once a year.
- ▶ Place a dot on the DevOps Health Radar.

SCALED AGILE® © Scaled Agile, Inc.

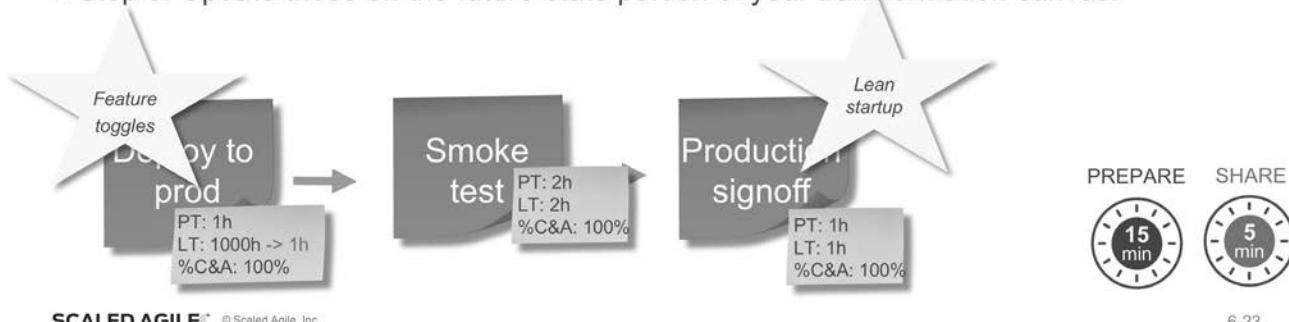


6-22



Thinking about the future of release

- ▶ Step 1: Review the competencies we just covered, and discuss which can you use to improve your current-state map.
- ▶ Step 2: Identify improvement opportunities (this might change steps, add steps, or eliminate steps). Then, identify how they will change the Lead Time, Process Time, and % Complete and Accurate.
- ▶ Step 3: Update these on the future-state portion of your transformation canvas.



Lesson review

In this lesson you learned how to:

- ▶ Release on demand
- ▶ Stabilize the solution
- ▶ Measure the business value
- ▶ Learn and react



SAFe® DevOps

Lesson 7: Taking Action



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

SCALED AGILE® © Scaled Agile, Inc.

7-1

Learning objectives

At the end of this lesson, you should be able to:

- ▶ Share your DevOps action plan



SCALED AGILE® © Scaled Agile, Inc.

7-2

7.1 Share your DevOps action plan



Prioritize your DevOps backlog

- ▶ Step 1: Write down all the improvement items from your board in your workbook.
- ▶ Step 2: Identify items that are outside the scope of things you can change and document them in the 'boundaries and limitations' section of the DevOps transformation canvas.
- ▶ Step 3: Identify the item that offers the least overall improvement based on the changes to lead time, process time, or percent complete and accurate. Mark that as '1' in the cost of delay column.
- ▶ Step 4: Estimate the cost of delay for all other items relative to the 1, using the numbers 1,2,3,5,8,13,20.
- ▶ Step 5: Estimate the job size of the items in a similar fashion; smallest job size again being a '1'.
- ▶ Step 6: Calculate WSJF.
- ▶ Step 7: Document the three highest priorities in the 'improvement items' section of the DevOps transformation canvas.
- ▶ Step 8: Share with the class.

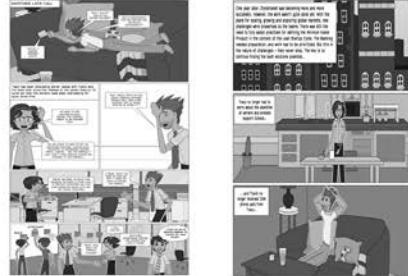
PREPARE SHARE





Chapter 6: The end

- ▶ Step 1: Read Chapter 6 ‘The end.’
- ▶ Step 2: Discuss as a team:
 - What are some of the future challenges you foresee for you and your team while on the path to DevOps transformation?
 - What are some of your takeaways from the story about Travis and Tracy, and the DynotransX, Inc. transformation?



PREPARE SHARE



7-5

SCALED AGILE® © Scaled Agile, Inc.

Take action after the class

- Share the DevOps transformation canvas and your insights with your peers
- Refine the Value Stream Map with your stakeholders
- Enter your top three improvements into your Program Backlog
- Enhance your software engineering practices

SCALED AGILE® © Scaled Agile, Inc.

7-6

Lesson review

In this lesson you:

- ▶ Shared your DevOps action plan



SAFe® DevOps

Lesson 8: About the Exam

SCALED AGILE® © Scaled Agile, Inc.



SAFe® Authorized Course Attending this course gives students access to the SAFe DevOps Practitioner exam and related preparation materials.

8-1

Welcome to Role-based learning and certification

Scaled Agile's Role-based learning and certification offerings focus on the skills, knowledge, and experience required to successfully perform the job.

As part of your SAFe learning journey, we encourage you to attend courses, read recommended books and articles, take advantage of videos and enablement, gain some real-world experience in the role, and then take the exams.

*We look forward to you becoming
a member of the SAFe Certified Community!*



SCALED AGILE® © Scaled Agile, Inc.

8-2

Get SAFe certified in 1-2-3

1

Take the course

- Course completion is the first step towards SAFe certification!
- After course completion you will receive access to the SAFe Community Platform, which also gives you access to study materials and the exam.

SCALED AGILE® © Scaled Agile, Inc.

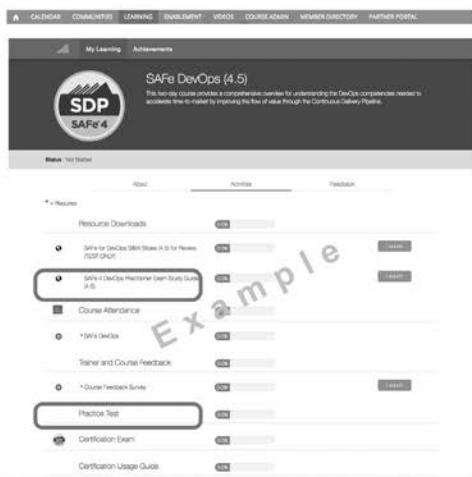
8-3

Get SAFe certified in 1-2-3

2

Study and learn

- An exam study guide and a practice test is available on the **SAFe Community Platform** in your Learning Plan. Use these assets to help you prepare for the exam and to further your learning
- A sample test is publicly available for all exams on **scaledagile.com** on each course page under Exam Details



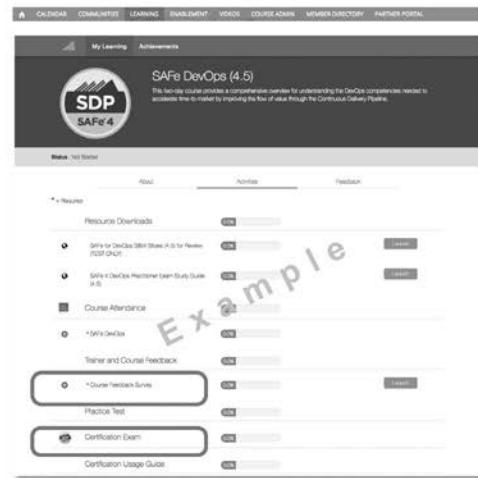
SCALED AGILE® © Scaled Agile, Inc.

8-4

Get SAFe certified in 1-2-3

3 Take the exam

- The exam is included in your Learning Plan and is available after completing the Course Feedback Survey.
- First exam attempt is included as part of the course registration fee if the exam is taken within 30 days of course completion. Each retake attempt costs \$50.
- Exam information including duration, number of questions, passing score, and a sample test is publicly available for all exams on scaledagile.com on each course page under Exam Details.



SCALED AGILE® © Scaled Agile, Inc.

8-5



Video resource: What's next?

Want to learn more about the next steps on your SAFe Journey?

Watch the short video at
www.vimeo.com/201877314

**The next steps on your
SAFe Journey**



SCALED AGILE® © Scaled Agile, Inc.

8-6