

SAFe® DevOps

Optimizing Your Value Stream

5.0

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





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 exam



Become a Certified SAFe Professional

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



Access SAFe Content and Tools

Access professional development resources, and your trainer enablement plan to teach SAFe instructor-led classes



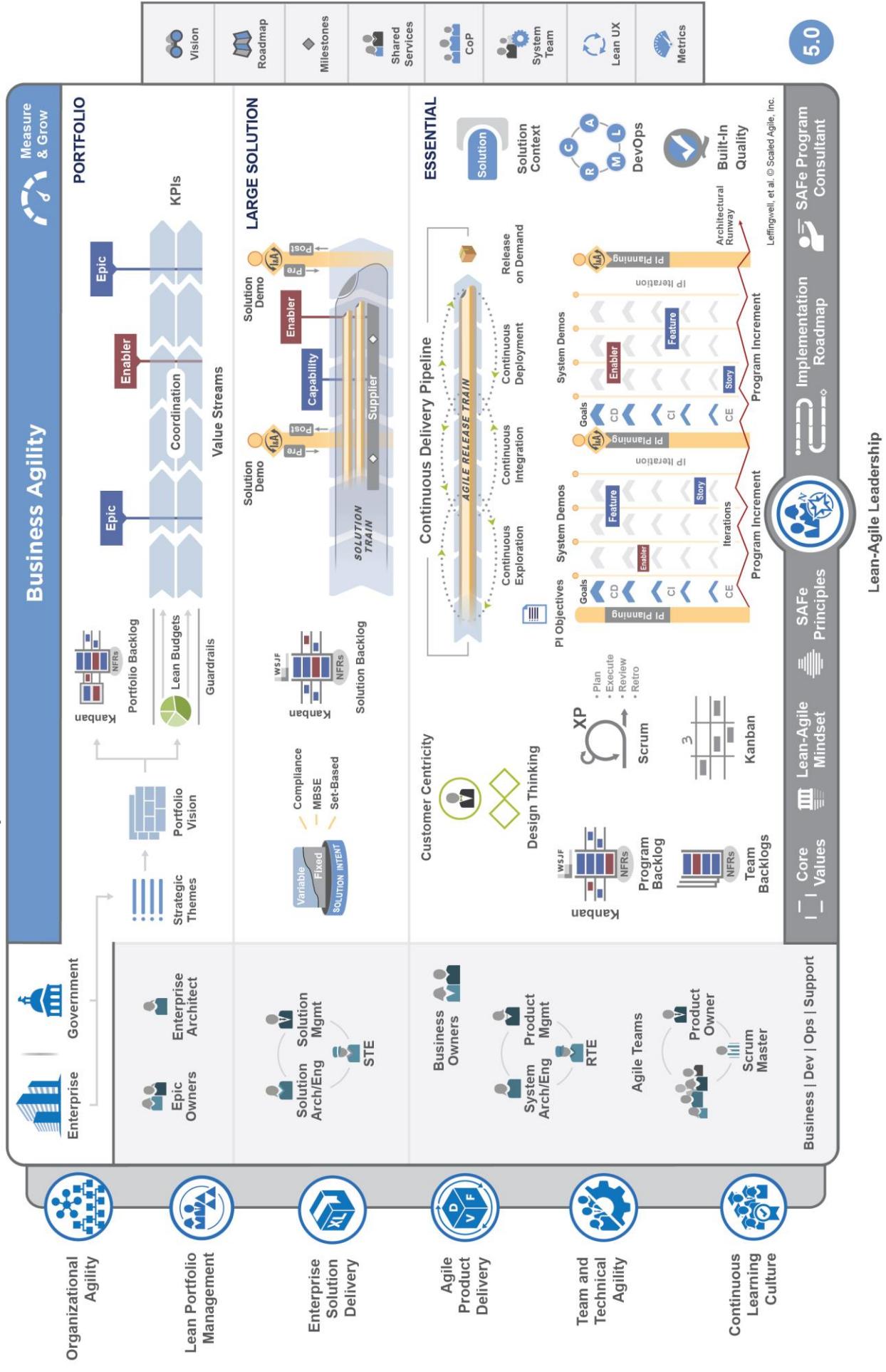
Showcase SAFe Credentials

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® for Lean Enterprises

Full Configuration

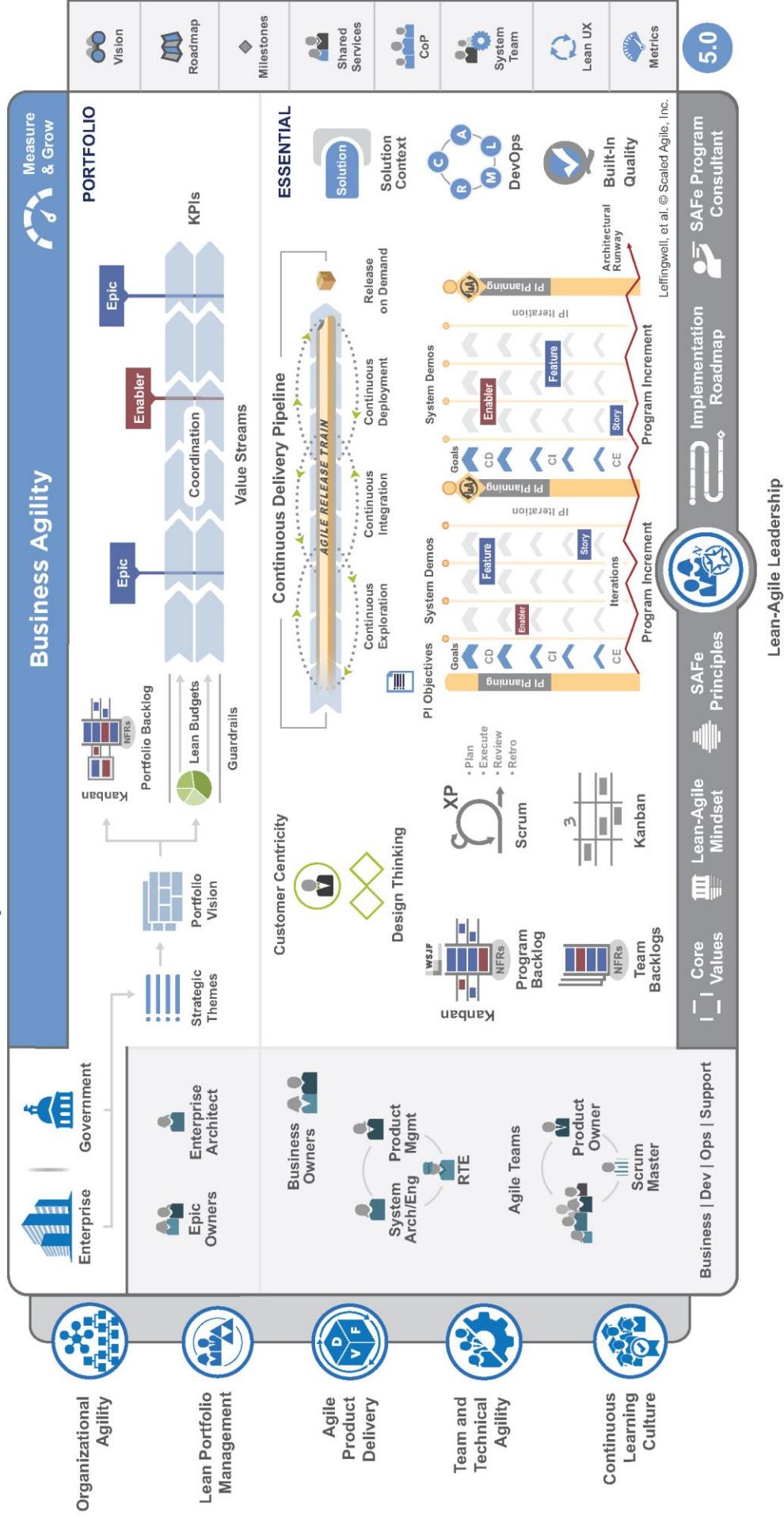


5.0

SAFe® PROVIDED BY SCALED AGILE INC.

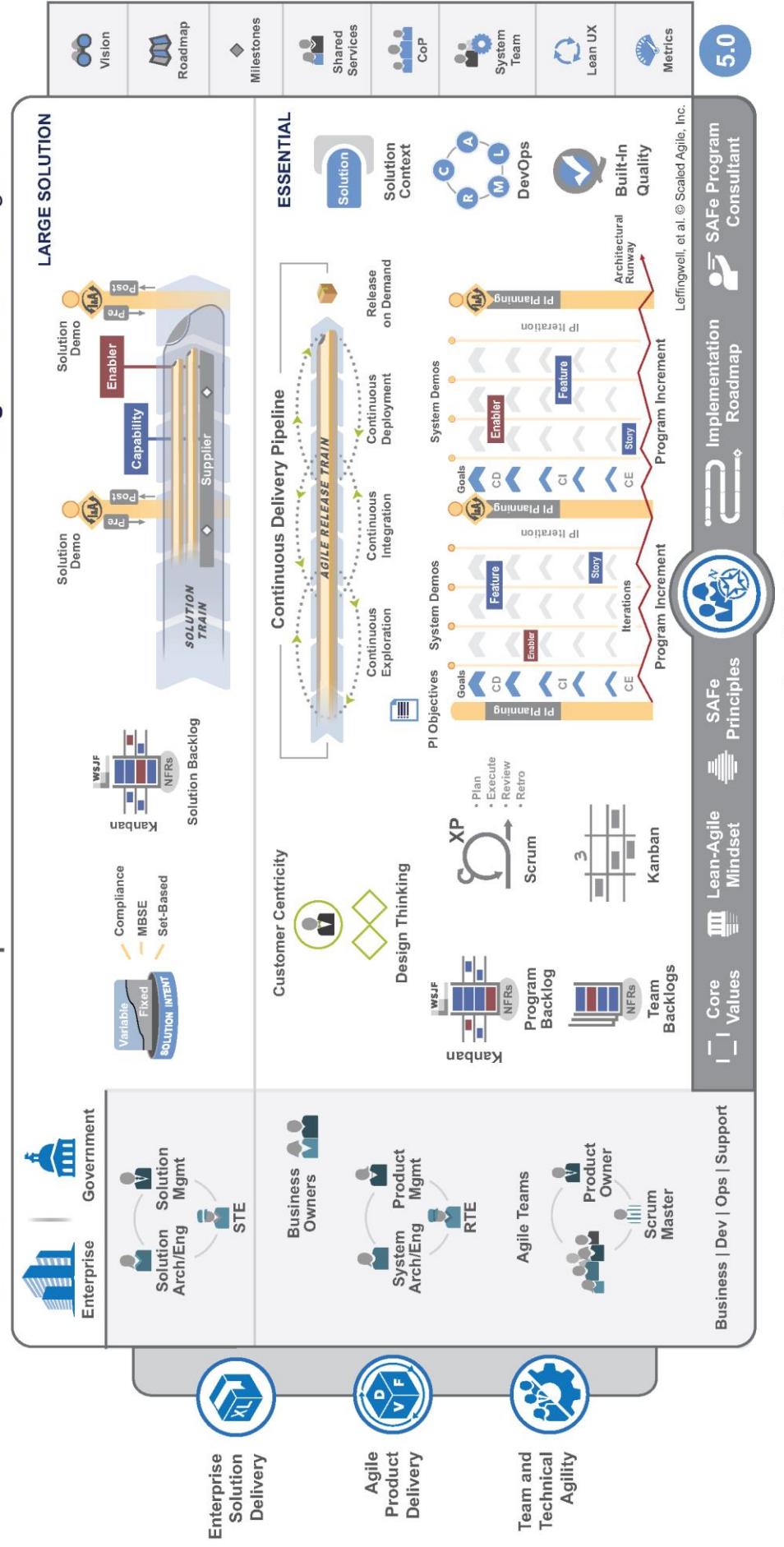
SAFe® for Lean Enterprises

Portfolio Configuration



5.0

SAFe® for Lean Enterprises



Lean-Agile Leadership

5.0

Leffingwell, et al. © Scaled Agile, Inc.

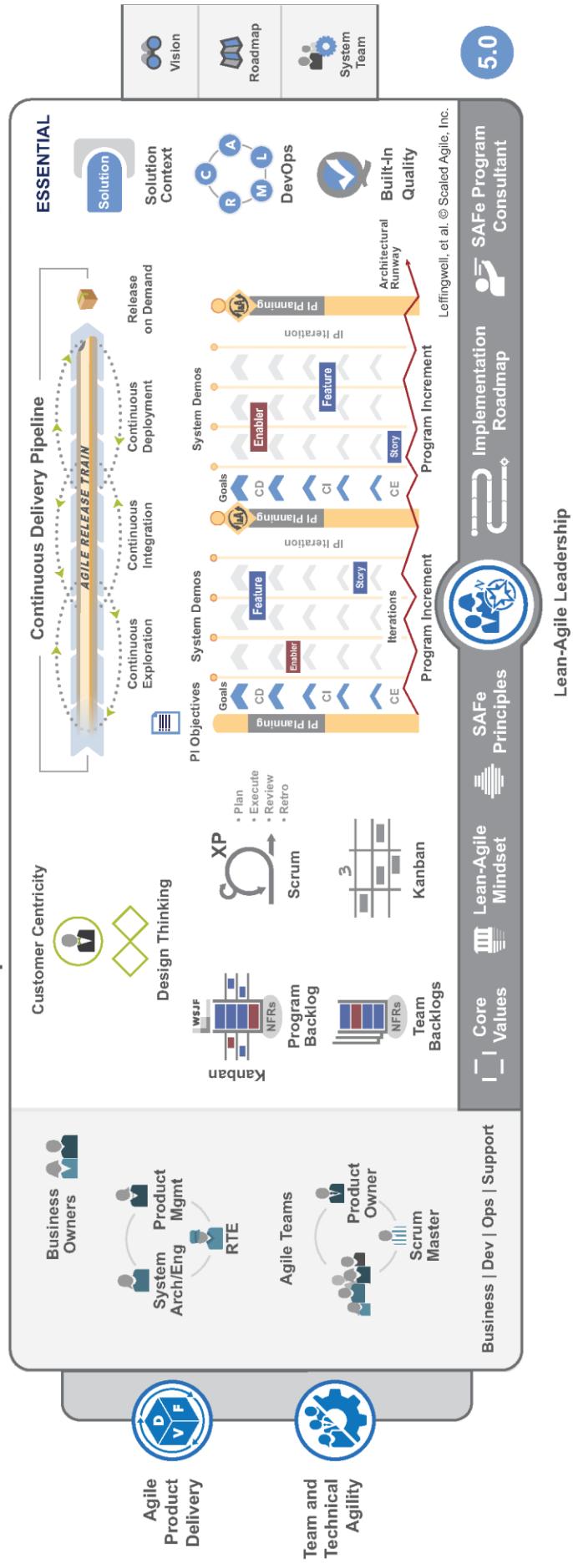
SAFE Program Consultant

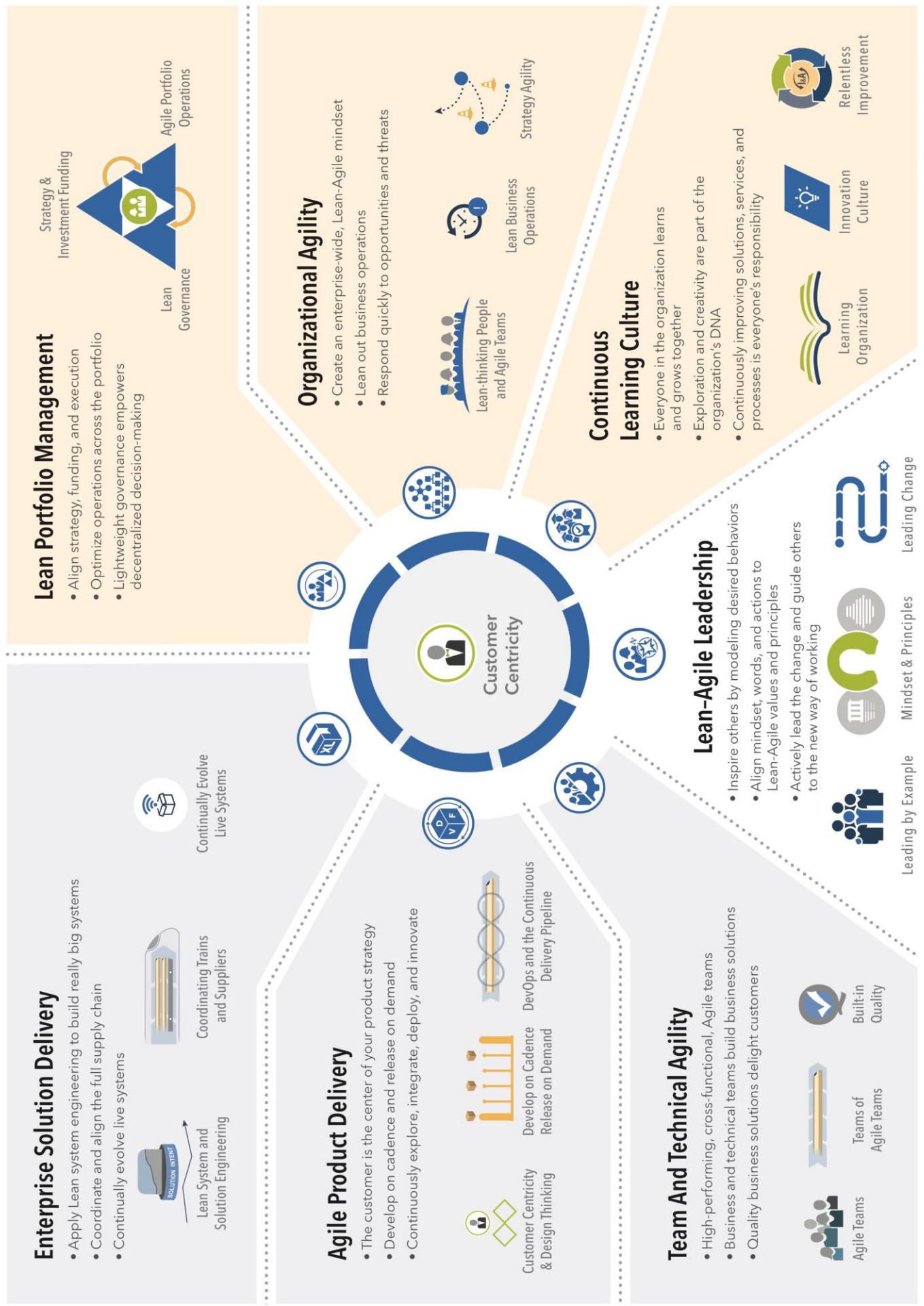
Implementation Roadmap

Lean-Agile Mindset

SAFe Principles

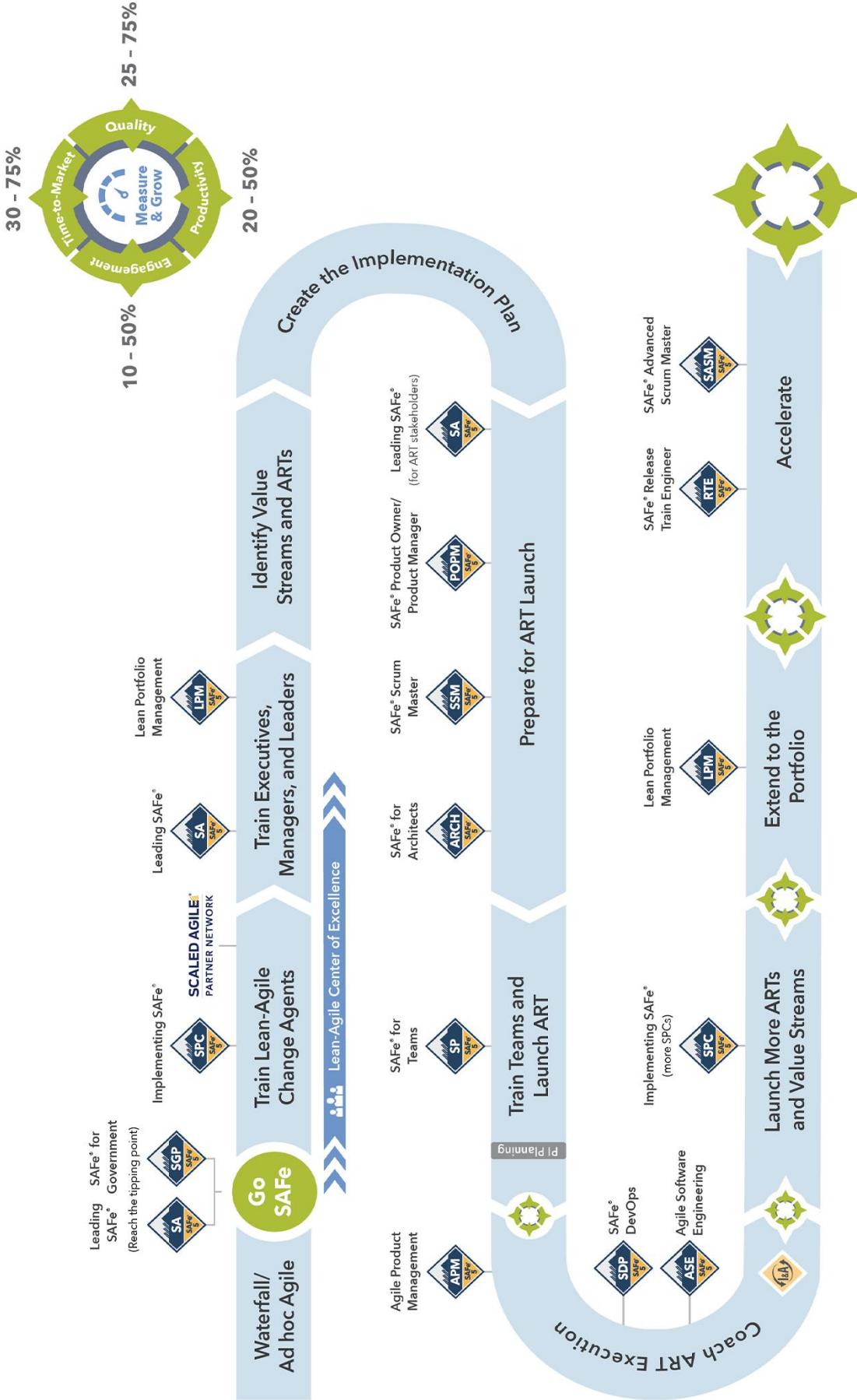
SAFe® for Lean Enterprises





SAFe® Implementation Roadmap

Business results



SAFe® Courses and Certifications

Course	Description	Certification
Leading SAFe®	Thriving in the Digital Age with Business Agility	 with SAFe® 5 Agilist Certification
Implementing SAFe®	Achieving Business Agility with the Scaled Agile Framework	 with SAFe® 5 Program Consultant Certification
SAFe® for Government	Applying Lean-Agile Practices in the Public Sector with SAFe®	 with SAFe® 5 Government Practitioner Certification
Lean Portfolio Management	Aligning Strategy with Execution	 with SAFe® 5 Lean Portfolio Manager Certification
SAFe® Product Owner/Product Manager	Delivering Value through Effective Program Increment Execution	 with SAFe® 5 Product Owner/Product Manager Certification
Agile Product Management	Using Design Thinking to Create Valuable Products in the Lean Enterprise	 with SAFe® 5 Agile Product Manager Certification
SAFe® Scrum Master	Applying the Scrum Master Role within a SAFe® Enterprise	 with SAFe® 5 Scrum Master Certification
SAFe® Advanced Scrum Master	Advancing Scrum Master Servant Leadership with SAFe®	 with SAFe® 5 Advanced Scrum Master Certification
SAFe® Release Train Engineer	Facilitating Lean-Agile Program Execution	 with SAFe® 5 Release Train Engineer Certification
SAFe® for Architects	Architecting for Continuous Value Flow with SAFe®	 with SAFe® 5 Architect Certification
SAFe® DevOps	Optimizing Your Value Stream	 with SAFe® 5 DevOps Practitioner Certification
SAFe® for Teams	Establishing Team Agility for Agile Release Trains	 with SAFe® 5 Practitioner Certification
Agile Software Engineering	Enabling Technical Agility for the Lean Enterprise	 with SAFe® 5 Agile Software Engineer Certification

Table of Contents

Privacy Notice	1
Digital Workbook User Guide.....	2
Lesson 1: Introducing DevOps.....	5
1.1 Recognize the problem to be solved.....	6
1.2 Explain DevOps and its benefits	12
1.3 Describe continuous security and testing	21
1.4 Describe core DevOps principles.....	23
Lesson 2: Mapping Your Value Stream	33
2.1 Explain the purpose of mapping the Value Stream	34
2.2 Evaluate the efficiency of your current Value Stream	38
Lesson 3: Gaining Alignment with Continuous Exploration.....	49
3.1 Create Solution hypotheses.....	54
3.2 Collaborate and research Customer needs	59
3.3 Architect the Solution for continuous delivery	62
3.4 Synthesize the Vision, the Roadmap, and the Program Backlog	66
Lesson 4: Building Quality In with Continuous Integration	73
4.1 Develop the Solution.....	78
4.2 Build continuously	83
4.3 Test end-to-end	87
4.4 Validate on a staging environment.....	91
Lesson 5: Reducing Time-to-Market with Continuous Deployment	97
5.1 Deploy to production	102
5.2 Verify the Solution.....	107

5.3 Monitor for problems	109
5.4 Respond and recover.....	113
Lesson 6: Delivering Business Value with Release on Demand	125
6.1 Release on Demand	130
6.2 Stabilize the Solution.....	132
6.3 Measure the business value	135
6.4 Learn and react.....	137
Lesson 7: Taking Action.....	143
7.1 Share your DevOps action plan	144
Lesson 8: Becoming a SAFe DevOps Practitioner.....	155
8.1 Becoming a Certified SAFe Professional	156
Glossary	161

Privacy Notice

Your name, company, and email address will be shared with Scaled Agile, Inc. for course fulfillment, including testing and certification. Your information will be used in accordance with the Scaled Agile privacy policy available at <https://www.scaledagile.com/privacy-policy/>.

Digital Workbook User Guide

Frequently Asked Questions

Q: How can I take notes in my digital workbook?

A: After each lesson, there is a notes page that allows you to type notes directly into the workbook. Remember to save your workbook to your personal computer to save any content you may have added.

If you open the digital workbook with a product like Adobe Acrobat, there are functions that allow you to add your own text boxes, add bookmarks, highlight text, and add comments. Remember to save your workbook to your personal computer to save any content you may have added.

Q: What other features are included in the digital workbook?

A: All videos have a hyperlink directly below the slide that will take you to the correct URL. If you click on assets in the front matter, you will be taken to ScaledAgileFramework.com resources like the Implementation Roadmap and course certification pages.

Q: Is my digital workbook saved on the community platform?

A: The original digital workbook file will always be available to you in your Learning Plan on the SAFe Community Platform. However, any text or content added to your digital workbook must be saved on your personal computer. Remember to save your workbook to your personal computer to save any content you may have added.

Q: Can I share my digital workbook with my coworkers?

A: No. You cannot share your digital workbook. It is for personal use only, so you may not reproduce or distribute it.

Q: Can I print the digital workbook?

A: Yes. You may print the digital workbook for your personal use. The file is in full color, so if you'd prefer to print the workbook in black and white only, make sure to adjust your printing preferences.

Logistics

- ▶ Course meeting times
- ▶ Breaks
- ▶ Eliminate distractions
- ▶ Ask questions
- ▶ Be patient
- ▶ Working agreements

Notes:

Course goals

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

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

Notes:

Course map

- ▶ Lesson 1: Introducing DevOps
- ▶ Lesson 2: Mapping Your Value Stream
- ▶ Lesson 3: Gaining Alignment with Continuous Exploration
- ▶ Lesson 4: Building Quality In 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: Becoming a SAFe DevOps Practitioner

Notes:

Lesson 1

Introducing DevOps

Learning Objectives:

- 1.1 Recognize the problem to be solved
- 1.2 Explain DevOps and its benefits
- 1.3 Describe continuous security and testing
- 1.4 Describe core DevOps principles



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

1.1 Recognize the problem to be solved

1.1 Recognize the problem to be solved

SCALED AGILE® © Scaled Agile, Inc.

4

Notes:



Digital disruption is
affecting every
industry across the
globe.

SCALED AGILE® © Scaled Agile, Inc.

5

Notes:



*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.

6

Notes:

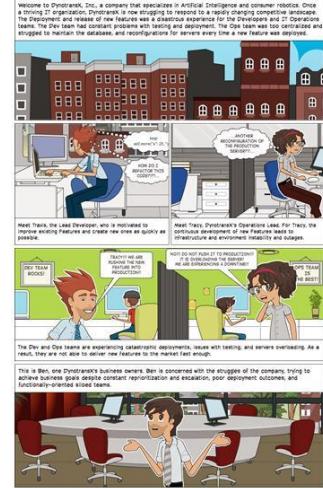


Activity: The Beginning

Prepare
10 min

Share
5 min

- ▶ **Step 1:** Read the story, "The Beginning" in your workbook
- ▶ **Step 2:** Summarize the challenges of the current situation at DynotransX
- ▶ **Step 3:** Discuss as a team:
 - What are some of the challenges you face at your company?
 - Do you agree with any of the solutions offered by Travis and Tracy?



SCALED AGILE® © Scaled Agile, Inc.

7

Notes:

Write down the challenges you face at your company:

Do you agree with any of the solutions offered by Travis and Tracy?

Ask the business to invest in infrastructure

Create a dedicated on-call team

Find a replacement for Travis

Establish fast recovery



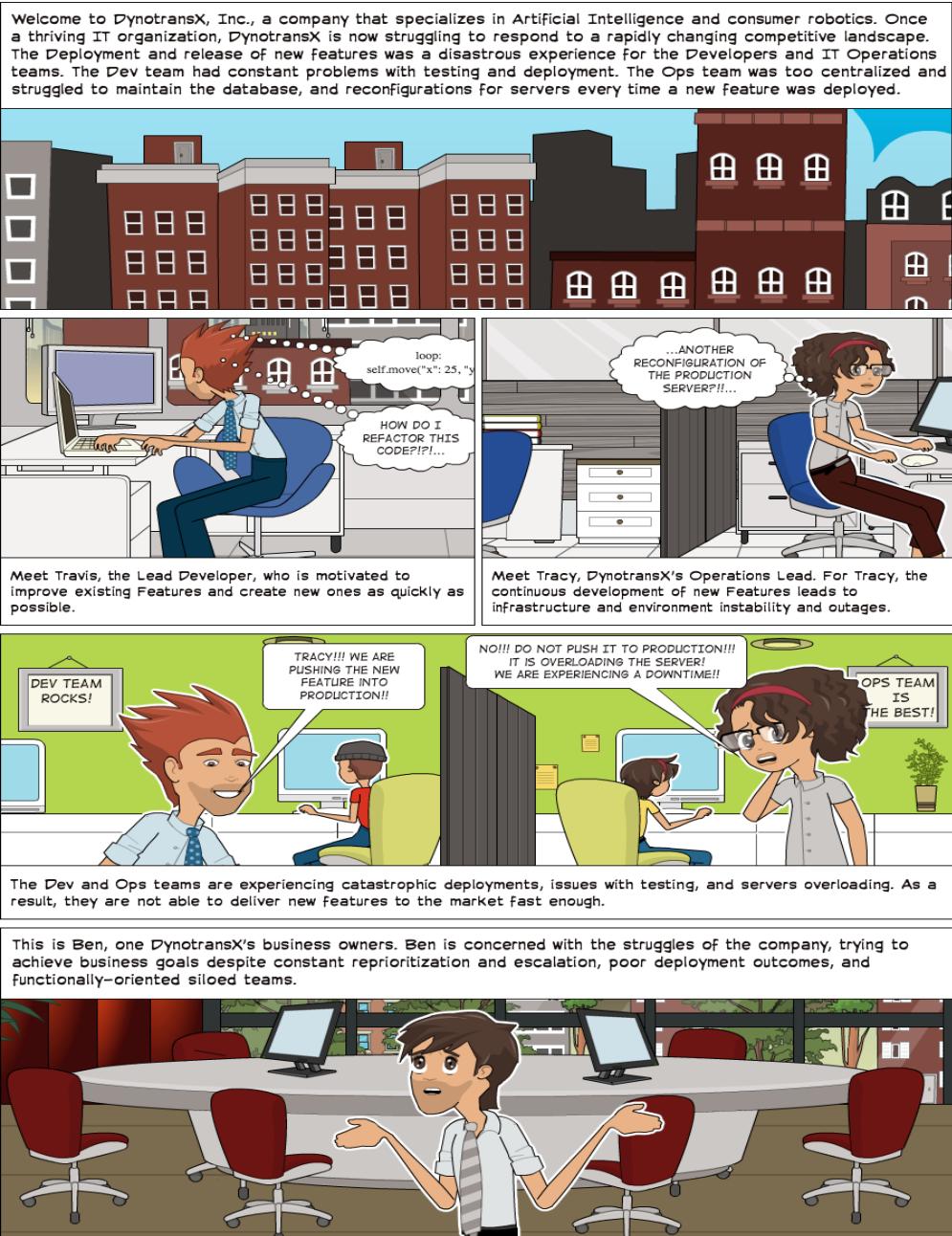
Hire more developers and testers

Prioritize automation

Accelerate work from Dev to Ops

Find a replacement for Tracy

1.1 Recognize the problem to be solved



The problem to be solved

- ▶ Every Enterprise must be able to quickly validate the wants and needs of Customers
- ▶ Every Enterprise must be able to release Features when Customers need them the most

Businesses with traditional software delivery processes struggle to compete effectively in the digital era.



Notes:

1.2 Explain DevOps and its benefits

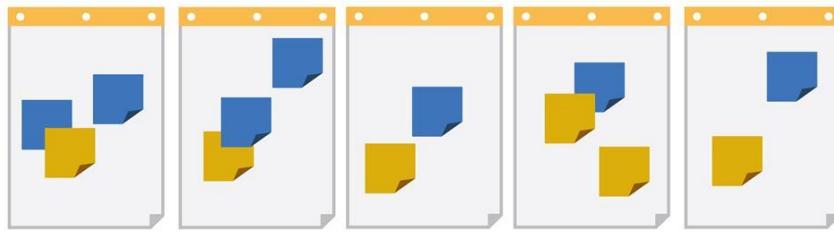
Notes:



Activity: What does DevOps mean to you?

Duration
5 min

- ▶ **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



SCALED AGILE® © Scaled Agile, Inc.

10

Notes:

1.2 Explain DevOps and its benefits

Video: What is DevOps?

Duration
2 min



What is DevOps?
with Morgan Campbell

SCALED AGILE®
Provider of SAFe

<https://vimeo.com/342037390/3a25026214>

SCALED AGILE® © Scaled Agile, Inc.

11

Notes:



Video link: <https://vimeo.com/342037390/3a25026214>

DevOps defined

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

—Amazon Web Services

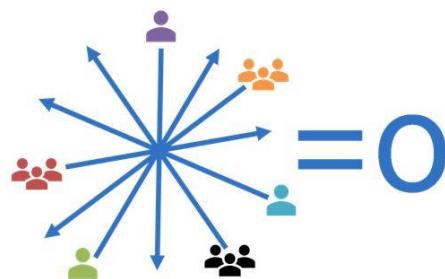
SCALED AGILE® © Scaled Agile, Inc.

12

Notes:

Lack of alignment impedes progress

- ▶ Different groups in the organization have different goals and directions
- ▶ 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.

13

Notes:

1.2 Explain DevOps and its benefits

Who is DevOps?

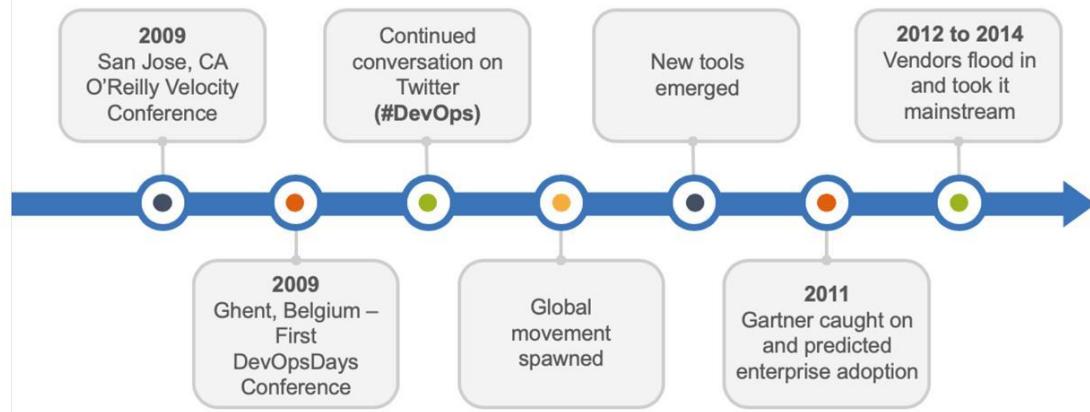


SCALED AGILE® © Scaled Agile, Inc.

14

Notes:

The history of DevOps

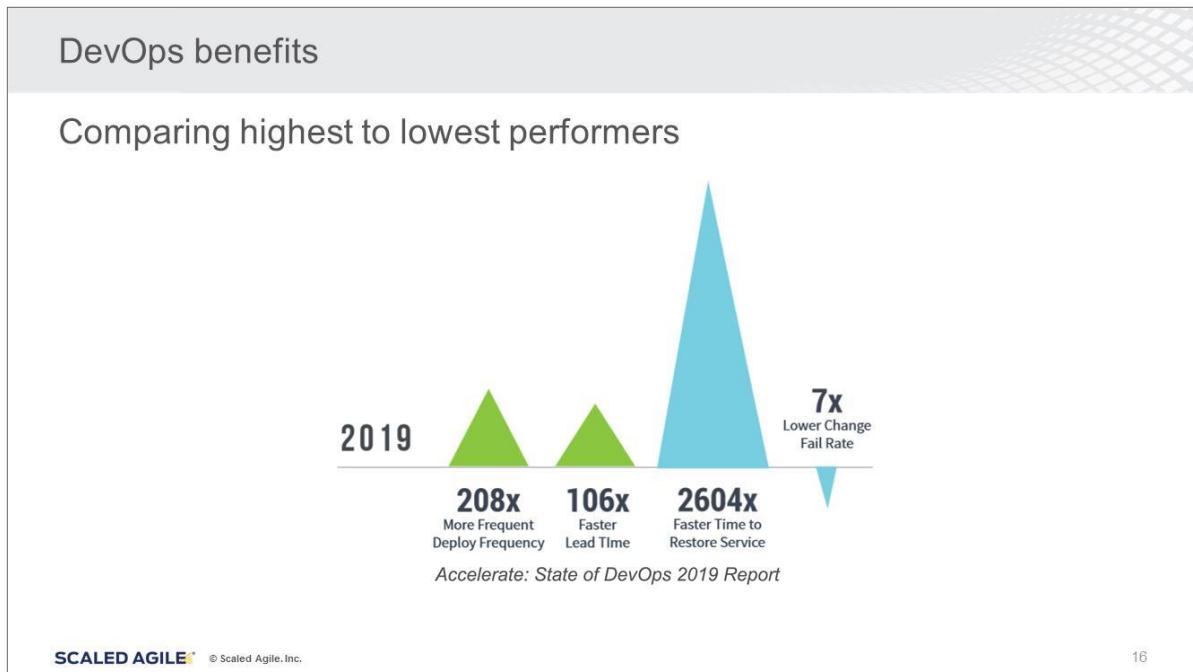


SCALED AGILE® © Scaled Agile, Inc.

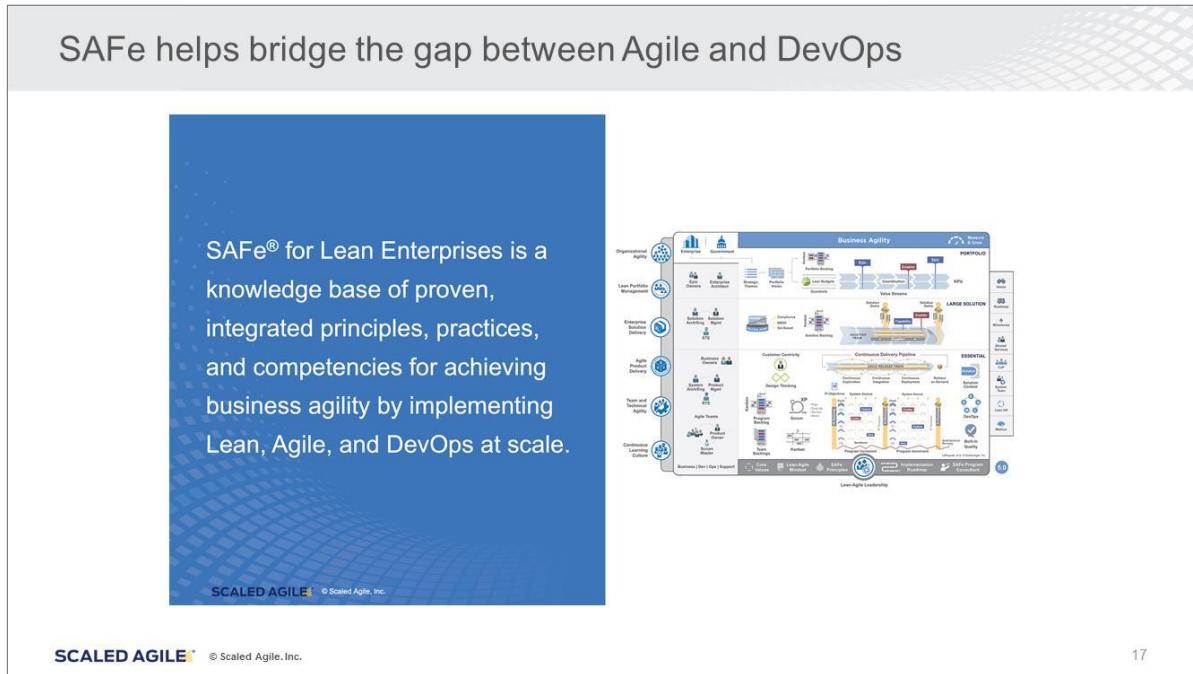
15

Notes:

1.2 Explain DevOps and its benefits



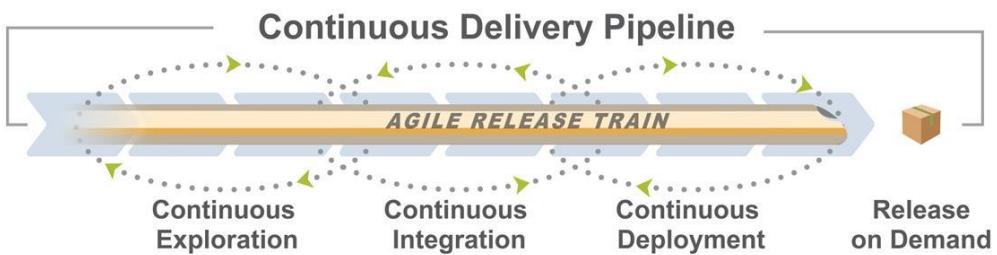
Notes:



Notes:

1.2 Explain DevOps and its benefits

The Continuous Delivery Pipeline enables the flow of value



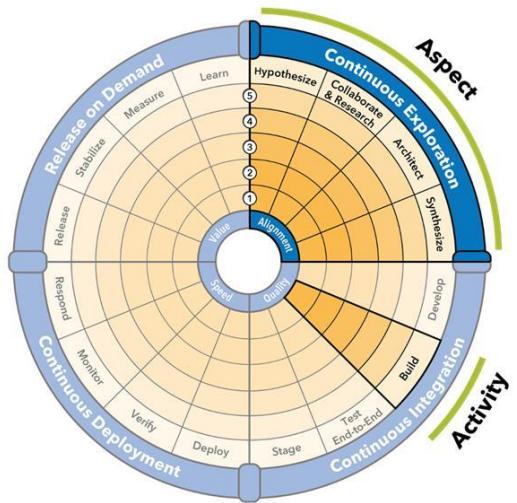
SCALED AGILE® © Scaled Agile, Inc.

18

Notes:

The DevOps Health Radar helps synchronize the pipeline

The DevOps Health Radar is the Continuous Delivery Pipeline divided into four quadrants called **aspects**. They embody alignment, quality, speed, and value. Each aspect contains four **activities** which can be used to assess an organization's maturity.



SCALED AGILE® © Scaled Agile, Inc.

19

Notes:

'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.

Notes:



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

Notes:

1.2 Explain DevOps and its benefits



Video: DevOps in a legacy world

Duration
1 min

- ▶ As enterprises grow over time, they inevitably add new applications and services and adopt new technology stacks. Yet, these enterprises still have to deal with legacy applications and systems.
- ▶ DevOps in the legacy world is not a mission impossible.



<https://youtu.be/8s46nFCzbck>

SCALED AGILE® © Scaled Agile, Inc.

22

Notes:



Video link: <https://youtu.be/8s46nFCzbck>

1.3 Describe continuous security and testing

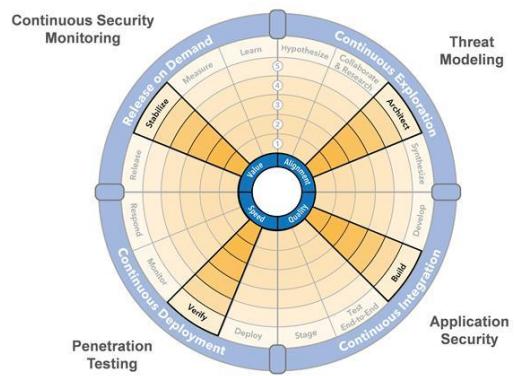
SCALED AGILE® © Scaled Agile, Inc.

23

Notes:

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 aspect of the continuous delivery cycle
- ▶ Information security should be part of every DevOps transformation



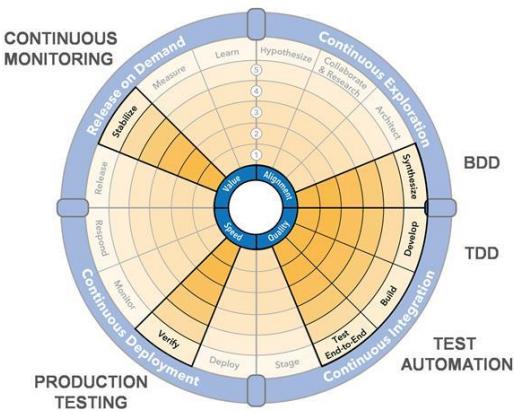
SCALED AGILE® © Scaled Agile, Inc.

24

Notes:

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.

25

Notes:

1.4 Describe core DevOps principles

1.4 Describe core DevOps principles

SCALED AGILE® © Scaled Agile, Inc.

26

Notes:

A CALMR approach to DevOps

- ▶ **Culture** - Establish a culture of shared responsibility for development, deployment, and operations.
- ▶ **Automation** - Automate the Continuous Delivery Pipeline.
- ▶ **Lean flow** - Keep batch sizes small, limit WIP, and provide extreme visibility.
- ▶ **Measurement** - Measure the flow through the pipeline. Implement full-stack telemetry.
- ▶ **Recovery** - Architect and enable low-risk releases. Establish fast recovery, fast reversion, and fast fix-forward.



SCALED AGILE® © Scaled Agile, Inc.

27

Notes:

1.4 Describe core DevOps principles

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



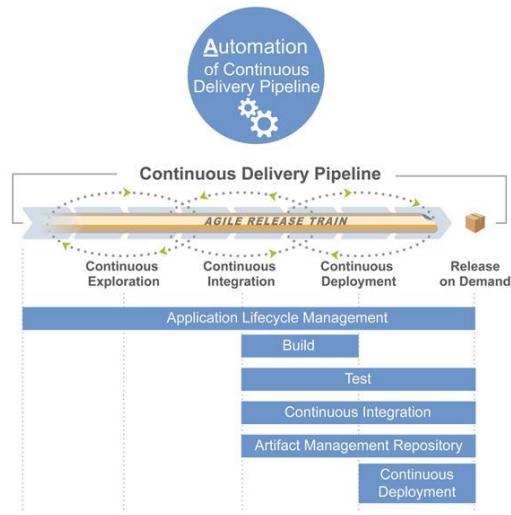
SCALED AGILE® © Scaled Agile, Inc.

28

Notes:

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.

29

Notes:

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

The diagram illustrates the core DevOps principles centered around Lean Flow. It features a central circle labeled "SAFe DevOps" with a gear icon. Surrounding it are four interconnected circles: "Culture of shared responsibility" (with a people icon), "Automation of Continuous Delivery Pipeline" (with a gear icon), "Measurement of everything" (with a gauge icon), and "Recovery enables low risk releases" (with a circular arrow icon). A large blue circle on the right is labeled "Lean Flow accelerates delivery" with a circular arrow icon. Arrows indicate a clockwise flow between the circles, starting from the bottom right and moving counter-clockwise.

SCALED AGILE® © Scaled Agile, Inc.

30

Notes:

1.4 Describe core DevOps principles

Video: One piece flow versus batch production

Duration: 1 min

Order = 10
Batch = 10

Process 1

Process 2

Process 3

FIN GOODS

Order = 10
Batch = 5

Process 1

Process 2

Process 3

FIN GOODS

Order = 10
Batch = 1

Process 1

Process 2

Process 3

FIN GOODS

0:19

0:19

0:19

<https://youtu.be/JoLHKSE8sfU>

SCALED AGILE® © Scaled Agile, Inc.

31

Notes:

Video link: <https://youtu.be/JoLHKSE8sfU>

💬

Discussion: Flow

Prepare
3 min

Share
2 min

► **Step 1:** With your team, considering the flow of work represented on the diagram, discuss:

- How do you think the PO for Team 3 feels? How do you think the PO for Team 1 feels?
- What is a large batch in your context and how can you improve flow?

► **Step 2:** Be prepared to share with the class

PO Team 1

Order = 10
Batch = 10

Process 1

Process 2

Process 3

0.07

PO Team 2

Order = 10
Batch = 5

Process 1

Process 2

Process 3

0.07

PO Team 3

Order = 10
Batch = 1

Process 1

Process 2

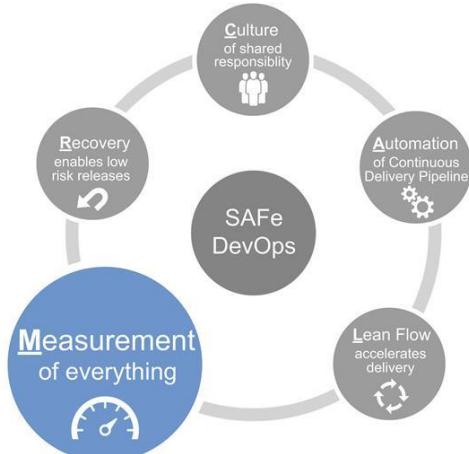
Process 3

0.07

Notes:

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



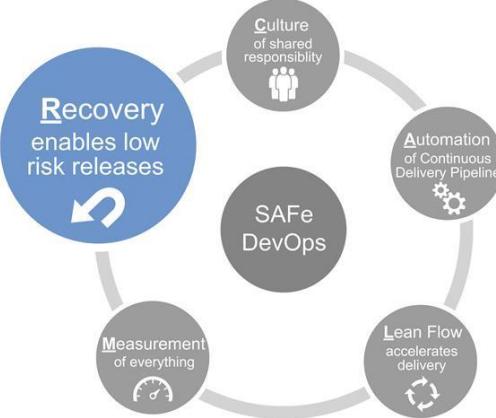
SCALED AGILE® © Scaled Agile, Inc.

33

Notes:

Architect for releasability and recovery

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



SCALED AGILE® © Scaled Agile, Inc.

34

Notes:



Activity: CALMR

Duration
5 min

- ▶ **Step 1:** Revisit the CALMR board

- ▶ **Step 2:** Discuss and summarize:

- Which CALMR concepts are strongly represented?
- Which ones are under represented? Why?



SCALED AGILE® © Scaled Agile, Inc.

35

Notes:

Lesson review

In this lesson you:

- ▶ Recognized the problem to be solved
- ▶ Explained DevOps and its benefits
- ▶ Described continuous security and testing
- ▶ Described core DevOps principles

Notes:

Lesson 1 notes



Click below to type your thoughts.

1.4 Describe core DevOps principles

Lesson 2

Mapping Your Value Stream

Learning Objectives:

- 2.1 Explain the purpose of mapping the Value Stream
- 2.2 Evaluate the efficiency of your current Value Stream



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

2.1 Explain the purpose of mapping the Value Stream

2.1 Explain the purpose of mapping the Value Stream

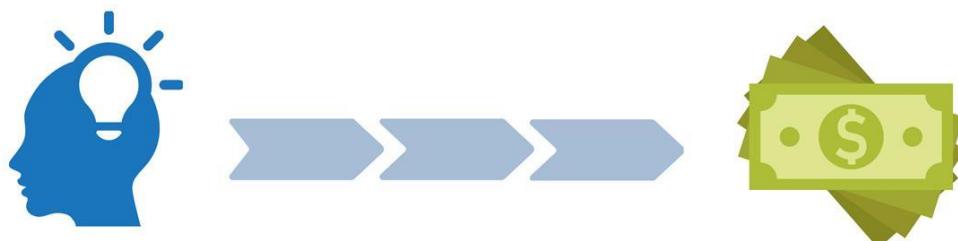
SCALED AGILE® © Scaled Agile, Inc.

37

Notes:

Why map the Value Stream?

- ▶ Understand how work flows through the organization from concept to cash
- ▶ Measure process quality and organizational efficiency
- ▶ Identify bottlenecks to the flow of value
- ▶ Understand how we can improve the flow of value

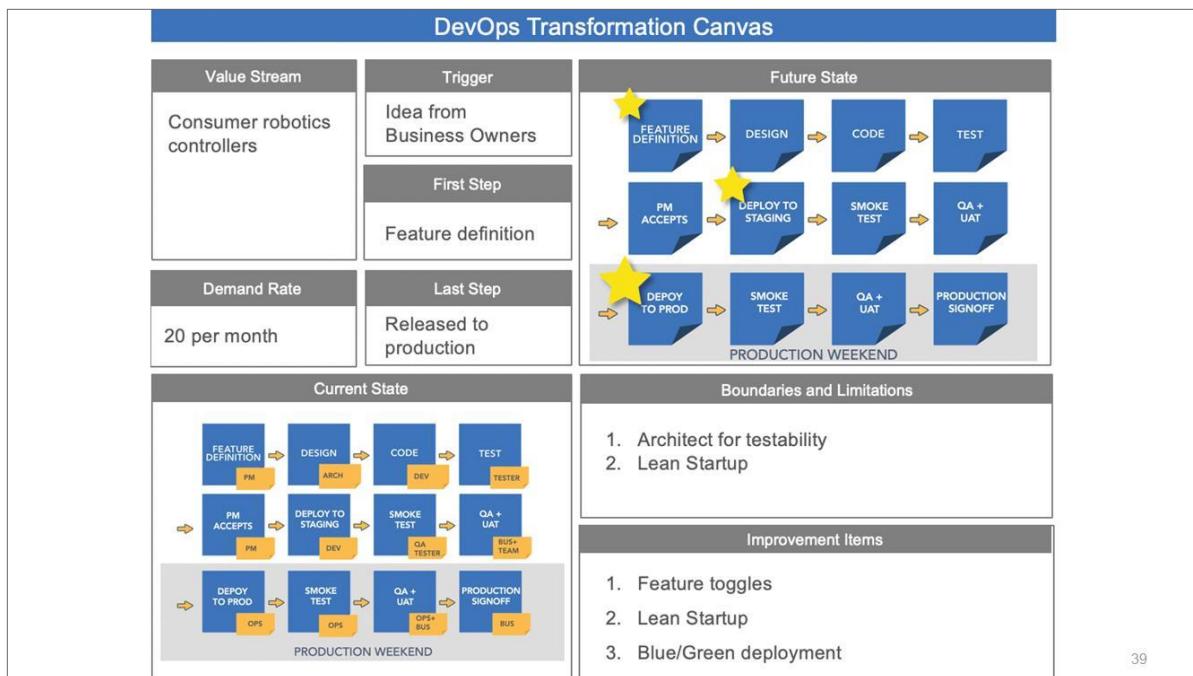


SCALED AGILE® © Scaled Agile, Inc.

38

Notes:

2.1 Explain the purpose of mapping the Value Stream



Notes:

2.1 Explain the purpose of mapping the Value Stream

Video: Value Stream Canvas

Duration
3 min

The DevOps Transformation Canvas is a visual tool for comparing current and future software development processes. It consists of several sections:

- Value Stream:** Consumer robotics controllers.
- Idea from Business Owners:** First Step.
- Demand Rate:** 20 per month.
- Last Step:** Released to production.
- Future State:** A flowchart showing a linear process: Feature Definition → DESIGN → CODE → TEST. Arrows indicate sequential steps. Yellow stars highlight "FEATURE DEFINITION" and "DEPLOY TO STAGING".
- Current State:** A flowchart showing a more complex process with multiple parallel paths and handoffs between teams (PM, DEV, QA, OPS). It also includes a "PRODUCTION WEEKEND" section.
- Boundaries and Limitations:** 1. Architect for testability, 2. Lean Startup.
- Improvement Items:** 1. Feature toggles, 2. Lean Startup, 3. Blue/Green deployment.

<https://vimeo.com/325445925/59913369d4>

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Video link: <https://vimeo.com/325445925/59913369d4>

2.1 Explain the purpose of mapping the Value Stream



Activity: Identify the context for Value Stream mapping

Duration
10 min

- ▶ **Step 1:** On a flip chart or a whiteboard, draw the DevOps Transformation Canvas.
- ▶ **Step 2:** Start filling out the canvas by identifying which Value Stream to improve.
- ▶ **Step 3:** Identify the the following components:
 - Trigger
 - First Step
 - Last Step
- ▶ **Step 4:** What is the demand rate?

DevOps Transformation Canvas		
Value Stream	Trigger	Future State
Consumer robotics controllers	Idea from Business Owners	
	First Step Feature definition	
Demand Rate	Last Step	Boundaries and Limitations
20 per month	Released to production	Improvement Items
Current State		

SCALED AGILE® © Scaled Agile, Inc.

Notes:

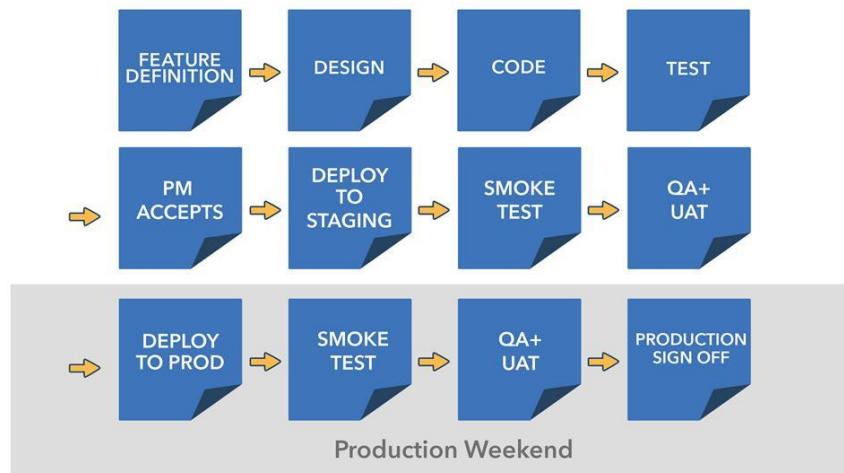
2.2 Evaluate the efficiency of your current Value Stream

SCALED AGILE® © Scaled Agile, Inc.

42

Notes:

Value Stream mapping – identify the steps: Customer example

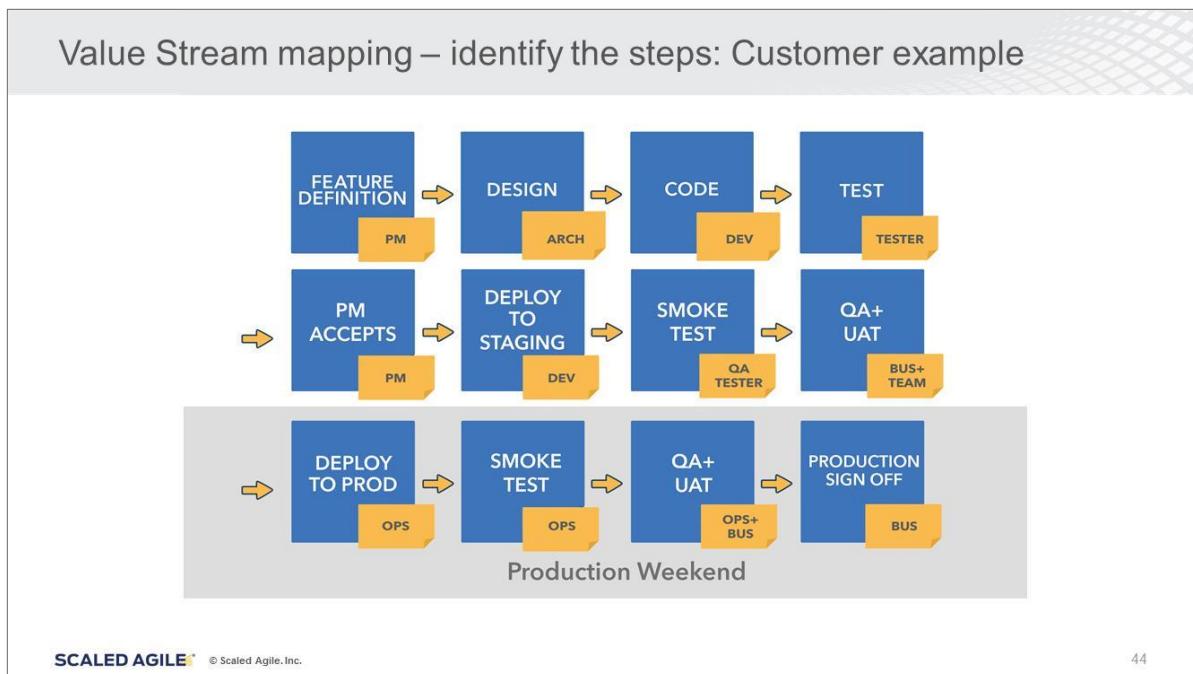


SCALED AGILE® © Scaled Agile, Inc.

43

Notes:

2.2 Evaluate the efficiency of your current Value Stream



44

Notes:



Activity: Visualize the Value Stream



- ▶ **Step 1:** On the current state section of the canvas, model your existing value stream using sticky notes. Start at the first step and finish at the last step identified previously.
- ▶ **Step 2:** Identify the role or team responsible for each step.

SCALED AGILE® © Scaled Agile, Inc.

45

Notes:

2.2 Evaluate the efficiency of your current Value Stream



Activity: Present your team's map

Prepare
3 min

Share
2 min

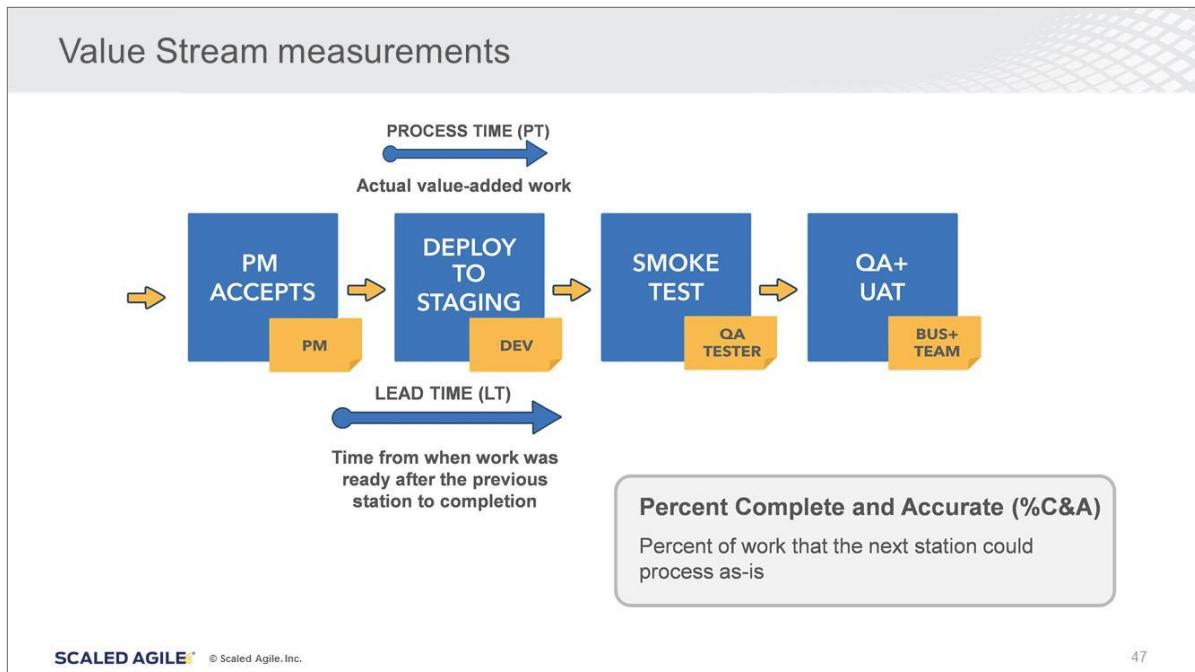
- ▶ **Step 1:** Walk around the room and review other people's maps
 - **Note:** Make sure that at any point there is at least one person to represent your team's map
- ▶ **Step 2:** Identify items from other maps that you might have missed at your own map
- ▶ **Step 3:** As a team, adjust your current state map

SCALED AGILE® © Scaled Agile, Inc.

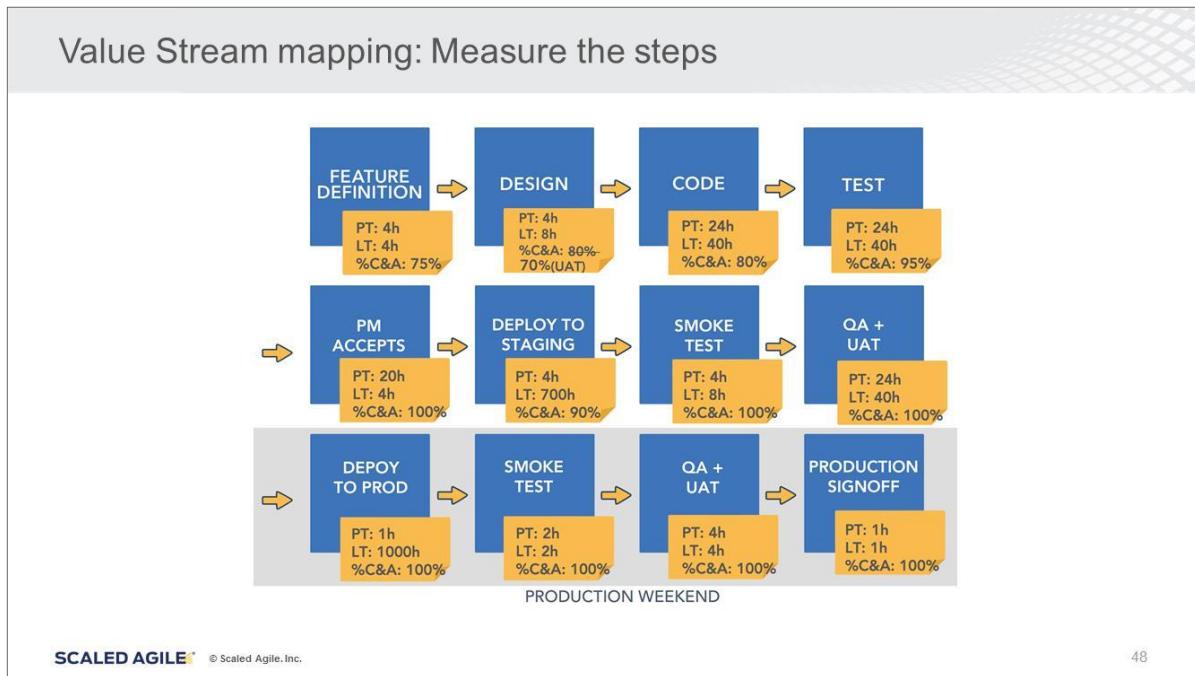
46

Notes:

2.2 Evaluate the efficiency of your current Value Stream

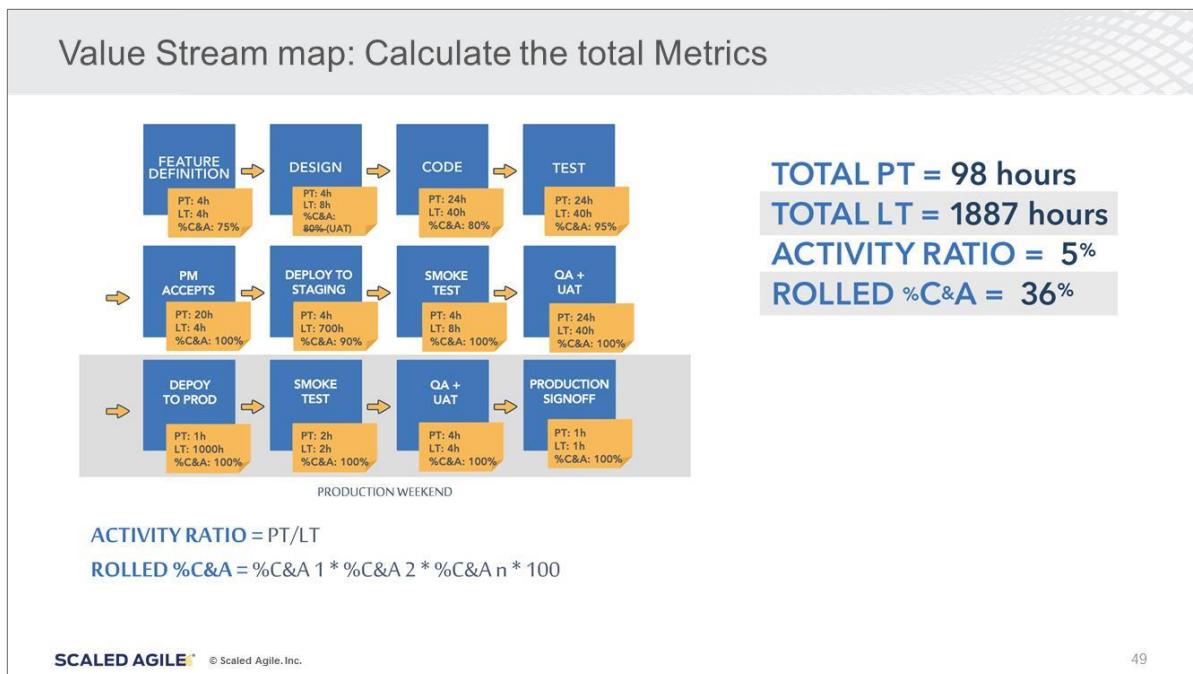


Notes:



Notes:

2.2 Evaluate the efficiency of your current Value Stream



49

Notes:



Activity: Measure Value Stream efficiency

Duration
30 min

- ▶ **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

Notes:



Discussion: Present your Value stream map

Duration
30 min

- ▶ **Step 1:** As a team, present your board
- ▶ **Step 2:** Address the following questions:
 - What is the total lead time? Total process time?
 - What is the activity ratio? Rolled % complete and accurate?
 - Based on your metrics, where are the biggest bottlenecks?

Notes:

Lesson review

In this lesson you:

- ▶ Evaluated the efficiency of your current Value Stream
- ▶ Explained the purpose of mapping the Value Stream

Notes:

Lesson 2 notes



Click below to type your thoughts.

2.2 Evaluate the efficiency of your current Value Stream

Lesson 3

Gaining Alignment with Continuous Exploration

Learning Objectives:

- 3.1 Create Solution hypotheses
- 3.2 Collaborate and research Customer needs
- 3.3 Architect the Solution for continuous delivery
- 3.4 Synthesize the Vision, the Roadmap, and the Program Backlog



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



Activity: A Brilliant Idea

Prepare
5 min
Share
5 min

- ▶ **Step 1:** Read the story, "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



SCALED AGILE® © Scaled Agile, Inc.

53

Notes:

From the perspective of the role you play, defend or challenge the statements of your character:

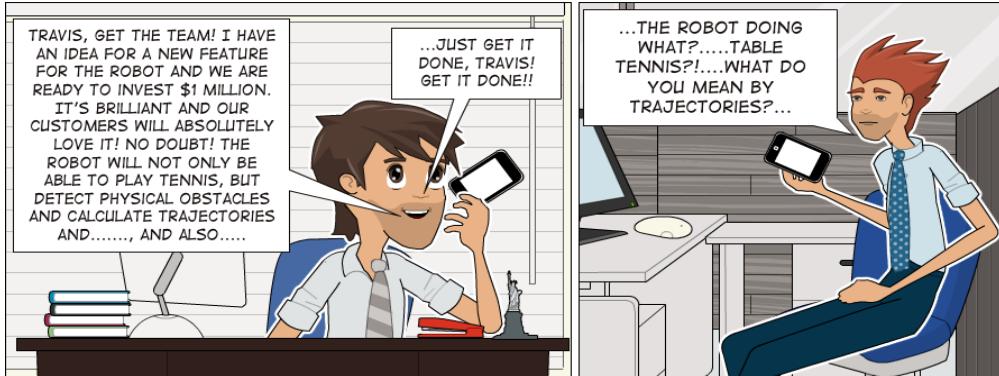
When there is a great idea, it should be prioritized immediately

Only Business Owners decide what work needs to be done



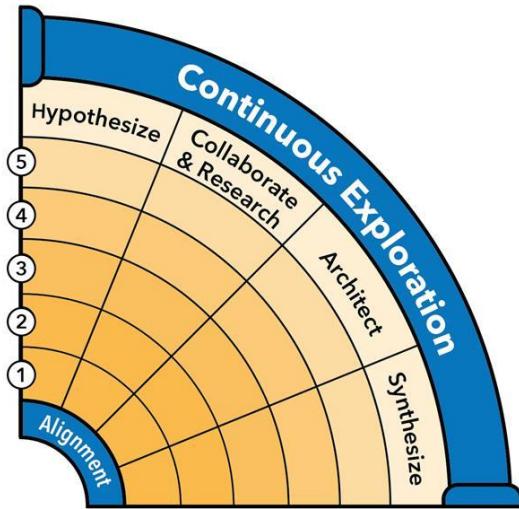
Working on a new feature must start immediately, regardless if it is in the Backlog

There has to be an extensive up-front definition of the feature prior to starting work



Continuous Exploration

- ▶ Business objective
 - Alignment
- ▶ IT objective
 - Continuous Exploration (CE)
- ▶ Activities
 - Hypothesize
 - Collaborate & Research
 - Architect
 - Synthesize



SCALED AGILE® © Scaled Agile, Inc.

54

Notes:

3.1 Create Solution hypotheses

3.1 Create Solution hypotheses

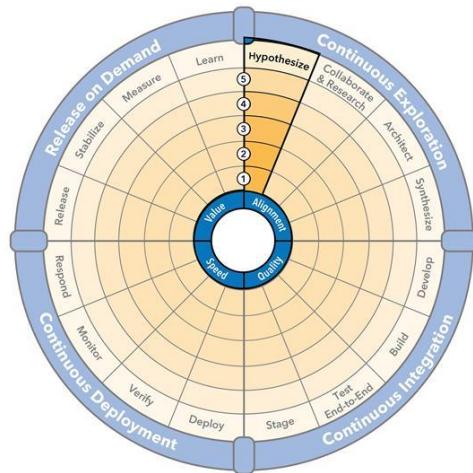
SCALED AGILE® © Scaled Agile, Inc.

55

Notes:

Hypothesize

- ▶ Purpose: Define the hypothesis to be validated through the Continuous Delivery Pipeline
- ▶ Skills:
 - Lean startup
 - Innovation accounting



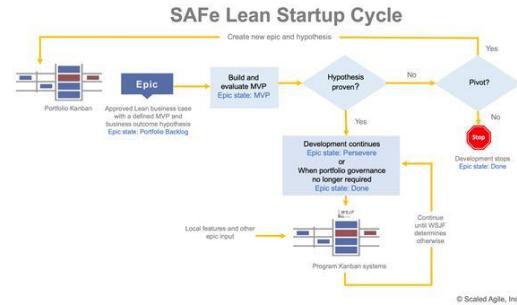
SCALED AGILE® © Scaled Agile, Inc.

56

Notes:

Lean startup

- ▶ The lean startup cycle focuses on identifying the viability of ideas
- ▶ It follows the plan-do-check-adjust cycle (PDCA)
- ▶ MVP - establish a baseline to test assumptions and gather objective data
- ▶ Evaluate the hypothesis:
 - If the benefit hypothesis has been proven true, the Value Streams will implement more Features
 - If the hypothesis is proven false, then a decision is made to either pivot with a new hypothesis or to stop work on the Epic

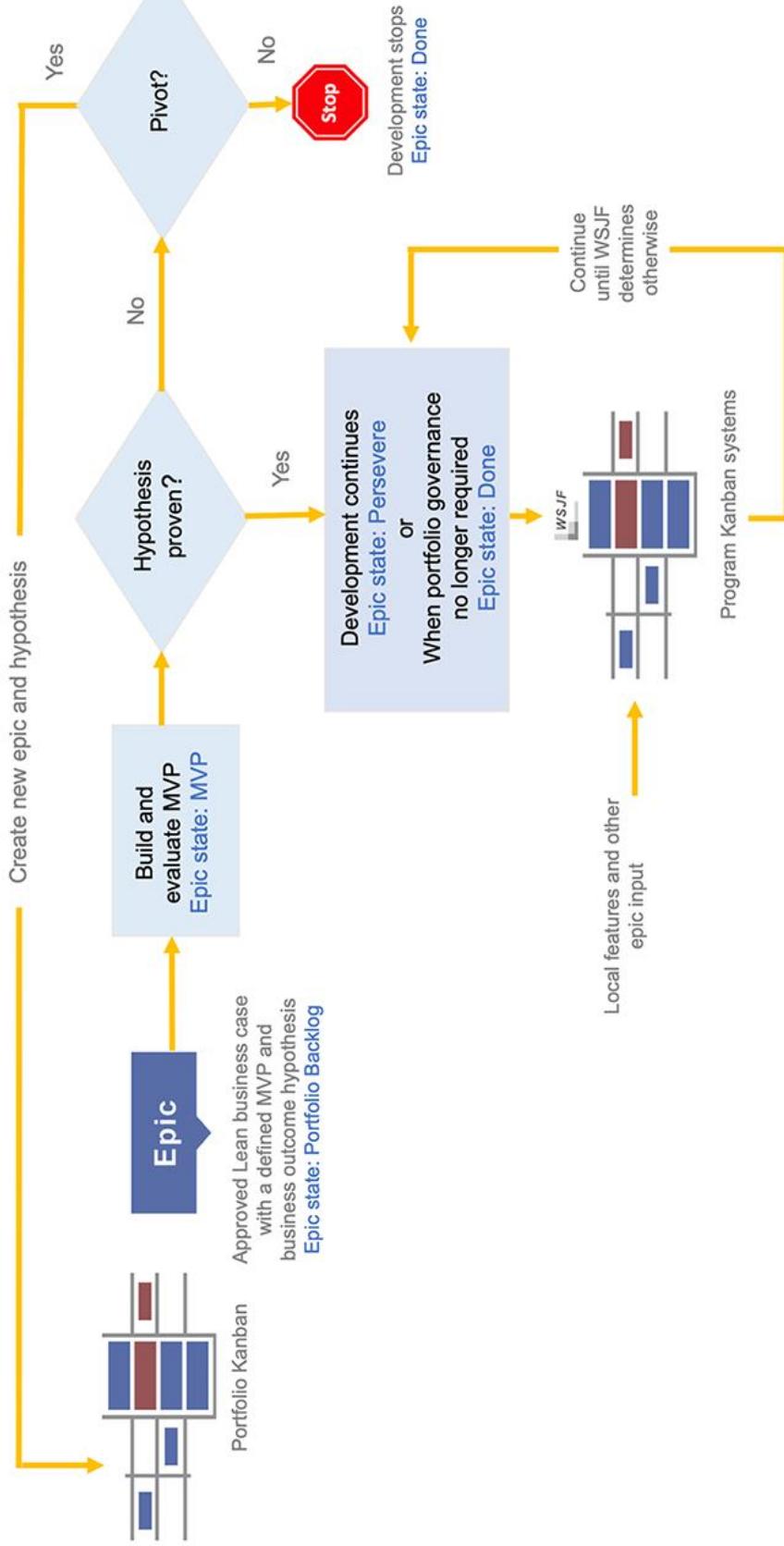


SCALED AGILE® © Scaled Agile, Inc.

57

Notes:

SAFe Lean Startup Cycle



3.1 Create Solution hypotheses

Video: Innovation Accounting

Duration
2 min



Innovation Accounting With Marc Rix

SCALED AGILE®
Provider of SAFe®

<https://vimeo.com/332677950/751f8b51dd>

SCALED AGILE® © Scaled Agile, Inc.

58

Notes:



Video link: <https://vimeo.com/332677950/751f8b51dd>

3.1 Create Solution hypotheses

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.

59

Notes:

Self-assessment: Hypothesize activity

Duration: 1 min

1. Sit	Ideas are vague or not defined.
2. Crawl	Ideas are defined (e.g., as Epics) but do not include hypothesis statements.
3. Walk	Some ideas are expressed as hypothesis statements with measurable outcomes.
4. Run	Most ideas are expressed as hypothesis statements with measurable outcomes and include MVPs.
5. Fly	All ideas are expressed as hypothesis statements with measurable outcomes and include MVPs.

The DevOps Radar is a circular diagram divided into three main quadrants: Continuous Experimentation (top), Continuous Deployment (bottom), and Continuous Integration (right). The center is labeled 'Build & Deliver'. The outer ring contains various DevOps practices: 'Relate on Demand', 'Refine', 'Reflow', 'Continuous Experimentation' (with sub-labels 'Collective Research & Research', 'Acquire', 'Synthesis'), 'Continuous Deployment' (with sub-labels 'Verify', 'Deploy', 'Stage', 'End-to-End', 'Build'), and 'Continuous Integration' (with sub-labels 'Test', 'End-to-End', 'Build'). A yellow dot is placed on the 'hypothesize' point in the Continuous Experimentation quadrant.

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

60

Notes:

3.2 Collaborate and research Customer needs

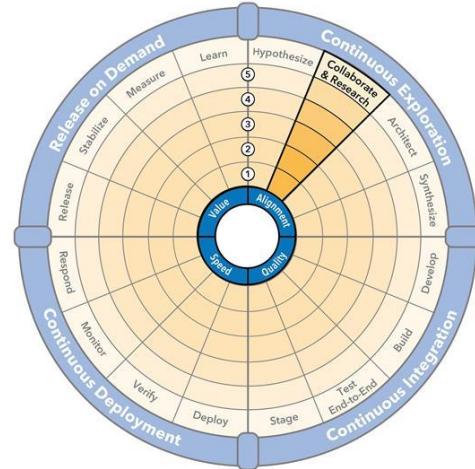
SCALED AGILE® © Scaled Agile, Inc.

61

Notes:

Collaborate & Research

- ▶ Purpose: Work with multiple stakeholders to understand Customer needs
- ▶ Skills:
 - Lean UX
 - Research



SCALED AGILE® © Scaled Agile, Inc.

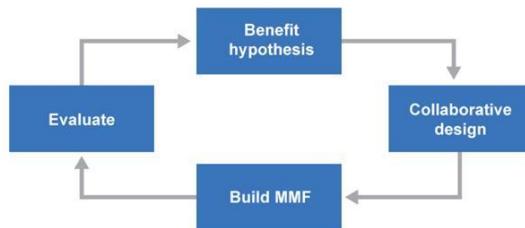
62

Notes:

3.2 Collaborate and research Customer needs

Lean UX

- ▶ Lean User Experience (Lean UX) is a mindset, culture, and 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.

63

Notes:

Research

- ▶ **Design Thinking** - Explore the problem and solution spaces using personas, journey maps, empathy maps, and other Agile Product Management techniques.
- ▶ **Customer visits and 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 analyses, 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.

64

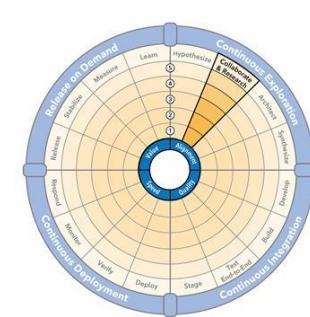
Notes:

3.2 Collaborate and research Customer needs

 Self-assessment: Research activity

Duration: 1 min

1. Sit	Product Management roles and responsibilities are not defined or followed.
2. Crawl	Product Management creates requirements in large batches with little Customer or development collaboration.
3. Walk	Product Management collaborates with business or development experts, but not both, when defining requirements.
4. Run	Product Management regularly collaborates with business, development, and operation experts but does not define minimum marketable features.
5. Fly	Product Management always collaborates with business-side, development-side, and operation-side experts and defines minimum marketable features.



Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

65

Notes:

3.3 Architect the Solution for continuous delivery

3.3 Architect the Solution for continuous delivery

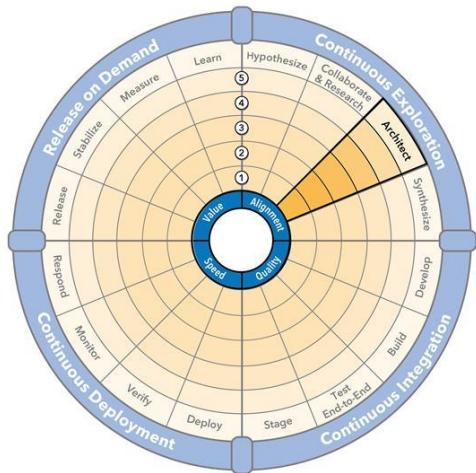
SCALED AGILE® © Scaled Agile, Inc.

66

Notes:

Architect

- ▶ Purpose: Architect for continuous delivery and DevOps
- ▶ Skills:
 - Architect for testability
 - Separate deploy and release
 - Decouple release elements
 - Architect for operations
 - Threat modeling



SCALED AGILE® © Scaled Agile, Inc.

67

Notes:

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
- ▶ Patterns that can accelerate value flow:
 - Domain Driven Design (DDD)
 - Loose coupling and API-driven architecture
 - Cloud-native architecture
 - Microservices and containerization
 - Serverless architecture
 - Strangler pattern



System
Arch/Eng

SCALED AGILE® © Scaled Agile, Inc.

68

Notes:

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



SCALED AGILE® © Scaled Agile, Inc.

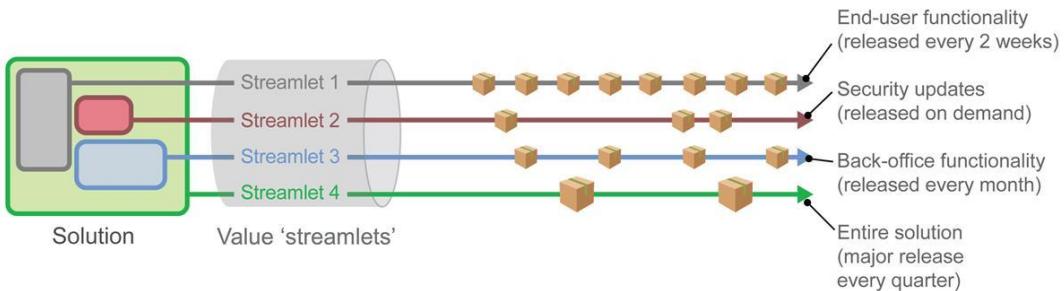
69

Notes:

3.3 Architect the Solution for continuous delivery

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.

70

Notes:

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.

71

Notes:

Threat modeling

- ▶ Information security considerations should start early
- ▶ Identify potential security threats and attack vectors
- ▶ Architect to address security concerns
- ▶ Ensure backlogs reflect important security requirements



SCALED AGILE® © Scaled Agile, Inc.

72

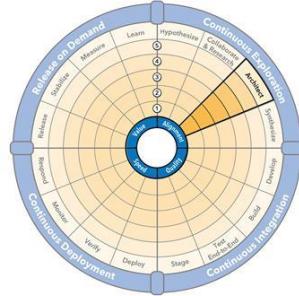
Notes:



Self-assessment: Architect activity

1. Sit	Architecture is monolithic and fragile; it is difficult to change and involves managing complex dependencies across many components and systems.
2. Crawl	Architecture is predominantly monolithic but some applications/systems are loosely coupled.
3. Walk	Architecture is mostly decoupled but doesn't allow Release on Demand.
4. Run	Architecture is aligned around value delivery with few dependencies across components and systems.
5. Fly	Architecture is built for Release on Demand and operability.

Duration

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

73

Notes:

65 | © Scaled Agile, Inc.

For personal use only - Viktorija Marimiseva - Sep 14, 2020 (ID:0050W000007ELwNQAW)

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

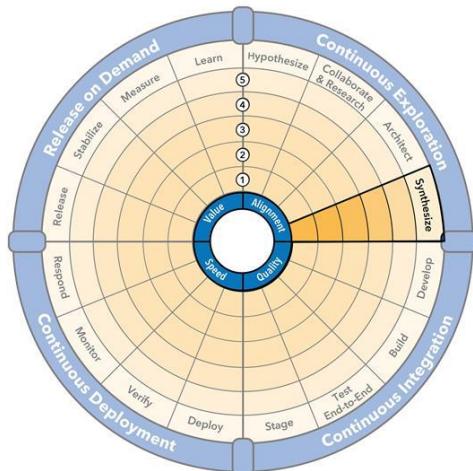
SCALED AGILE® © Scaled Agile, Inc.

74

Notes:

Synthesize

- ▶ Purpose: Synthesize the hypotheses, research, collaboration, and architecture into a Vision, Roadmap, and backlog. Gain alignment with PI Planning.
- ▶ Skills:
 - Feature writing
 - Behavior-driven development (BDD)
 - Economic prioritization
 - PI Planning



SCALED AGILE® © Scaled Agile, Inc.

75

Notes:

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

SSO example:

Cruise Control & Speed Finder

Benefit hypothesis

Cruise control and speed finder will allow us to sell 50% more cars

Acceptance criteria

Given Vehicle is traveling with cruise control at the posted speed limit

When a new speed limit is detected

Then vehicle automatically adjusts to the new speed limit

Notes:

Behavior-driven development (BDD)

- ▶ Behavior-driven development is a test-first approach to writing requirements
- ▶ Gherkin syntax (Given-When-Then) is commonly used
- ▶ Permits building executable specifications
- ▶ Acceptance criteria become executable tests

Given _____
When _____
Then _____

Test 

Notes:

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

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

SCALED AGILE® © Scaled Agile, Inc.

78

Notes:

PI Planning

Program Increment (PI) Planning is a cadence-based, face-to-face event that serves as the heartbeat of the Agile Release Train (ART), aligning all the teams on the ART to a shared mission and Vision.



PI Planning

SCALED AGILE® © Scaled Agile, Inc.

79

Notes:

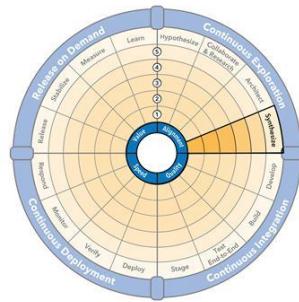
3.4 Synthesize the Vision, the Roadmap, and the Program Backlog



Self-assessment: Synthesize

Duration: 1 min

1. Sit	The Program Backlog does not exist or is not shared.
2. Crawl	The Program Backlog exists, but the Features are incomplete and prioritization is an afterthought.
3. Walk	The Program Backlog contains fully defined Features but are not prioritized using WSJF.
4. Run	Features in the Program Backlog are complete, prioritized using WSJF and matched to the delivery capacity of the ART.
5. Fly	The Program Backlog is a collection of minimum marketable features created using BDD and prioritized using WSJF.



Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

Notes:

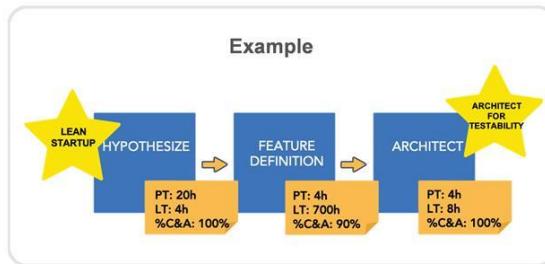


Activity: Apply Continuous Exploration to your Value Stream

Prepare
15 min

Share
5 min

- ▶ **Step 1:** Identify the skills in this lesson that would improve flow in your Value Stream between idea capture and ready to code. Write each on a star-shaped sticky note.
- ▶ **Step 2:** Construct a new view of this part of the Value Stream in the Future State portion of the DevOps Transformation Canvas. (Reuse, reorder, or replace steps as desired.)
- ▶ **Step 3:** Place the stars on the steps they would enable and provide new process time, lead time, and %C&A for each step.



SCALED AGILE® © Scaled Agile, Inc.

81

Notes:

Lesson review

In this lesson you:

- ▶ Explored how to create Solution hypotheses
- ▶ Discussed how to collaborate and research Customer needs
- ▶ Explored how to architect the Solution for continuous delivery
- ▶ Explored how to synthesize the Vision, the Roadmap, and the Program Backlog

Notes:

Lesson 3 notes



Click below to type your thoughts.

Lesson 4

Building Quality In with Continuous Integration

Learning Objectives:

- 4.1 Develop the Solution
- 4.2 Build continuously
- 4.3 Test end-to-end
- 4.4 Validate on a staging environment



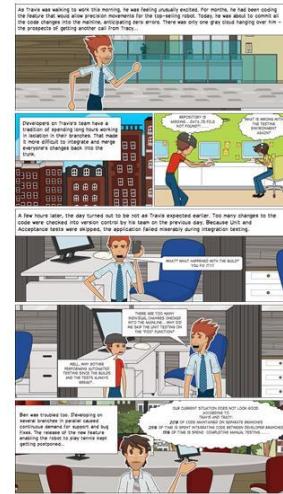
SAFe® Course Attending this course gives students access to the SAFe® DevOps Practitioner exam and related preparation materials.



Activity: Continuous Problems

Prepare 5 min Share 5 min

- ▶ **Step 1:** Read the story, “Continuous 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:** Discuss as a team:
 - Do you share similar CI challenges with other members in your group?



SCALED AGILE® © Scaled Agile, Inc.

83

Notes:

Summarize the biggest challenge with CI in your own context:

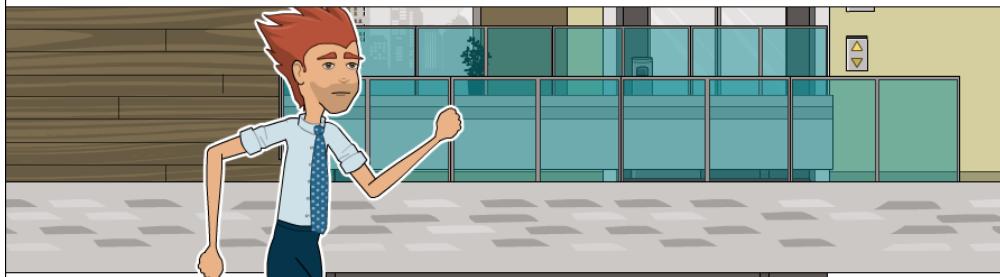
Do you share similar CI challenges with other members in your group?

Are any of the options below, possible solutions towards CI?

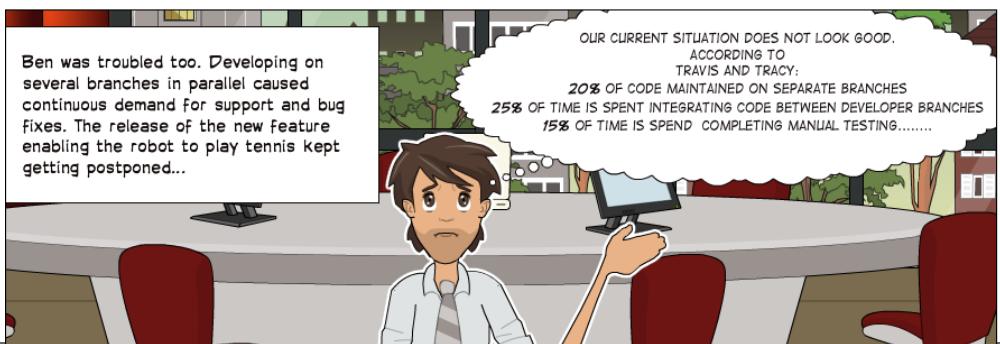


- Frequent code commits, larger batch sizes
- Less frequent code commits, small batches
- Establish gated commits
- Adopt trunk-based development to predict higher throughput

As Travis was walking to work this morning, he was feeling unusually excited. For months, he had been coding the feature that would allow precision movements for the top-selling robot. Today, he was about to commit all the code changes into the mainline, anticipating zero errors. There was only one gray cloud hanging over him – the prospects of getting another call from Tracy...

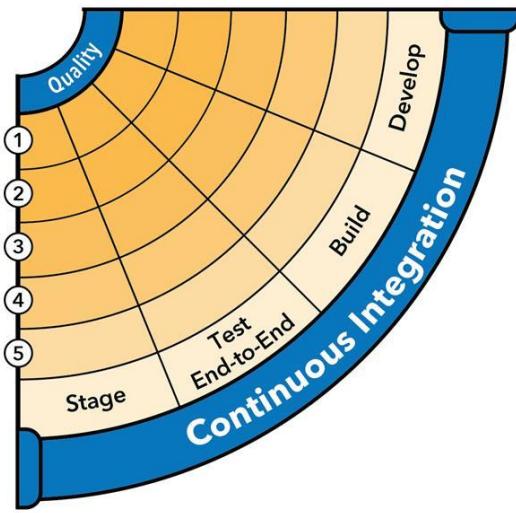


A few hours later, the day turned out to be not as Travis expected earlier. Too many changes to the code were checked into version control by his team on the previous day. Because Unit and Acceptance tests were skipped, the application failed miserably during integration testing.



Continuous Integration

- ▶ Business objective
 - Built-in quality
- ▶ IT objective
 - Continuous Integration (CI)
- ▶ Activities
 - Develop
 - Build
 - Test end-to-end
 - Stage



SCALED AGILE® © Scaled Agile, Inc.

84

Notes:

4.1 Develop the Solution

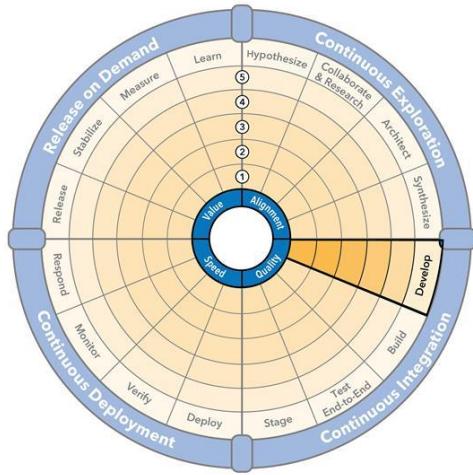
SCALED AGILE® © Scaled Agile, Inc.

85

Notes:

Develop

- ▶ Purpose: Implement a Story or a part of a Story and commit the code
- ▶ Skills:
 - Break Features into Stories
 - Test-driven development
 - Version control
 - Engineering practices
 - Pair work
 - Application telemetry
 - Threat modeling (covered in Architect)



SCALED AGILE® © Scaled Agile, Inc.

86

Notes:

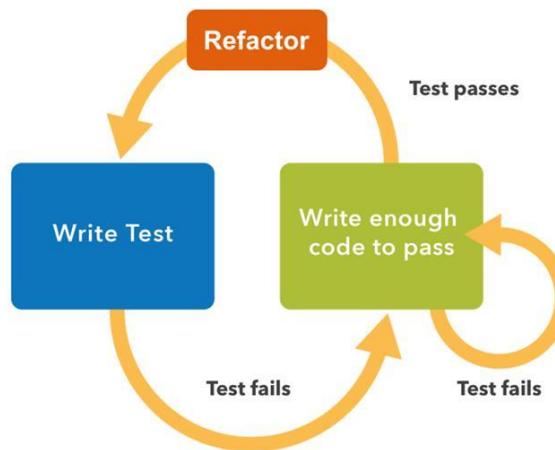
Break Features into Stories within their boundaries

Techniques for splitting Features and Stories, 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

Notes:

Test-driven development (TDD)



Notes:

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.

89

Notes:

Agile Software Engineering

- ▶ Modern engineering practices boost speed and quality. They include:
 - Test-first mindset
 - Agile modeling
 - Emergent design
 - Pattern-based coding
 - Code reviews



Built-In Quality

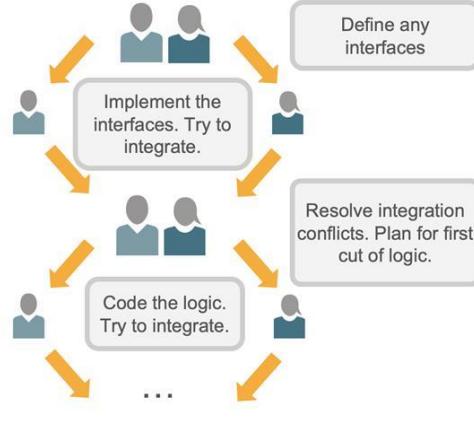
SCALED AGILE® © Scaled Agile, Inc.

90

Notes:

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.
 - Is broader and less constraining than pair programming
 - Typically represents 20% to 80% of any given team member's time
 - Can be spontaneous, scheduled, or performed on rotation



Example user story implementation flow

SCALED AGILE® © Scaled Agile, Inc.

91

Notes:

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.

92

Notes:

4.1 Develop the Solution

Self-assessment: Develop

Duration: 1 min

1. Sit	The Team Backlog does not exist or is not used to manage daily work.
2. Crawl	Stories are either incomplete or too verbose; unit tests are generally not written; peer reviews are not conducted.
3. Walk	Stories are complete; most changes have unit tests; peer reviews are usually conducted.
4. Run	Code is checked in daily; unit test coverage is 80+%; peer reviews are always conducted.
5. Fly	Code is checked in multiple times per day; tests are written before code (TDD); pair work and other Built-in Quality practices are the norm.

The DevOps Radar diagram is a circular chart divided into four quadrants. The top-right quadrant is labeled 'Continuous Exploration' and contains 'Collaborate & Research', 'Adapt', 'Experiment', and 'Hypothesize'. The bottom-right quadrant is labeled 'Continuous Integration' and contains 'Build', 'Test', and 'Deploy'. The bottom-left quadrant is labeled 'Continuous Deployment' and contains 'Release', 'Revert', 'Automate', and 'Verify'. The top-left quadrant is labeled 'Release on Demand' and contains 'Refactor', 'Rewrite', 'Simplify', and 'Measure'. The center of the radar is labeled 'All'. The outer ring of the radar has segments for 'Continuous Delivery' and 'Continuous Deployment'.

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

Notes:

4.2 Build continuously

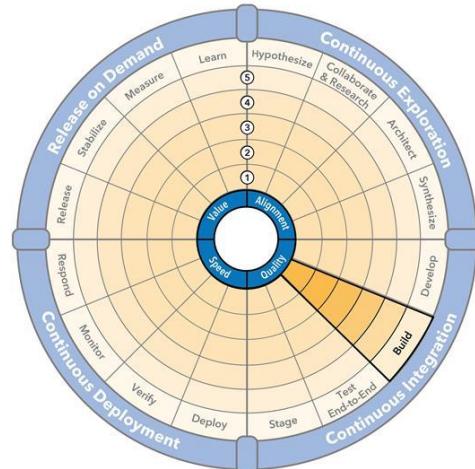
SCALED AGILE® © Scaled Agile, Inc.

94

Notes:

Build activity

- ▶ Purpose: Compile source files into deployable binaries, verify that code functions as the developer(s) intended, and merge dev branches to trunk
- ▶ **Skills:**
 - Continuous code integration
 - Build and test automation
 - Trunk-based development
 - Gated commit
 - Application security

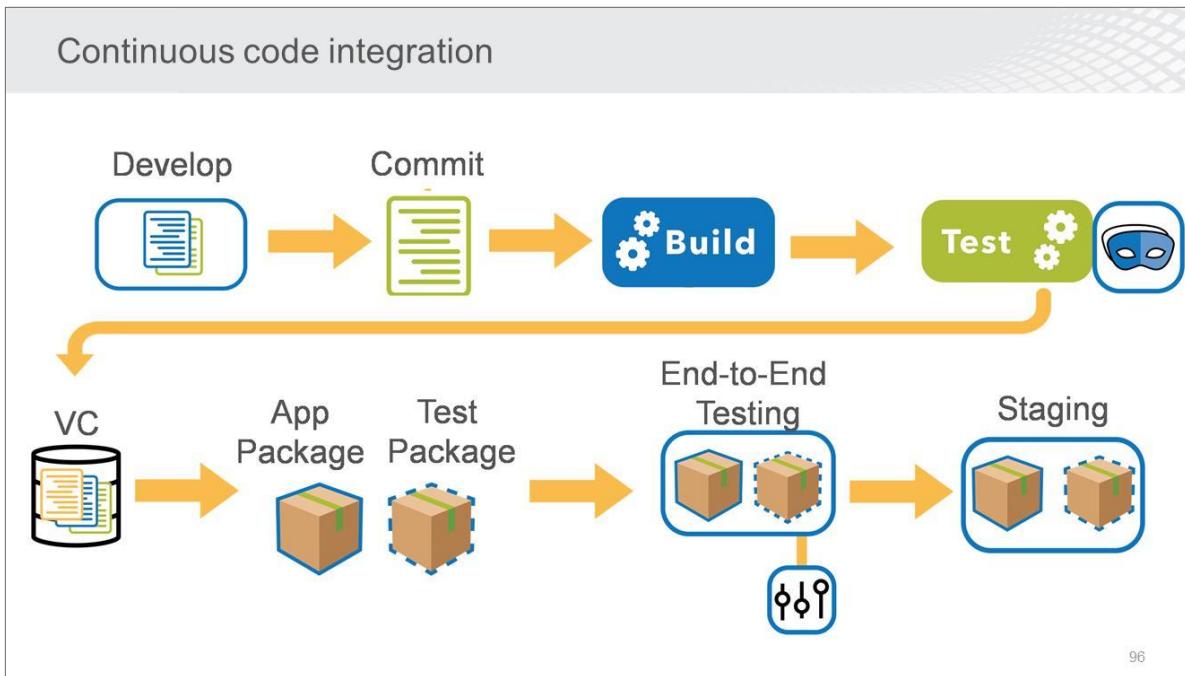


SCALED AGILE® © Scaled Agile, Inc.

95

Notes:

4.2 Build continuously

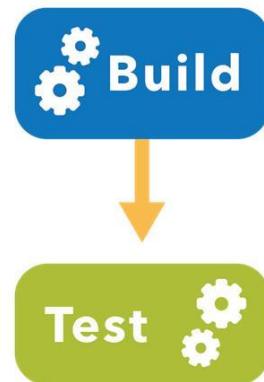


96

Notes:

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



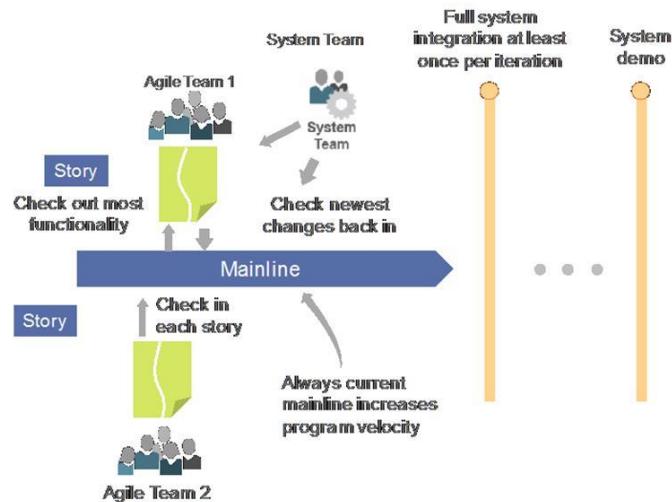
97

Notes:

4.2 Build continuously

Trunk-based development

- ▶ Single trunk/main for all teams
- ▶ Each commit merges to main
- ▶ Avoid long-lived branches
- ▶ Avoid multiple open branches



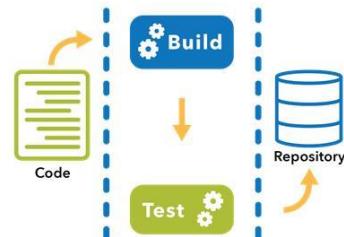
SCALED AGILE® © Scaled Agile, Inc.

98

Notes:

Gated commit

- ▶ Gated commits ensure that broken code doesn't block the rest of the developers or the pipeline
- ▶ Only changes that have passed all build and quality checks are committed to version control
- ▶ Alert the code author immediately upon rejection



SCALED AGILE® © Scaled Agile, Inc.

99

Notes:

4.2 Build continuously

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.

100

Notes:

Self-assessment: Build

1. Sit	Builds are run fewer than once per iteration and/or are completely manual.
2. Crawl	Builds are run once per iteration and are partially automated; dev branches are open for a month or more; builds break often.
3. Walk	Automated builds run once a day; broken builds are corrected in 2 – 4 hours; manual unit tests are run against each build; dev branches are open for 2 – 4 weeks.
4. Run	Builds run automatically upon code commit; broken builds are corrected within 1 hour; automated unit tests are run against each build; dev branches are merged to trunk every iteration.
5. Fly	Builds run on every commit; builds include static code analysis and security testing; gated commits prevent defects from entering the version control; dev branches are merged to trunk on every commit.

Duration: 1 min

The DevOps Radar diagram is a circular chart divided into three main quadrants: **Continuous Experimentation** (top), **Continuous Deployment** (bottom-left), and **Continuous Integration** (bottom-right). The center of the radar is labeled "Build & Release". The outer ring contains various DevOps practices: "Release on Demand", "Refactor", "Automate", "Optimize", "Collaborate & Research", "Acquire", "Simplify", "Deploy", "Test", "End-to-end", "Verify", "Stage", "Deploy", "Continuous Integration", "Continuous Deployment", "Continuous Experimentation", "Hypothesize", "Measure", "Learn", "Refine", "Stabilize", and "Release".

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

101

Notes:

4.3 Test end-to-end

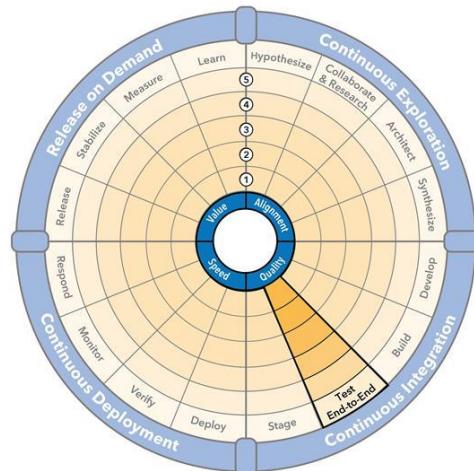
SCALED AGILE® © Scaled Agile, Inc.

102

Notes:

Test End-to-End

- ▶ Purpose: To validate changes against acceptance criteria in an integrated, production-simulated environment
- ▶ Skills:
 - Test and production environment congruity
 - Test automation
 - Test data management
 - Service virtualization
 - Nonfunctional requirements



SCALED AGILE® © Scaled Agile, Inc.

103

Notes:

Test and production environment congruity

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

—Em Campbell-Pretty, Tribal Unity

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

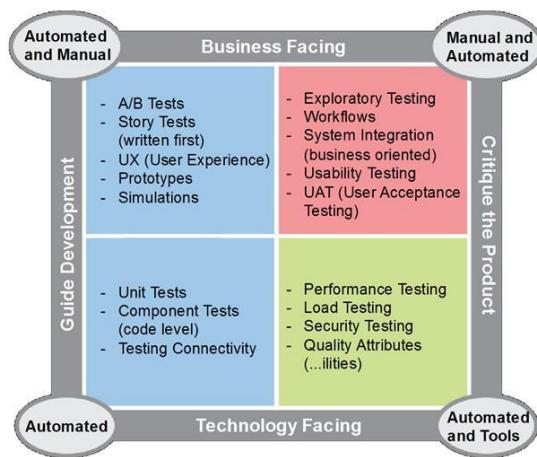
SCALED AGILE® © Scaled Agile, Inc.

104

Notes:

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



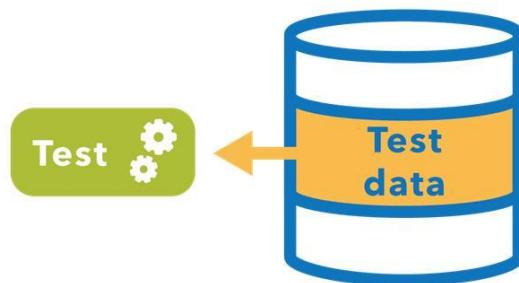
SCALED AGILE® © Scaled Agile, Inc.

105

Notes:

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



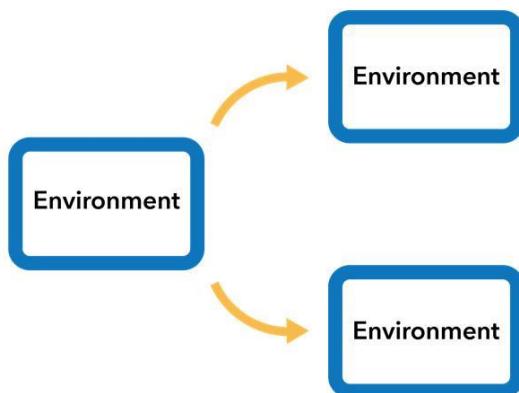
SCALED AGILE® © Scaled Agile, Inc.

106

Notes:

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.

107

Notes:

4.3 Test end-to-end

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.
- ▶ NFRs are constraints on backlog items
- ▶ NFRs are not backlog items themselves



SCALED AGILE® © Scaled Agile, Inc.

108

Notes:

Self-assessment: Test End-to-End

1. Sit	Testing is performed manually in environments that do not mimic production; testing occurs in large batches during a scheduled testing phase.
2. Crawl	Testing is mostly manual in non-production-like environments; Stories are implemented and tested independently within a single PI.
3. Walk	Half of the testing is automated and performed in production-like or production-simulated environments every PI.
4. Run	The majority of tests are automated and run in production-like environments; Stories are implemented and fully tested every Iteration.
5. Fly	Successful builds trigger automatic deployment to production-like test environments; all tests are automated; tests run in parallel and changes are fully validated after every commit.

Duration

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

109

Notes:

90 | © Scaled Agile, Inc.

For personal use only - Viktorija Marimiseva - Sep 14, 2020 (ID:0050W000007ELwNQAW)

4.4 Validate on a staging environment

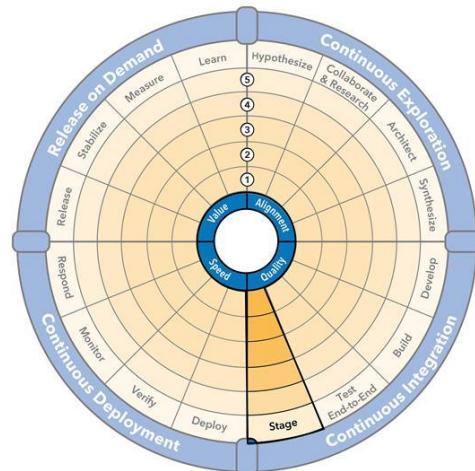
SCALED AGILE® © Scaled Agile, Inc.

110

Notes:

Stage

- ▶ Purpose: Host fully-validated systems in a production-grade environment from which they can be deployed to production
- ▶ Skills:
 - Maintain a staging environment
 - Blue/green deployment
 - System demo



SCALED AGILE® © Scaled Agile, Inc.

111

Notes:

4.4 Validate on a staging environment

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
- ▶ Ideally, deploy to staging automatically after all build and end-to-end tests have passed



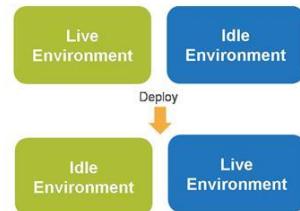
SCALED AGILE® © Scaled Agile, Inc.

112

Notes:

Blue/green deployment

- ▶ Maintain two identical environments: idle and live.
- ▶ New Features are deployed to the idle environment on a continuous basis. While staged, the Features can be tested and showcased in preparation for release.
- ▶ New code is released to the idle environment where it is thoroughly tested. When the code is ready to be released, the team makes the idle environment active.
- ▶ If problems are discovered after the switch, the active environment is switched back to idle, restoring the previously active environment.
- ▶ Switching between environments is typically done by redirecting traffic at the load balancer.



SCALED AGILE® © Scaled Agile, Inc.

113

Notes:

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.

114

Notes:

4.4 Validate on a staging environment

Self-assessment: Stage
Duration
1 min

1. Sit	No staging environment exists or we use a test environment for staging.
2. Crawl	Features are deployed manually to a staging environment once every PI.
3. Walk	Features are deployed to a staging environment once per month and demonstrated to Product Management.
4. Run	Features and infrastructure are auto-deployed to a staging environment every iteration and accepted by Product Management.
5. Fly	Stories, changes, and infrastructure are auto-deployed to a staging environment, and deployment is validated.

The DevOps Transformation Canvas is a circular diagram divided into three concentric rings. The innermost ring is blue and labeled 'Continuous Experimentation' at the top right, 'Autonomise' at the bottom, and 'Release on Demand' at the top left. The middle ring is yellow and contains various DevOps practices: 'Continuous Deployment' (left), 'Continuous Integration' (right), 'Stage' (center), 'Build', 'Test', 'Deploy', 'Verify', 'Revert', 'Reserve', 'Release', 'Service', 'Measure', 'Learn', 'Hypothesise', 'Collaborate & Iterate', 'Adapt', and 'Simplify'. The outermost ring is light blue and labeled 'Continuous Exploration' at the top right, 'Continuous Deployment' at the bottom left, and 'Continuous Integration' at the bottom right.

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.
115

Notes:

Activity: Apply Continuous Integration to your Value Stream
Prepare
15 min
Share
5 min

- ▶ **Step 1:** Identify the skills in this lesson that would improve flow in your Value Stream between ready to code and ready to deploy. Write each on a star-shaped sticky note.
- ▶ **Step 2:** Construct a new view of this part of the Value Stream in the Future State portion of the DevOps Transformation Canvas. (Reuse, reorder, or replace steps as desired.)
- ▶ **Step 3:** Place the stars on the steps they would enable and provide new process time, lead time, and %C&A for each step.

The diagram shows a simplified value stream with three main steps: CODE, TEST, and PM ACCEPTS. Each step has a yellow star above it. The CODE step has a TDD label. The TEST step has a TEST DATA MNGT label. The PM ACCEPTS step has a PT: 2h LT: 40h %C&A: 100% label. Arrows indicate the flow from CODE to TEST and from TEST to PM ACCEPTS. A legend at the top right indicates that yellow stars represent 'Example'.

SCALED AGILE® © Scaled Agile, Inc.
116

Notes:

Lesson review

In this lesson you:

- ▶ Explored how to develop the Solution
- ▶ Discussed building continuously
- ▶ Analyzed testing end-to-end
- ▶ Explored how to validate on a staging environment

Notes:

Lesson 4 notes



Click below to type your thoughts.

Lesson 5

Reducing Time-to-Market with Continuous Deployment

Learning Objectives:

- 5.1 Deploy to production
- 5.2 Verify the Solution
- 5.3 Monitor for problems
- 5.4 Respond and recover



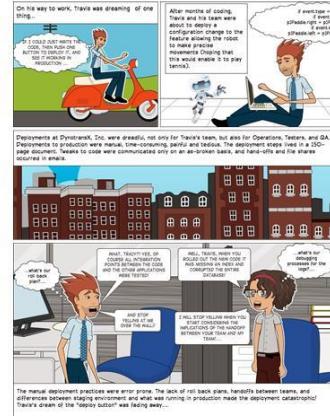
SAFe® Course Attending this course gives students access to the SAFe® DevOps Practitioner exam and related preparation materials.



Activity: Deployment Gone Wrong



- ▶ **Step 1:** Read the story, “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.



SCALED AGILE® © Scaled Agile, Inc.

118

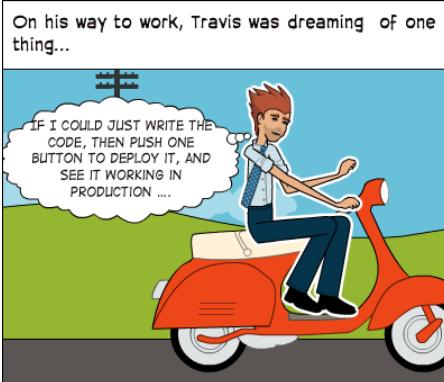
Notes:

Considering some of the immediate solutions suggested by your character, could any of them be the right course of action?

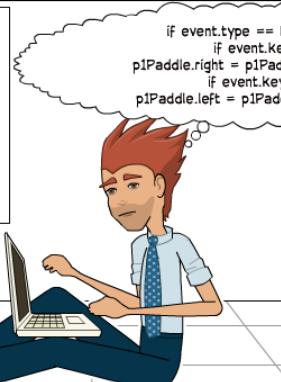


- Build immutable infrastructure
- Recovery issues are not real issues
- Add more servers
- Resolve production issues before they cause business disruption

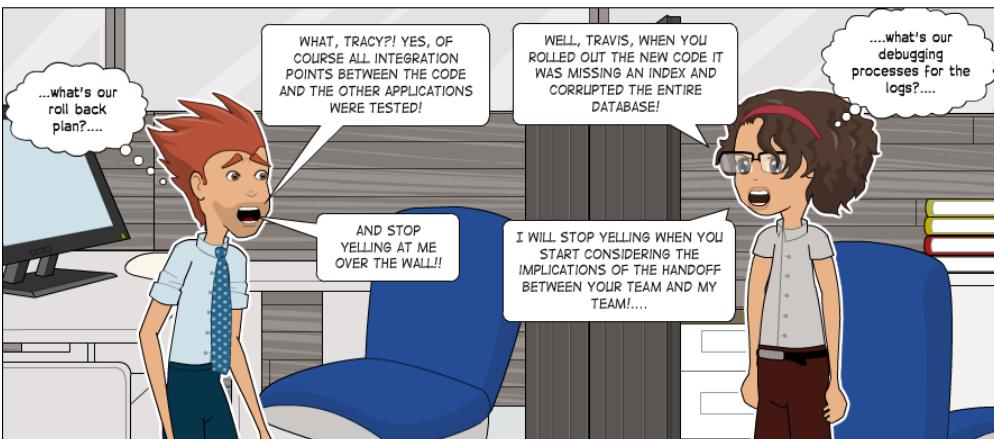
- Deploy to select production environments
- Create two databases: blue and orange
- Match the staging and prod environments
- Decouple database changes from app changes



After months of coding, Travis and his team were about to deploy a configuration change to the feature allowing the robot to make precise movements (hoping that this would enable it to play tennis).



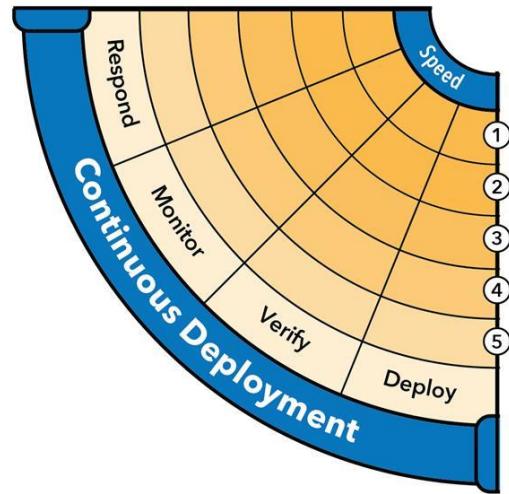
Deployments at DynotransX, Inc. were dreadful, not only for Travis's team, but also for Operations, Testers, and QA. Deployments to production were manual, time-consuming, painful and tedious. The deployment steps lived in a 150-page document. Tweaks to code were communicated only on an as-broken basis, and hand-offs and file shares occurred in emails.



The manual deployment practices were error prone. The lack of roll back plans, handoffs between teams, and differences between staging environment and what was running in production made the deployment catastrophic! Travis's dream of the "deploy button" was fading away....

Deployment

- ▶ Business objective
 - Time-to-market
- ▶ IT objective
 - Continuous Deployment (CD)
- ▶ Activities
 - Deploy
 - Verify
 - Monitor
 - Respond



SCALED AGILE® © Scaled Agile, Inc.

119

Notes:

5.1 Deploy to production

5.1 Deploy to production

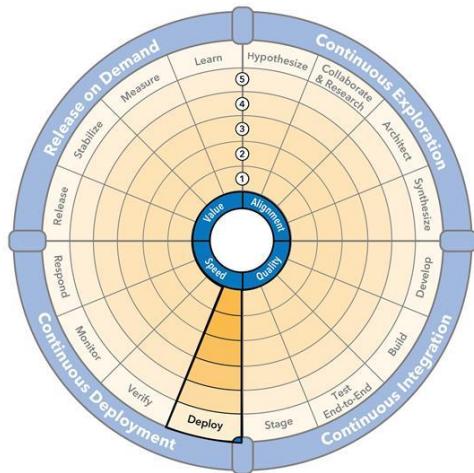
SCALED AGILE® © Scaled Agile, Inc.

120

Notes:

Deploy

- ▶ Purpose: Deploy changes into production with high frequency and low risk
- ▶ Skills:
 - Dark launches
 - Feature toggles
 - Infrastructure as code
 - Deployment automation
 - Self-service deployment
 - Selective deployment
 - Version control (see Build)
 - Blue/green deployment (see Stage)



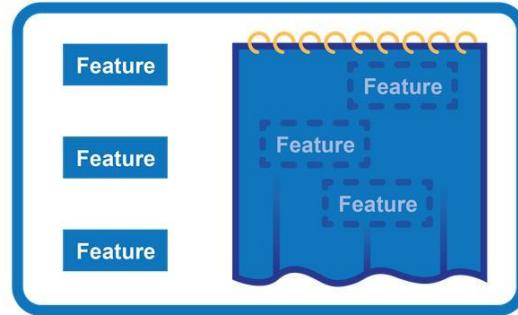
SCALED AGILE® © Scaled Agile, Inc.

121

Notes:

Dark launches

- ▶ Separate deploy (to production) from release (to end users)
- ▶ Enables testing and monitoring system behavior in the actual production environment before exposing new functionality to users



SCALED AGILE® © Scaled Agile, Inc.

122

Notes:

Feature toggles

- ▶ Dynamically show and hide Features in production
- ▶ Enables separation of deploy and release
- ▶ Enables rapid rollback of problematic Features
- ▶ Test toggles in both on and off position
- ▶ *Be careful of toggle overload and testing complexity*



Be careful of toggle overload and testing complexity

SCALED AGILE® © Scaled Agile, Inc.

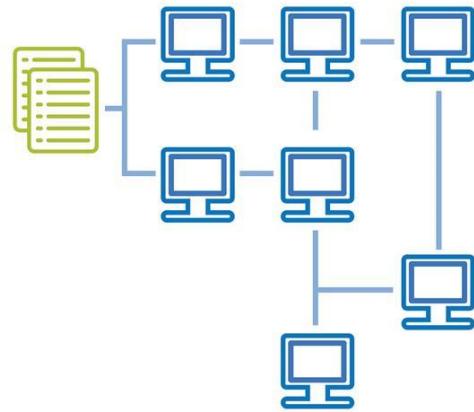
123

Notes:

5.1 Deploy to production

Infrastructure as code

- ▶ Automates environment setup and teardown
- ▶ Manages all infrastructure assets and configurations in version control
- ▶ Accelerates and de-risks deployment by provisioning standard/gold environments on demand
- ▶ Enables infrastructure to always be in a deployable state just like application code



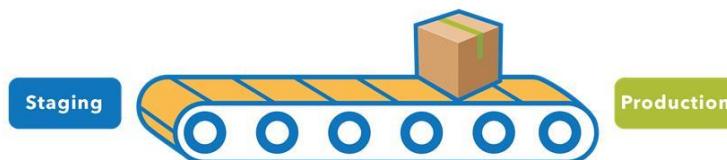
SCALED AGILE® © Scaled Agile, Inc.

124

Notes:

Deployment automation

- ▶ Automate all 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.

125

Notes:

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.

126

Notes:

Selective deployment

- ▶ Deploy to select production targets or environments
- ▶ This can be differentiated by data center, geography, or Customers
- ▶ Enables more flexible and sophisticated release strategies



SCALED AGILE® © Scaled Agile, Inc.

127

Notes:

5.1 Deploy to production

Self-assessment: Deploy

Assess 1 min

1. Sit	Features are deployed to production every 3+ months; deployments are manual and painful; deployed means released.
2. Crawl	Features are deployed to production at PI boundaries; deployments are mostly manual; deployed means released.
3. Walk	Features are deployed to production every Iteration; deployments are mostly automated; some Features can be deployed without being released.
4. Run	Features are deployed to production every Iteration and fully automated through the Continuous Delivery Pipeline; dark releases are common.
5. Fly	Features are deployed continuously throughout each Iteration; Agile Teams initiate deployments directly via Continuous Delivery Pipeline tools; release is completely decoupled from deployment; dark releases are the norm.

The DevOps Radar diagram is a circular chart divided into four quadrants. The top-left quadrant is labeled "Release on Demand" and contains "Initiate", "Build", "Test", "Deploy", "Stop", "Verify", "Refine", and "Responsible". The top-right quadrant is labeled "Continuous Exploration" and contains "Continuous Integration", "Develop", "Automate", "Collaborate & Research", "Hypothesize", "Learn", "Measure", and "Subdivide". The bottom-left quadrant is labeled "Continuous Deployment" and contains "Automate", "Develop", "Build", "Test", "Deploy", "Stop", "Verify", and "Responsible". The bottom-right quadrant is labeled "Continuous Delivery Pipeline" and contains "Initiate", "Build", "Test", "Deploy", "Stop", "Verify", "Refine", and "Responsible". The center of the radar is labeled "Agile Team".

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

Notes:

5.2 Verify the Solution

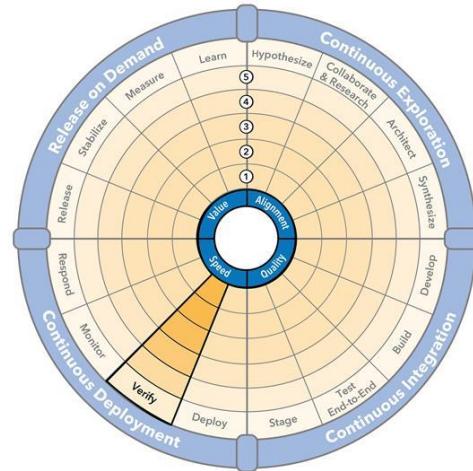
SCALED AGILE® © Scaled Agile, Inc.

129

Notes:

Verify

- ▶ Purpose: Assure that deployment Solutions behave as expected in production before they are released to end users
- ▶ Skills:
 - Production testing
 - Test automation (see build)
 - Test data management (see test end-to-end)
 - Nonfunctional requirements (see test end-to-end)



SCALED AGILE® © Scaled Agile, Inc.

130

Notes:

5.2 Verify the Solution

Production testing

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



SCALED AGILE® © Scaled Agile, Inc.

131

Notes:

Self-assessment: Verify

1. Sit	Deployments are not verified in production before being released to end users.
2. Crawl	Deployments are verified with manual smoke tests and/or UAT; we address deployment issues within a stated grace/warranty period; recovery is painful or impossible; we often correct issues directly in production.
3. Walk	Deployments are verified with manual tests prior to releasing to end users; recovery is painful; we do not make changes directly in production.
4. Run	Deployments are verified using automated production tests, synthetic transactions, and security tests prior to release; we can easily roll back or fix forward to recover from failed deployments.
5. Fly	Automated production tests run on an ongoing basis and feed monitoring systems; failed deployments can be rolled back instantly or fixed forward through the entire Continuous Delivery Pipeline.

Assess

Read the table and place a dot on your DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

132

Notes:

108 | © Scaled Agile, Inc.

For personal use only - Viktorija Marimiseva - Sep 14, 2020 (ID:0050W000007ELwNQAW)

5.3 Monitor for problems

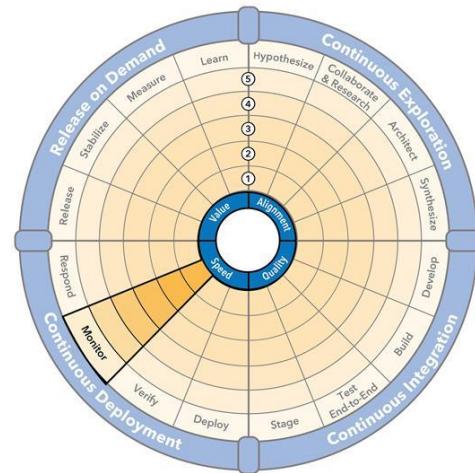
SCALED AGILE® © Scaled Agile, Inc.

133

Notes:

Monitor

- ▶ Purpose: Quantitatively measure system and user behavior in real time
- ▶ Skills:
 - Full-stack telemetry
 - Visual displays
 - Federated monitoring
 - AIOps



SCALED AGILE® © Scaled Agile, Inc.

134

Notes:

Full-stack telemetry

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



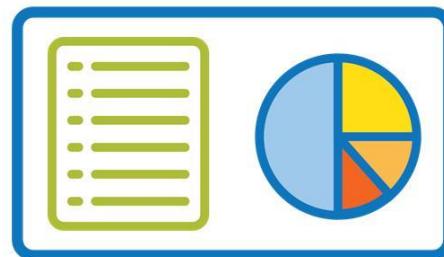
SCALED AGILE® © Scaled Agile, Inc.

135

Notes:

Visual displays

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



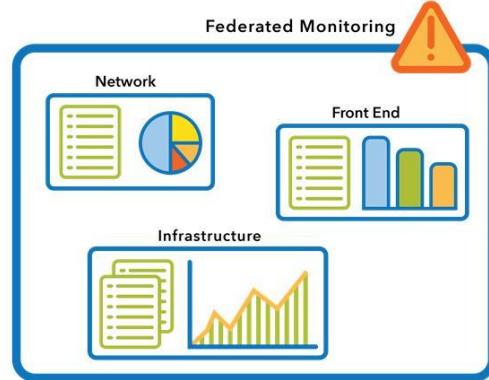
SCALED AGILE® © Scaled Agile, Inc.

136

Notes:

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



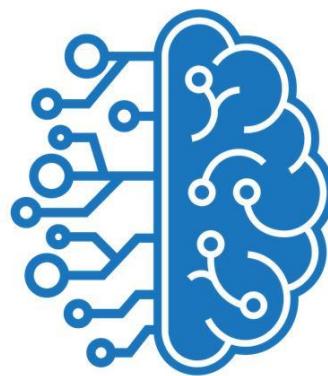
SCALED AGILE® © Scaled Agile, Inc.

137

Notes:

Artificial Intelligence for Operations (AIOps)

- ▶ The monitoring required for effective DevOps produces a flurry of data, events, and alerts
- ▶ AIOps uses machine learning and big data techniques to quickly:
 - Aggregate, correlate, and analyze events
 - Separate meaningful events from the 'noise'
 - Identify and predict root causes of issues
 - Significantly reduce mean time to Restore (MTTR)

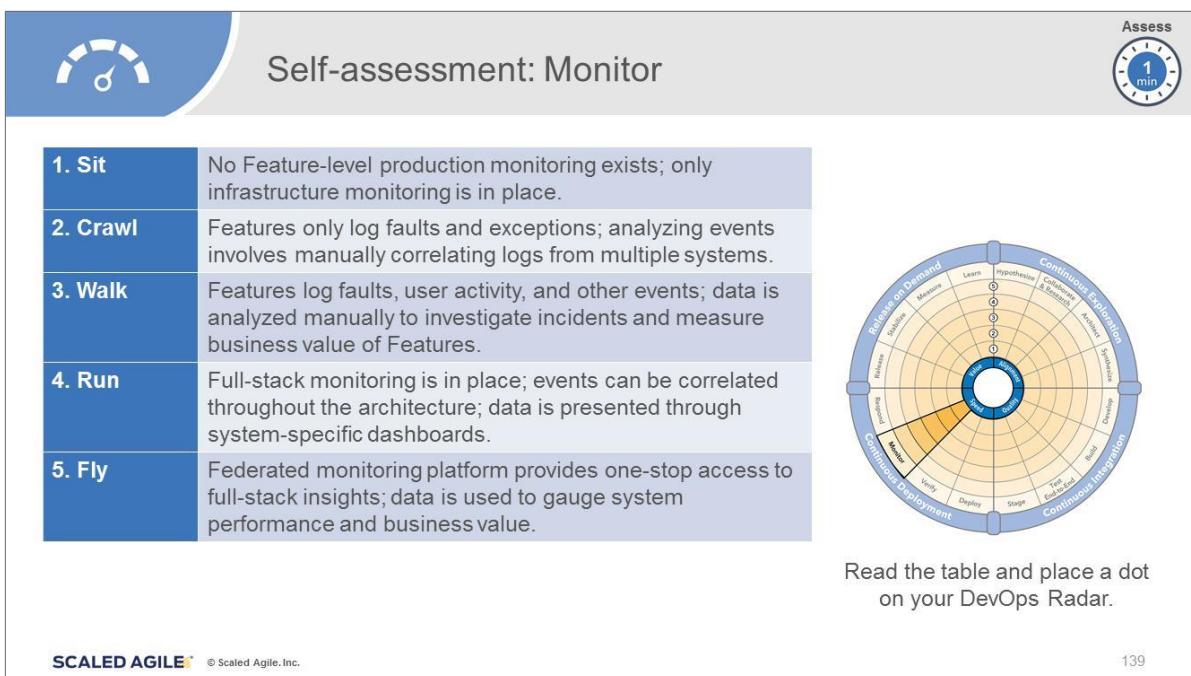


SCALED AGILE® © Scaled Agile, Inc.

138

Notes:

5.3 Monitor for problems



The slide features a blue header bar with a speedometer icon on the left and a timer icon with "Assess" and "1 min" on the right. Below the header is a title "Self-assessment: Monitor". A table below the title lists five levels of monitoring from Sit to Fly. To the right of the table is a circular "DevOps Radar" diagram divided into four quadrants: Continuous Exploration (top), Continuous Deployment (bottom-left), Continuous Integration (bottom-right), and Release on Demand (top-left). The center of the radar is labeled "Adaptive System".

1. Sit	No Feature-level production monitoring exists; only infrastructure monitoring is in place.
2. Crawl	Features only log faults and exceptions; analyzing events involves manually correlating logs from multiple systems.
3. Walk	Features log faults, user activity, and other events; data is analyzed manually to investigate incidents and measure business value of Features.
4. Run	Full-stack monitoring is in place; events can be correlated throughout the architecture; data is presented through system-specific dashboards.
5. Fly	Federated monitoring platform provides one-stop access to full-stack insights; data is used to gauge system performance and business value.

Read the table and place a dot on your DevOps Radar.

Notes:

5.4 Respond and recover

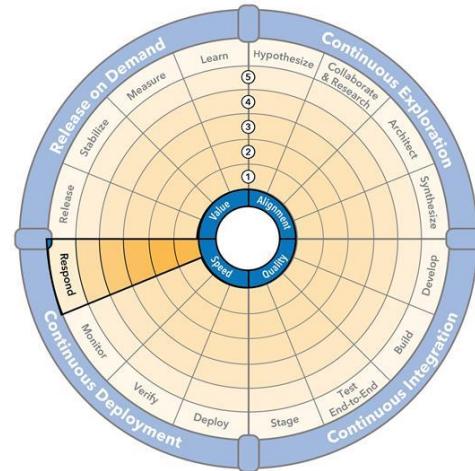
SCALED AGILE® © Scaled Agile, Inc.

140

Notes:

Respond

- ▶ Purpose: Proactively detect and resolve production issues before they cause business disruption
- ▶ Skills:
 - Chaos engineering
 - Proactive detection
 - Cross-team collaboration
 - Session replay
 - Rollback and fix forward
 - Immutable infrastructure
 - Version control (see Build)



SCALED AGILE® © Scaled Agile, Inc.

141

Notes:



Discussion: Chaos engineering



- ▶ **Step 1:** Read the blog post in your workbook
- ▶ **Step 2:** Discuss as a team:
 - How did Netflix prepare for the AWS outage?
 - Would your systems be prepared for a similar event?
 - How might your organization apply chaos engineering?

SCALED AGILE® © Scaled Agile, Inc.

142

Notes:

The Netflix Simian Army

Source: Netflix Technology Blog, Jul 19, 2011

The cloud is all about redundancy and fault-tolerance. Since no single component can guarantee 100% uptime (and even the most expensive hardware eventually fails), we have to design a cloud architecture where individual components can fail without affecting the availability of the entire system. In effect, we have to be stronger than our weakest link. We can use techniques like graceful degradation on dependency failures, as well as node-, rack-, datacenter-/availability-zone-, and even regionally-redundant deployments. But just designing a fault tolerant architecture is not enough. We have to constantly test our ability to actually survive these "once in a blue moon" failures.

Imagine getting a flat tire. Even if you have a spare tire in your trunk, do you know if it is inflated? Do you have the tools to change it? And, most importantly, do you remember how to do it right? One way to make sure you can deal with a flat tire on the freeway, in the rain, in the middle of the night is to poke a hole in your tire once a week in your driveway on a Sunday afternoon and go through the drill of replacing it. This is expensive and time-consuming in the real world, but can be (almost) free and automated in the cloud.

This was our philosophy when we built Chaos Monkey, a tool that randomly disables our production instances to make sure we can survive this common type of failure without any customer impact. The name comes from the idea of unleashing a wild monkey with a weapon in your data center (or cloud region) to randomly shoot down instances and chew through cables—all the while we continue serving our customers without interruption.

5.4 Respond and recover

By running Chaos Monkey in the middle of a business day, in a carefully monitored environment with engineers standing by to address any problems, we can still learn the lessons about the weaknesses of our system, and build automatic recovery mechanisms to deal with them. So next time an instance fails at 3 am on a Sunday, we won't even notice.

Inspired by the success of the Chaos Monkey, we've started creating new simians that induce various kinds of failures, or detect abnormal conditions, and test our ability to survive them; a virtual Simian Army to keep our cloud safe, secure, and highly available.

Latency Monkey induces artificial delays in our RESTful client-server communication layer to simulate service degradation and measures if upstream services respond appropriately. In addition, by making very large delays, we can simulate a node or even an entire service downtime (and test our ability to survive it) without physically bringing these instances down. This can be particularly useful when testing the fault-tolerance of a new service by simulating the failure of its dependencies, without making these dependencies unavailable to the rest of the system.

Conformity Monkey finds instances that don't adhere to best-practices and shuts them down. For example, we know that if we find instances that don't belong to an auto-scaling group, that's trouble waiting to happen. We shut them down to give the service owner the opportunity to re-launch them properly.

Doctor Monkey taps into health checks that run on each instance as well as monitors other external signs of health (e.g. CPU load) to detect unhealthy instances. Once unhealthy instances are detected, they are removed from service and after giving the service owners time to root-cause the problem, are eventually terminated.

Janitor Monkey ensures that our cloud environment is running free of clutter and waste. It searches for unused resources and disposes of them.

Security Monkey is an extension of Conformity Monkey. It finds security violations or vulnerabilities, such as improperly configured AWS security groups, and terminates the offending instances. It also ensures that all our SSL and DRM certificates are valid and are not coming up for renewal.

10-18 Monkey (short for Localization-Internationalization, or l10n-i18n) detects configuration and run time problems in instances serving customers in multiple geographic regions, using different languages and character sets.

Chaos Gorilla is similar to Chaos Monkey, but simulates an outage of an entire Amazon availability zone. We want to verify that our services automatically re-balance to the functional availability zones without user-visible impact or manual intervention.

With the ever-growing Netflix Simian Army by our side, constantly testing our resilience to all sorts of failures, we feel much more confident about our ability to deal with the inevitable failures that we'll encounter in production and to minimize or eliminate their impact to our subscribers. The cloud model is quite new for us (and the rest of the industry); fault-tolerance is a work in progress and we have ways to go to fully realize its benefits. Parts of the Simian Army have already been built, but much remains an aspiration—waiting for talented engineers to join the effort and make it a reality.

Ideas for new simians are coming in faster than we can keep up and if you have ideas, we'd love to hear them! The Simian Army is one of many initiatives we've launched to put the spotlight on increasing the reliability of our service and delivering to our customers an uninterrupted stream of entertainment.

Yury Izrailevsky, Director of Cloud & Systems Infrastructure

Ariel Tseitlin, Director of Cloud Solutions

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.

143

Notes:

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



SCALED AGILE® © Scaled Agile, Inc.

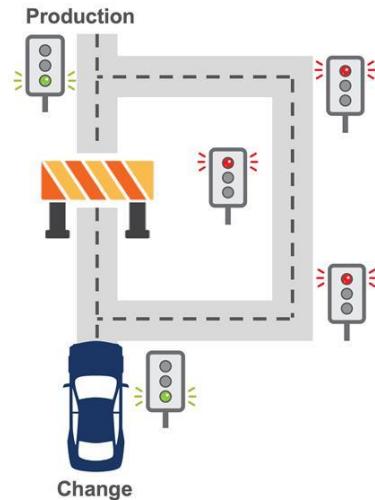
144

Notes:

Immutable infrastructure

Production

- ▶ 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.

145

Notes:

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.

146

Notes:

5.4 Respond and recover

Self-assessment: Respond

Duration: 1 min

1. Sit	Customers find issues before we do; resolving high priority issues is time consuming and reactive; Customers have low confidence in our ability to recover from production issues.
2. Crawl	Operations owns production issues; development involvement requires significant escalation; teams blame each other in times of crisis.
3. Walk	Our monitoring systems detect most issues before our Customers do; Dev and Ops work proactively to recover from major incidents.
4. Run	Our monitoring systems detect most issues before our Customers do; Dev and Ops work proactively to recover from major incidents.
5. Fly	Our monitoring systems alert us to dangerous conditions based on carefully designed tolerance thresholds; developers are responsible for supporting their own code and proactively issue fixes through the pipeline before users are affected.

SCALED AGILE® © Scaled Agile, Inc.

The DevOps Radar diagram is a circular chart divided into four quadrants. The top-right quadrant is labeled "Continuous Exploration" and contains the following clockwise cycle: Release on Demand, Subsidize, Measure, Learn, Hypothesize, Collaborate & Research, Assess, Implement, and Optimize. The bottom-right quadrant is labeled "Continuous Deployment" and contains the following clockwise cycle: Release, Revert, Verify, Deploy, Stage, Build, Continuous Integration, and Develop. The center of the radar is labeled "DevOps".

Read the table and place a dot on the DevOps Radar.

147

Notes:

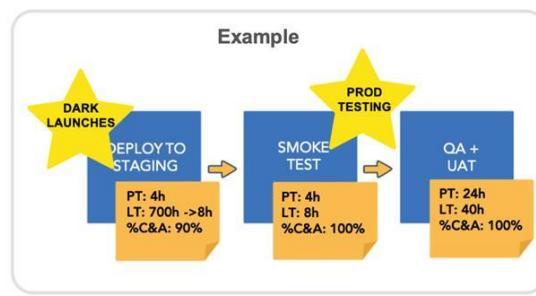


Activity: Apply Continuous Deployment to your Value Stream

Prepare
15 min

Share
5 min

- ▶ **Step 1:** Identify the skills in this lesson that would improve flow in your Value Stream between ready to deploy and ready to release. Write each on a star-shaped sticky note.
- ▶ **Step 2:** Construct a new view of this part of the Value Stream in the Future State portion of the DevOps Transformation Canvas. (Reuse, reorder, or replace steps as desired.)
- ▶ **Step 3:** Place the stars on the steps they would enable and provide new process time, lead time, and %C&A for each step.



SCALED AGILE® © Scaled Agile, Inc.

148

Notes:

Lesson review

In this lesson you:

- ▶ Explored how to deploy to production
- ▶ Discussed how to verify the Solution
- ▶ Analyzed how to monitor for problems
- ▶ Discussed how to respond and recover

Notes:

Lesson 5 notes



Click below to type your thoughts.

5.4 Respond and recover

Lesson 6

Delivering Business Value with Release on Demand

Learning Objectives:

- 6.1 Release on Demand
- 6.2 Stabilize the Solution
- 6.3 Measure the business value
- 6.4 Learn and react



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



Activity: What did we release?

Prepare
10 min
Share
5 min

- ▶ **Step 1:** Read the story, “What did we release?”
- ▶ **Step 2:** Review the clues provided by the characters.
- ▶ **Step 3:** Considering your own context, discuss:
 - What are some actions you could take to prevent situations like the one at DynotransX from occurring so often?



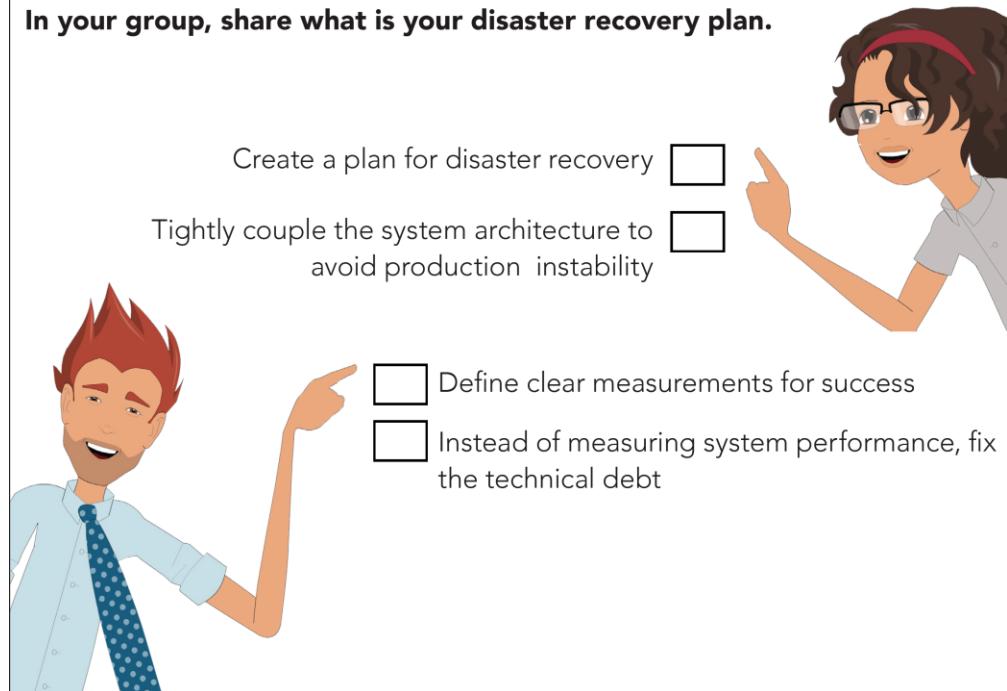
SCALED AGILE® © Scaled Agile, Inc.

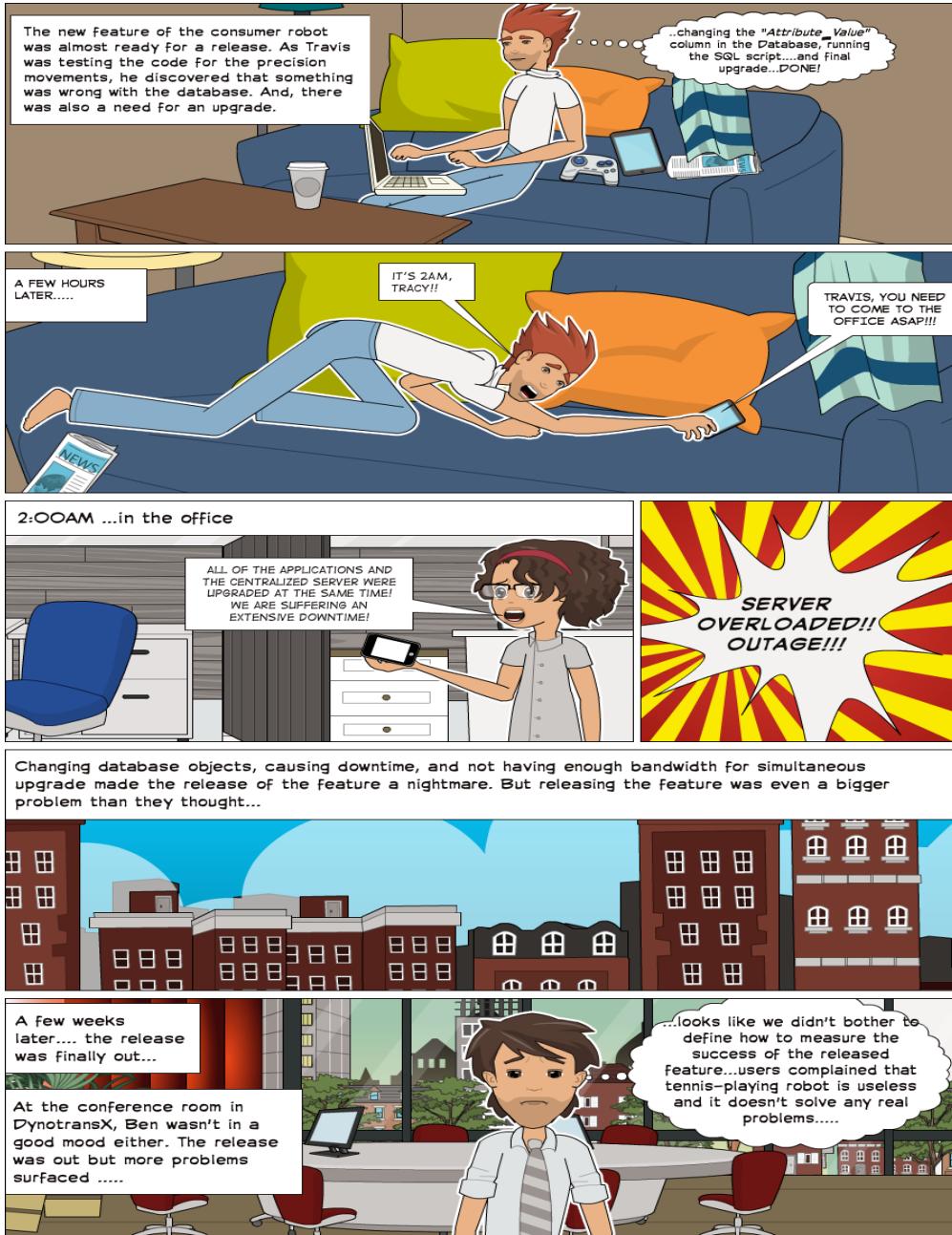
150

Notes:

What could you do in your company to prevent situations like the one at DynotransX from occurring so often?

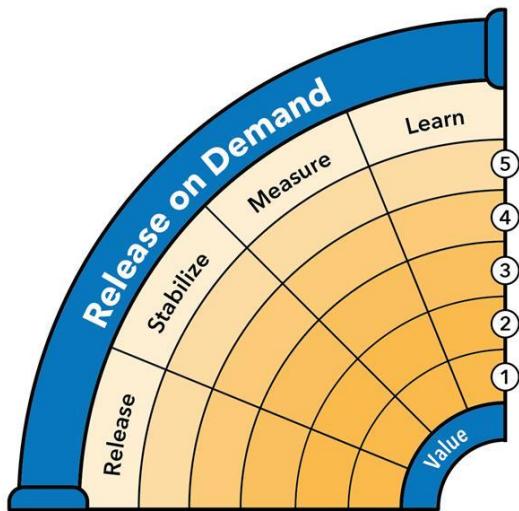
In your group, share what is your disaster recovery plan.





Release on Demand

- ▶ Business objective
 - Business value
- ▶ IT objective
 - Release on Demand
- ▶ Activities
 - Release
 - Stabilize
 - Measure
 - Learn



SCALED AGILE® © Scaled Agile, Inc.

151

Notes:

6.1 Release on Demand

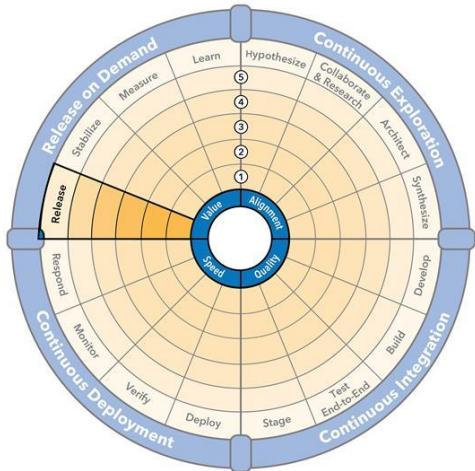
SCALED AGILE® © Scaled Agile, Inc.

152

Notes:

Release

- ▶ Purpose: Release value to Customers all at once or incrementally
- ▶ Skills:
 - Canary releases
 - Feature toggles (see Deploy)
 - Decouple release elements (see Architect)
 - Dark launches (see Deploy)



SCALED AGILE® © Scaled Agile, Inc.

153

Notes:

6.1 Release on Demand

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 roll out



SCALED AGILE® © Scaled Agile, Inc.

154

Notes:

The page contains two main sections: a self-assessment table and a DevOps Radar diagram.

Self-assessment: Release activity

Activity Level	Description
1. Sit	Releases are tightly coupled to deployments, and Customers are extremely dissatisfied with the frequency of releases.
2. Crawl	Releases are tightly coupled to deployments, but Customers are somewhat dissatisfied with the frequency of releases.
3. Walk	Release and deployment are coupled but both occur continuously or on demand.
4. Run	Release is decoupled from deployment; deployed Features are released to the end user population based on business readiness.
5. Fly	Deployed Features can be released to individual segments of the user population; Feature toggles are refactored when no longer used.

DevOps Radar

The DevOps Radar is a circular diagram divided into four quadrants:

- Top Quadrant (Continuous Exploration):** Hypothesize, Calibrate & Iterate, Architect, Design.
- Bottom Quadrant (Continuous Deployment):** Release, Integrate, Automate, Verify, Deploy.
- Left Quadrant (Release on Demand):** Measure, Learn.
- Right Quadrant (Continuous Integration):** Stage, Test, End-to-End.

A central circle contains the text "DevOps".

Duration: 1 min

SCALED AGILE® © Scaled Agile, Inc.

155

Notes:

131 | © Scaled Agile, Inc.

For personal use only - Viktorija Marimiseva - Sep 14, 2020 (ID:0050W000007ELwNQAW)

6.2 Stabilize the Solution

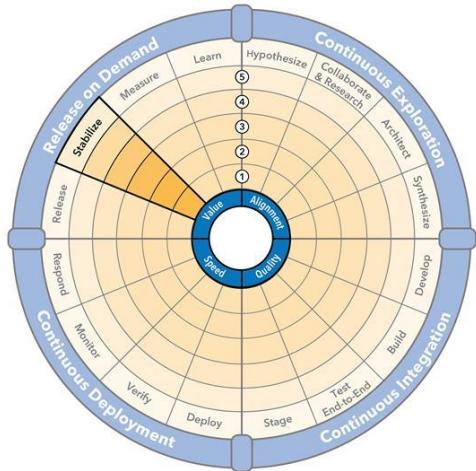
SCALED AGILE® © Scaled Agile, Inc.

156

Notes:

Stabilize

- ▶ Purpose: Assure sustainably high levels of business continuity, application service levels, and data protection
- ▶ Skills:
 - Site reliability engineering (SRE)
 - Failover/disaster recovery
 - Continuous security monitoring
 - Architect for operations (See Architect)
 - Nonfunctional requirements (See Test End-to-End)



SCALED AGILE® © Scaled Agile, Inc.

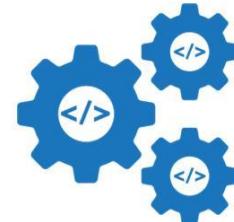
157

Notes:

Site reliability engineering (SRE)

Ensure large systems are highly reliable and scalable via:

- ▶ Shared ownership of system stability between Dev and Ops
- ▶ T-shaped engineers with deep development and operations expertise
- ▶ Approaching all operational activities as a software concern
- ▶ Closely managing Service Level Indicators (SLIs) and Service Level Objectives (SLOs)

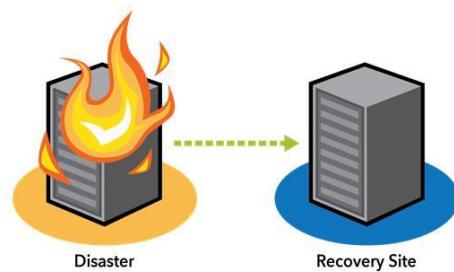


Notes:

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



Notes:

Continuous security monitoring

- ▶ Detect intrusions and attacks on production services and infrastructure (detective controls)
- ▶ Security as code and penetration testing focus on preventing known vulnerabilities from getting to production (preventive controls)
- ▶ Test services continuously for newly discovered and reported vulnerabilities
- ▶ Most breach studies show time to detect a breach is over 200 days, and typically breaches are detected by external parties rather than internal processes or monitoring



SCALED AGILE® © Scaled Agile, Inc.

160

Notes:

Self-assessment: Stabilize activity

Duration: 1 min

1. Sit	We experience frequent unplanned outages and/or security breaches with long recovery times.
2. Crawl	We experience occasional unplanned outages but recover within our service level agreements.
3. Walk	We have very few unplanned outages; availability, security, and disaster recovery measures are effective.
4. Run	We have no unplanned outages; we plan and rehearse failure and recovery.
5. Fly	We maximize resiliency by deliberately injecting faults into our production environment and rehearsing recovery procedures.

Read the table and place a dot on the DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

161

Notes:

6.3 Measure the business value

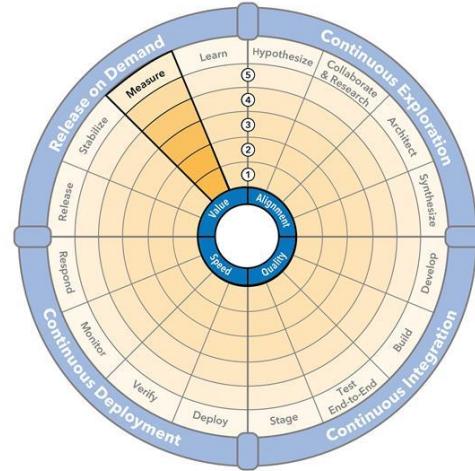
SCALED AGILE® © Scaled Agile, Inc.

162

Notes:

Measure

- ▶ Purpose: Determine actual business value delivered through feedback collected from production
- ▶ Skills:
 - Innovation accounting (see Hypothesize)
 - Evaluate hypothesis



SCALED AGILE® © Scaled Agile, Inc.

163

Notes:

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.

164

Notes:

Self-assessment: Measure activity

Duration: 1 min

1. Sit	We don't define or measure the value of Features.
2. Crawl	We've defined what 'value' is but don't know how to measure it.
3. Walk	We capture qualitative feedback from the business about the value of our Features.
4. Run	We capture qualitative and quantitative feedback from the business and our monitoring systems about the value of our Features.
5. Fly	We aggregate the quantitative and qualitative feedback to objectively validate the original hypothesis and inform pivot-or-persevere decisions.

Read the table and place a dot on the DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc.

165

Notes:

6.4 Learn and react

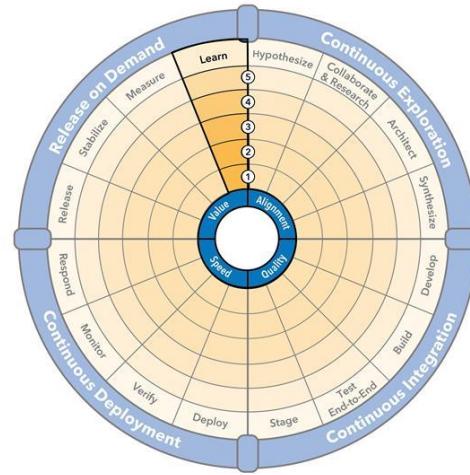
SCALED AGILE® © Scaled Agile, Inc.

166

Notes:

Learn

- ▶ Purpose: Learn from the hypothesis on whether to pivot or persevere as well as how to improve the flow of value
- ▶ Skills:
 - Lean startup (see Hypothesize)
 - Relentless improvement
 - Value Stream mapping



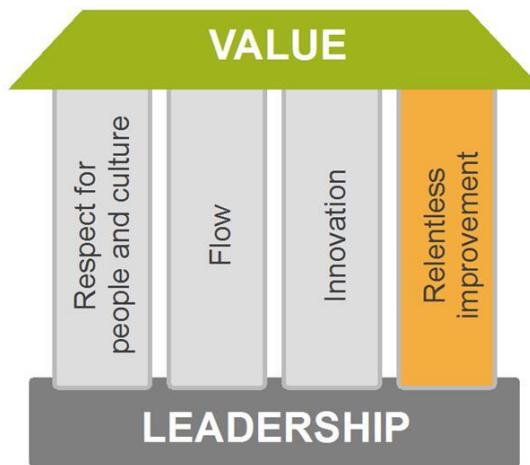
SCALED AGILE® © Scaled Agile, Inc.

167

Notes:

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



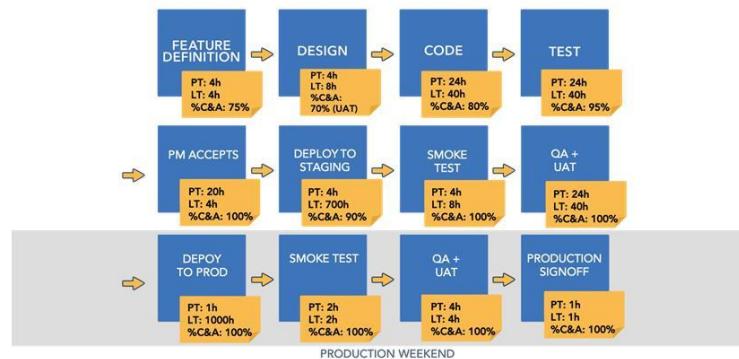
SCALED AGILE® © Scaled Agile, Inc.

168

Notes:

Value Stream mapping

- Continuously apply value stream mapping to optimize value flow
- Apply current state mapping and future state mapping so objectives are clear



SCALED AGILE® © Scaled Agile, Inc.

169

Notes:

6.4 Learn and react



Self-assessment: Learn

Duration
1 min

1. Sit	Features are never evaluated post-release.
2. Crawl	Features are sometimes evaluated using subjective information.
3. Walk	Hypotheses are evaluated using objective measures, but actions are heavily influenced by corporate politics.
4. Run	Hypotheses are objectively evaluated; pivot or persevere decisions are made without mercy or guilt.
5. Fly	Continuous learning and experimentation are ingrained in the DNA of the organization.



Read the table and place a dot on the DevOps Radar.

SCALED AGILE® © Scaled Agile, Inc. 170

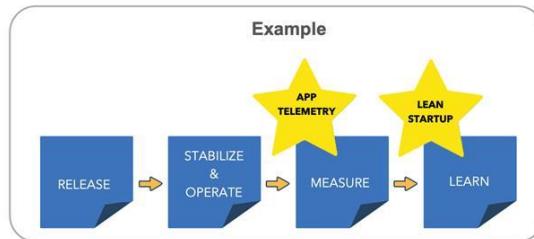
Notes:



Activity: Apply Release on Demand to your Value Stream



- ▶ **Step 1:** Identify the skills in this lesson that would improve your Value Stream between ready to release and hypothesis tested. Write each on a star-shaped sticky note.
- ▶ **Step 2:** Construct a new view of this part of the Value Stream in the Future State portion of the DevOps Transformation Canvas. (Reuse, reorder, or replace steps as desired.)
- ▶ **Step 3:** Place the stars on the steps they would enable and provide new process time, lead time, and %C&A for each step.



SCALED AGILE® © Scaled Agile, Inc.

171

Notes:

Lesson review

In this lesson you:

- ▶ Explored how to release on demand
- ▶ Discussed how to stabilize the Solution
- ▶ Explored how to measure the business value
- ▶ Discussed how to evaluate a hypothesis and make a decision

Notes:

Lesson 6 notes



Click below to type your thoughts.

Lesson 7

Taking Action

Learning Objectives:

7.1 Share your DevOps action plan



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

7.1 Share your DevOps action plan

Notes:



Taking action: Prioritize your DevOps backlog (Part 1)

Duration
5 min

- ▶ **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.

SCALED AGILE® © Scaled Agile, Inc.

174

Notes:

Improvement Item	Cost of delay	Job size	WSJF
	÷	=	
	÷	=	
	÷	=	
	÷	=	
	÷	=	
	÷	=	
	÷	=	
	÷	=	
	÷	=	

7.1 Share your DevOps action plan



Taking action: Prioritize your DevOps backlog (Part 2)

Duration
10 min

- ▶ **Step 1:** Estimate the cost of delay for all other items relative to the 1, using the numbers 1, 2, 3, 5, 8, 13, and 20
- ▶ **Step 2:** Estimate the job size of the items in a similar fashion; again, assign the smallest job size a 1
- ▶ **Step 3:** Calculate WSJF
- ▶ **Step 4:** Document the three highest priorities in the “Improvement Items” section of the DevOps transformation canvas

SCALED AGILE® © Scaled Agile, Inc.

175

Notes:

7.1 Share your DevOps action plan



Taking action: DevOps transformation canvas

Duration
15 min

- ▶ **Step 1:** Review other teams' DevOps transformation canvases.
 - Note: Each team should have a person by their canvas to answer questions. Make sure the people who present the canvases rotate and get the chance to walk the gallery.
- ▶ **Step 2:** Share some observations with the class.

SCALED AGILE® © Scaled Agile, Inc.

176

Notes:

7.1 Share your DevOps action plan

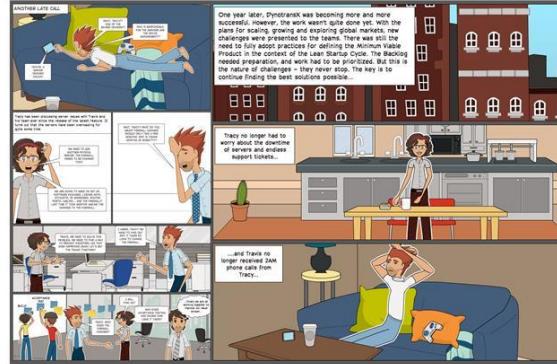


Activity: The End

Prepare
8 min

Share
2 min

- ▶ **Step 1:** Read the story, “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 DevOps transformation journey?
 - What are some of your takeaways from the story about Travis and Tracy and the DynotransX, Inc. transformation?



SCALED AGILE® © Scaled Agile, Inc.

177

Notes:

Future challenges:

Key takeaways:



7.1 Share your DevOps action plan



7.1 Share your DevOps action plan



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 with Agile Software Engineering
- ▶ Explore the architect role with SAFe for Architects



SCALED AGILE® © Scaled Agile, Inc.

178

Notes:

Lesson review

In this lesson you:

- ▶ Shared your DevOps action plan

Notes:

Lesson 7 notes



Click below to type your thoughts.

Lesson 8

Becoming a SAFe DevOps Practitioner

Learning Objectives:

8.1 Becoming a Certified SAFe Professional



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

Make the most of your learning



Access the SAFe Community Platform

Manage your member profile, continue your learning with toolkits and videos, and access communities of practice and the member directory



Prepare Yourself

Extend your SAFe knowledge and prepare for certification with your learning plan, course workbook, study materials, and practice test before your exam



Become a Certified SAFe Professional

Demonstrate your validated knowledge, skills, and mindset to participate in SAFe methods



Showcase Your SAFe Credentials

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

Notes:



Video: Become a Certified SAFe Professional

Duration
3 min

Continue to build on the foundation of SAFe learning you began in class by studying and taking the certification exam.

Earning this certification demonstrates and establishes your new knowledge.

Certification details at:

<https://www.scaledagile.com/certification/about-safe-certification/>



<https://vimeo.com/307578726>

SCALED AGILE® © Scaled Agile, Inc.

181

Notes:



Video link: <https://vimeo.com/307578726>



About SAFe certification: <https://www.scaledagile.com/certifications/about-safe-certification/>

8.1 Becoming a Certified SAFe Professional



Video: Welcome to the SAFe Community Platform

Duration
5 min

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

Access the SAFe Community Platform and discover all the SAFe resources available for your use!



<https://vimeo.com/201877314>

SALEAD AGILE® © Scaled Agile, Inc.

182

Notes:



Video link: <https://vimeo.com/201877314>

Certification Exam Sample Questions

These sample questions provide examples of the format and type of questions to expect on the exam (these are not the actual exam questions). Performance on the sample questions is NOT an indicator of the performance on the exam, and it should NOT be considered an assessment tool. A web-enabled version of the sample questions are now available in a flashcard style format (internet required). Use the link below to access the sample question bank and begin preparing for certification.

To get started:

1. Click the link below
2. A browser window will open with the sample questions site
3. Click "Start"
4. Use the left-side menu to scroll and select your course
5. Click "Start" to access the sample questions



Sample questions: <http://bit.ly/3aqP4O>

Lesson 8 notes



Click below to type your thoughts.

Appendix 1

Glossary

Glossary



SAFe Glossary:

Visit the Scaled Agile Framework site (scaledagileframework.com/glossary) to download glossaries translated into other languages