

The Principles of the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity - the art of maximizing the amount of work not done - is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: Principles behind the Agile Manifesto: <https://agilemanifesto.org/principles.html>

The Principles of the Agile Manifesto

Principles	Works as is	Not Applicable	Requires rethinking at scale
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. Business people and developers must work together daily throughout the project.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7. Working software is the primary measure of progress.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9. Continuous attention to technical excellence and good design enhances agility.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10. Simplicity—the art of maximizing the amount of work not done—is essential.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11. The best architectures, requirements, and designs emerge from self-organizing teams.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Notes

2.2 Lean and Agile at scale with the SAFe Principles

SCALED AGILE™ © Scaled Agile, Inc.

2-23

SAFe Lean-Agile Principles

#1 Take an economic view

#2 Apply systems thinking

#3 Assume variability; preserve options

#4 Build incrementally with fast, integrated learning cycles

#5 Base milestones on objective evaluation of working systems

#6 Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 Apply cadence, synchronize with cross-domain planning

#8 Unlock the intrinsic motivation of knowledge workers

#9 Decentralize decision-making

#10 Organize around value

SCALED AGILE™ © Scaled Agile, Inc.

2-24

Why focus on the principles?

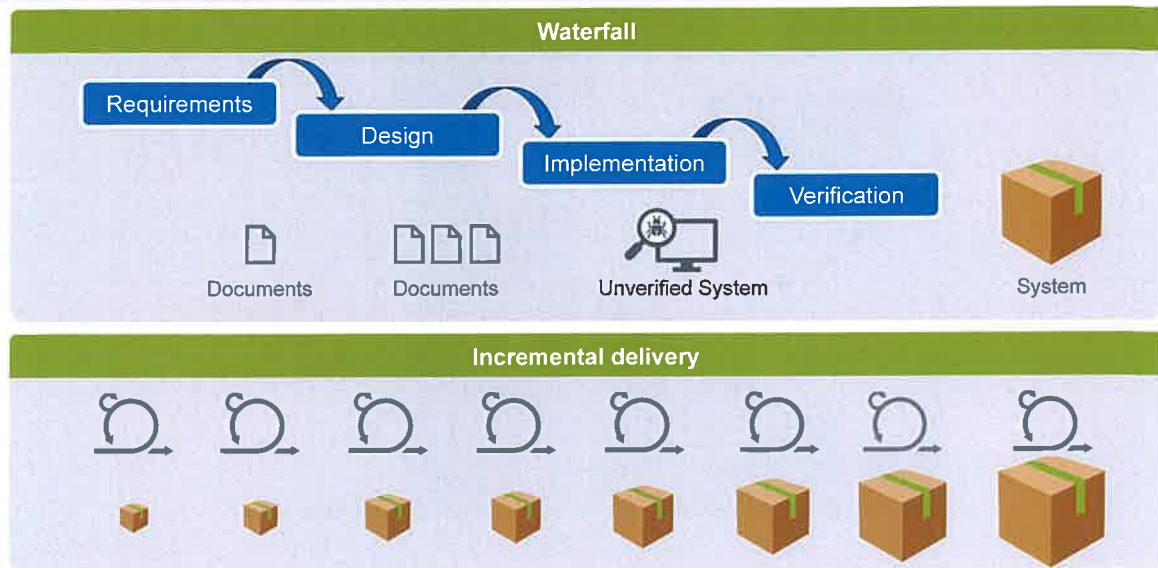
A common disease that afflicts management the world over is the impression that "Our problems are different." They are different to be sure, but the principles that will help to improve the quality of products and services are universal in nature.

—W. Edwards Deming

- ▶ A Lean-Agile transformation will deliver substantial benefits
- ▶ However, it is a significant change, and every implementation is different
- ▶ Leaders should understand why the practices work; it's part of 'knowing what it is they must do'
- ▶ If a practice needs to change, understanding the principles will assure the change moves the Enterprise in the right direction

#1 Take an economic view

Agile economics: Deliver early and often

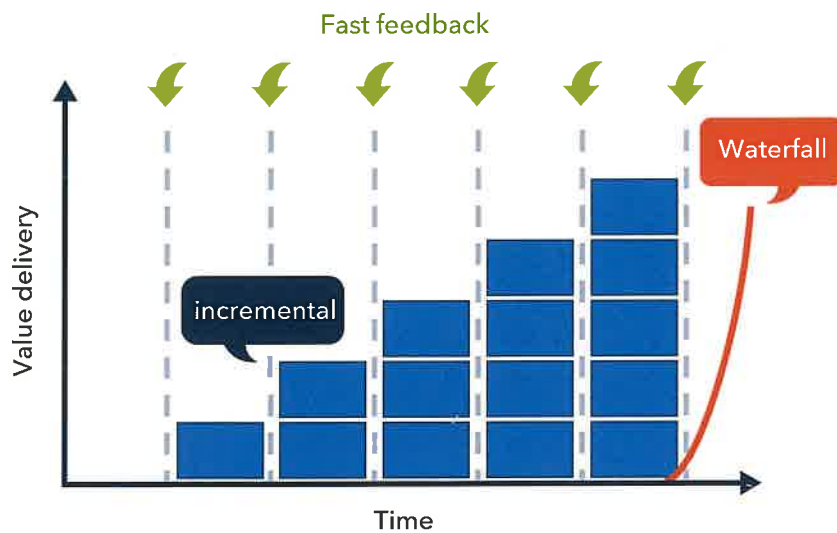


OKAYEITH!
ZPĚTNÁ
VAZBA

SCALED AGILE © Scaled Agile, Inc.

2-27

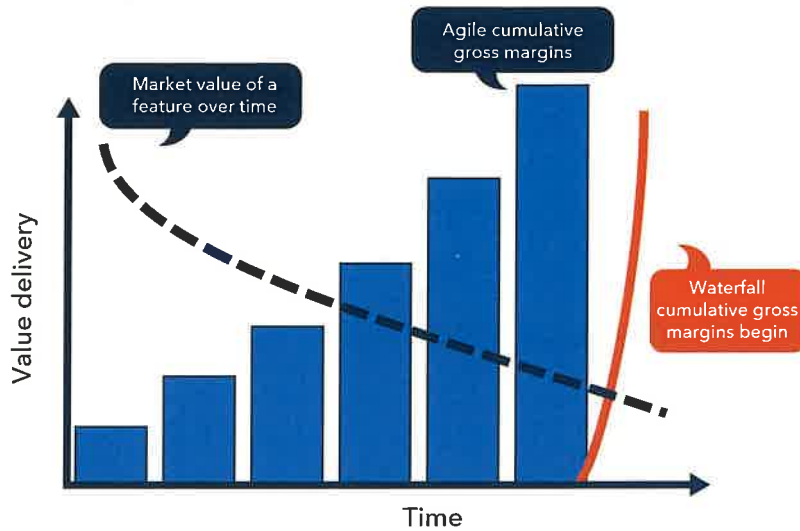
Deliver value incrementally



SCALED AGILE © Scaled Agile, Inc.

2-28

Early delivery has higher value



SCALED AGILESM © Scaled Agile, Inc.

2-29



Activity: Accelerating value delivery

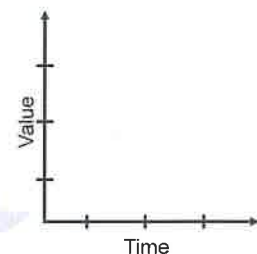


- ▶ **Step 1:** Consider that your backlog has three Features. Each will take the entire team one month to complete and delivers one unit of value.
- ▶ **Step 2:** Plot the value delivery of serial and simultaneous/parallel implementation scenarios for delivering the Features.
 - **NOTE:** Assume 20% task switching overhead for each team member in Scenario 2: Parallel delivery
 - **HINT:** Plot the serial case first

Scenario 1: Serial delivery



Scenario 2: Parallel delivery



SCALED AGILESM © Scaled Agile, Inc.

2-30

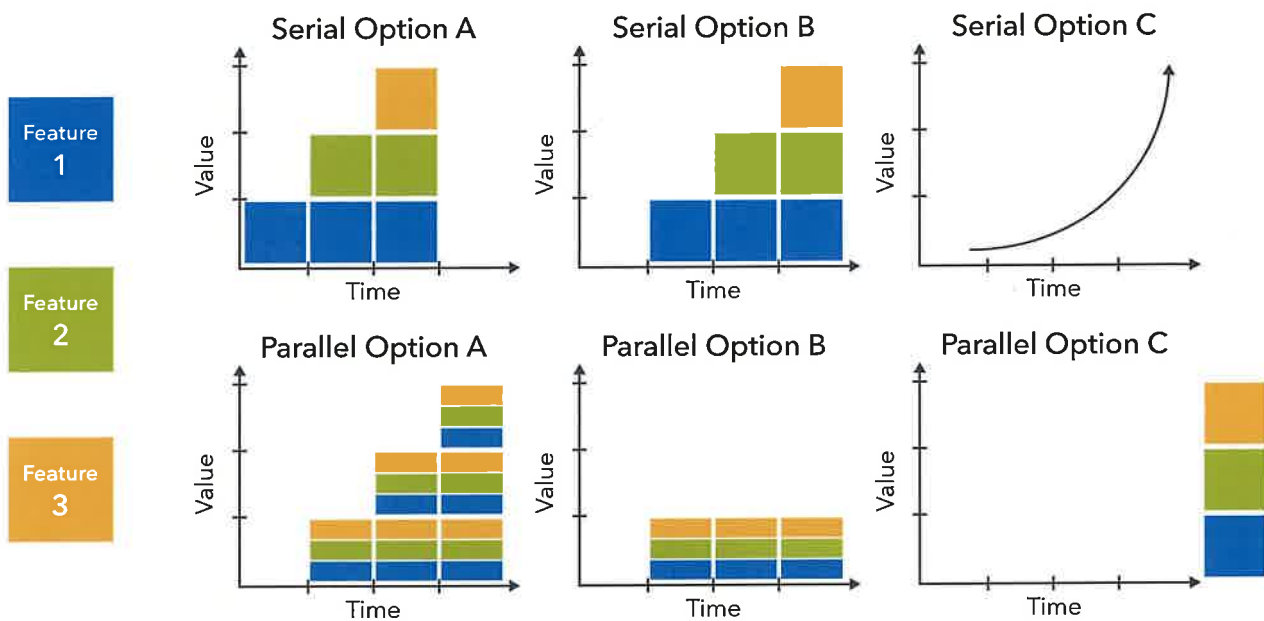
Agile Manifesto Principles at Scale

Instructions: In your groups, discuss the graphs for the serial and parallel approaches. Be prepared to discuss with the class. Consider these questions:

For the serial approach, which graph is correct?

For the parallel approach, which graph is correct?

Which approach will deliver more value?

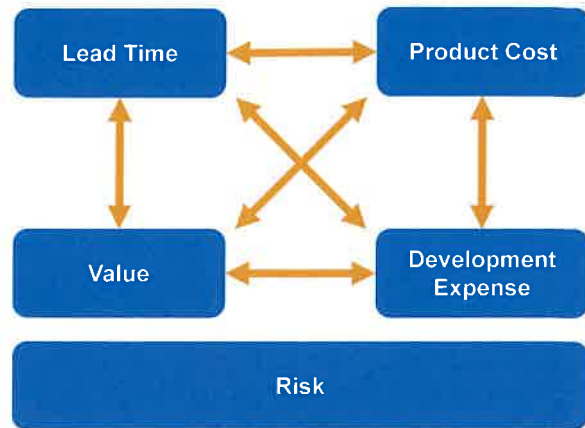


Notes

Solution economic trade-offs

Understanding trade-off parameters:

- ▶ Sequence jobs for maximum benefit
- ▶ Do not consider money already spent
- ▶ Make economic choices continuously
- ▶ Empower local decision making
- ▶ If you only quantify one thing, quantify the cost of delay



WSSS → PRIORITIZATION! FW

#2 Apply systems thinking



A system must be managed. It will not manage itself.



Left to themselves, components become selfish, independent profit centers and thus destroy the system...

The secret is cooperation between components toward the aim of the organization.

—W. Edwards Deming

2-33

Attributes of systems thinking

The Solution and the Enterprise are both affected by the following:

- ▶ Optimizing a component does not optimize the system
- ▶ For the system to behave well as a system, a higher-level understanding of behavior and architecture is required
- ▶ The value of a system passes through its interconnections
- ▶ A system can evolve no faster than its slowest integration point



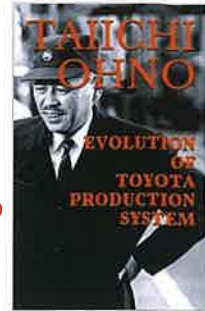
SCALED AGILE © Scaled Agile, Inc.

2-34

Optimize the full Value Stream

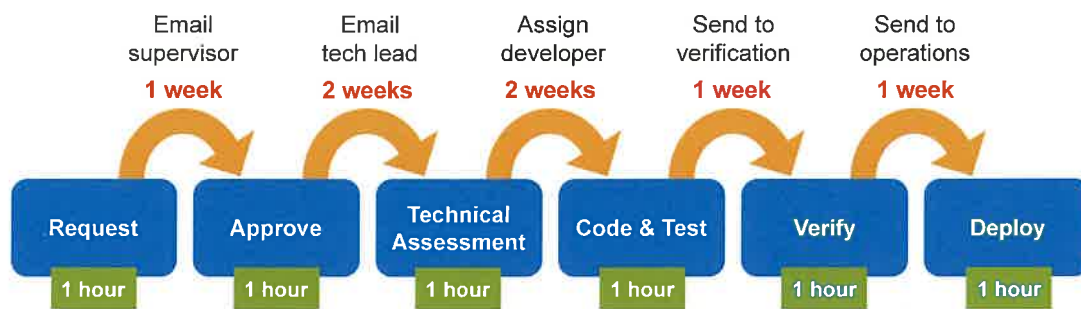
All we are doing is looking at the timeline from when the customer gives us an order to when we collect the cash. And we are reducing the timeline by reducing the non-value-added wastes.

—Taiichi Ohno



- ▶ Most problems with your process will surface as *delays*
- ▶ Most of the time spent getting to market is a result of these delays
- ▶ Reducing delays is the fastest way to reduce time to market

Focus on the delays!



6 hours of value...

...delivered in 7 weeks

$\frac{1 \text{ day}}{49 \text{ days}} = 2\% \text{ Flow Efficiency}$



Discussion: Identifying delays

Prepare



Share



- ▶ **Step 1:** Identify three delays from your context and write them down.
- ▶ **Step 2:** Write down what you think might be some potential causes for the delays.
- ▶ **Step 3:** Consider how systems thinking relates to finding possible solutions for the delays. Who is ultimately responsible for the optimization of the full Value Stream?
- ▶ **Step 4:** Share your insights with the class.

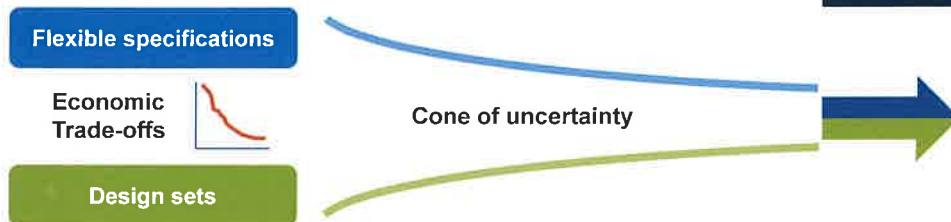
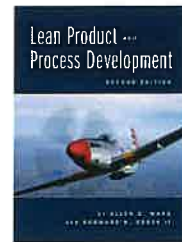
#3 Assume variability; preserve options

Development occurs in an uncertain world

Aggressively evaluate alternatives. Converge specifications and solution set.

—Allen Ward

- ▶ You cannot possibly know everything at the start
- ▶ Requirements must be flexible to make economic design choices
- ▶ Designs must be flexible to support changing requirements
- ▶ Preservation of options improves economic results



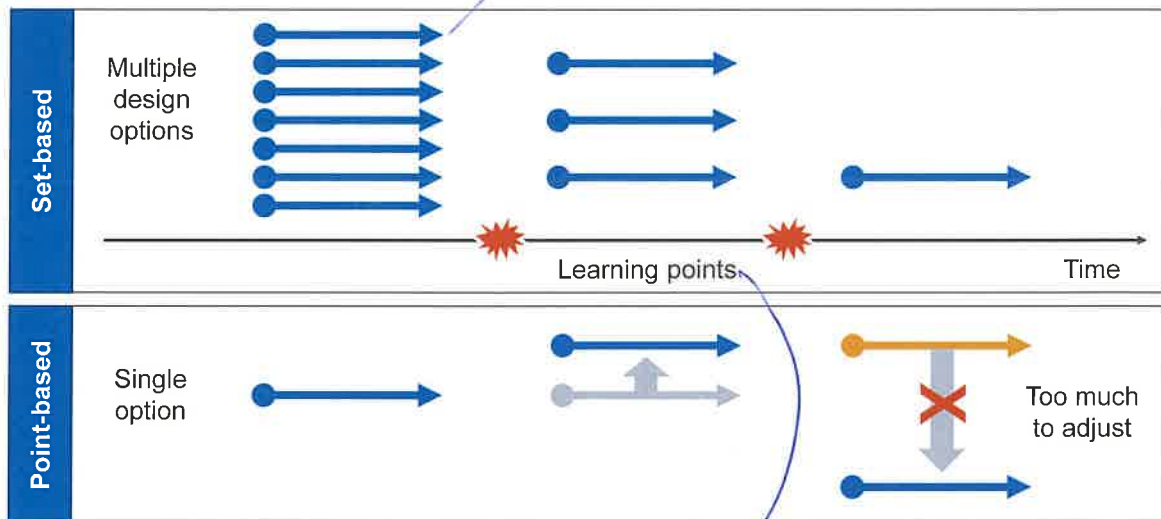
SCALED AGILE © Scaled Agile, Inc.

2-39

LEAN MACHINE BOOK

FEASIBILITY STUDIES

Apply a set-based approach



SCALED AGILE © Scaled Agile, Inc.

2-40

DEMA

#4 Build incrementally with fast, integrated learning cycles

SCALED AGILE © Scaled Agile, Inc.

2-41

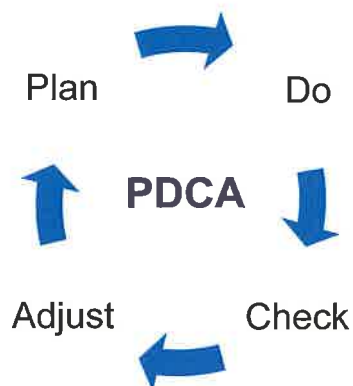
Apply fast learning cycles

Fast feedback accelerates knowledge.

- ▶ Improves learning efficiency by decreasing the time between action and effect
- ▶ Reduces the cost of risk-taking by truncating unsuccessful paths quickly
- ▶ Is facilitated by small batch sizes
- ▶ Requires increased investment in development environment

The shorter the cycles, the faster the learning.

The iterative learning cycle

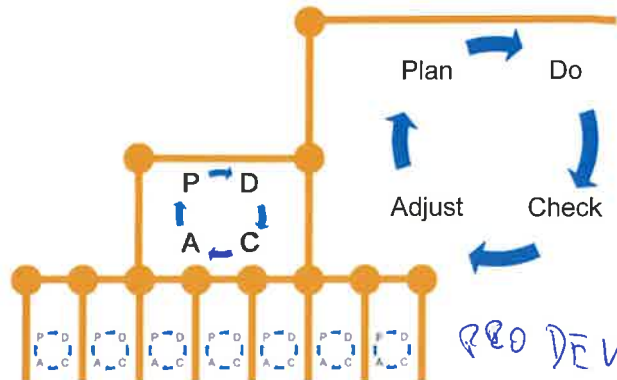


SCALED AGILE © Scaled Agile, Inc.

2-42

Integration points control product development

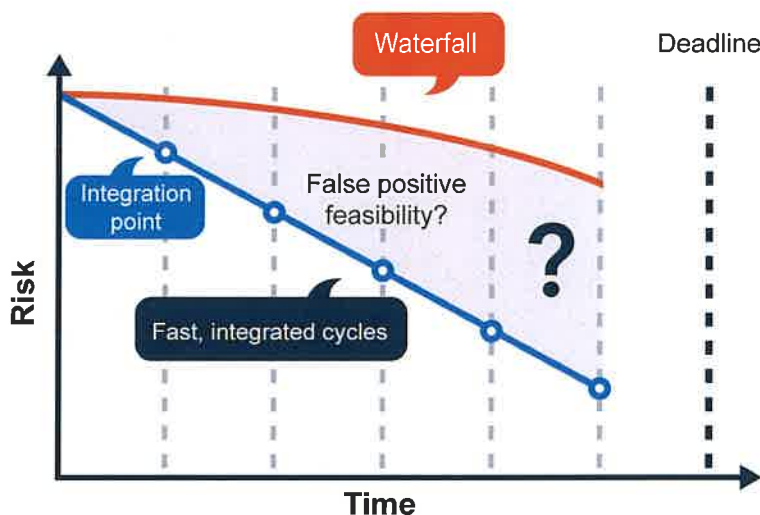
- ▶ Integration points accelerate learning
- ▶ Development can proceed no faster than the slowest learning loop
- ▶ Improvement comes through synchronization of design loops and faster learning cycles



SCALED AGILE © Scaled Agile, Inc.

2-43

Integration points reduce risk



SCALED AGILE © Scaled Agile, Inc.

2-44

#5 Base milestones on objective evaluation of working systems

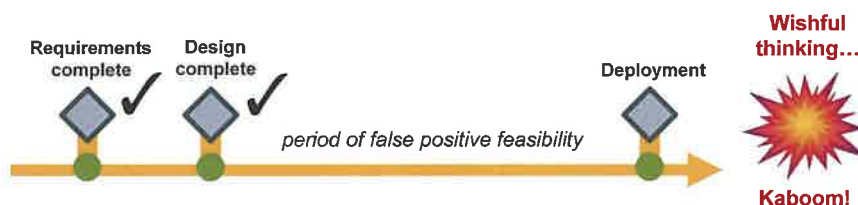
SCALED AGILE © Scaled Agile, Inc.

2-45

The problem of phase-gate milestones

There was in fact no correlation between exiting phase gates on time and project success... the data suggested the inverse might be true. —Dantar Oosterwal, Lean Machine

- ▶ They force design decisions too early; this encourages false-positive feasibility.
- ▶ They assume a 'point' Solution exists and can be built correctly the first time.
- ▶ They create huge batches and long queues, and they centralize requirements and design in program management.

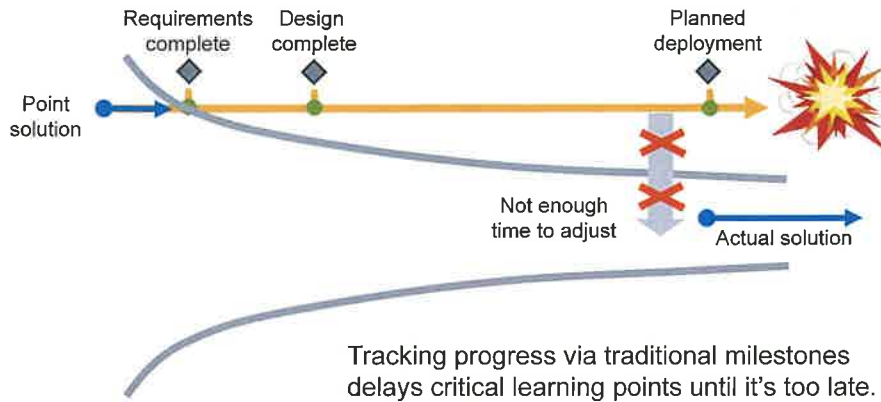


SCALED AGILE © Scaled Agile, Inc.

2-46

The problem of phase-gate milestones

Phase gates fix requirements and designs too early, making adjustments too late and costly as new facts emerge.

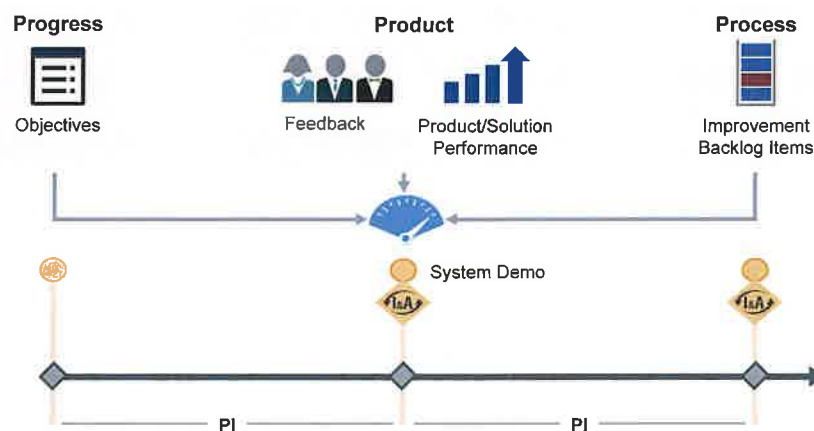


SCALED AGILE © Scaled Agile, Inc.

2-47

Apply objective Milestones

Program Increment (PI) System Demos are orchestrated to deliver objective progress, product, and process Metrics.

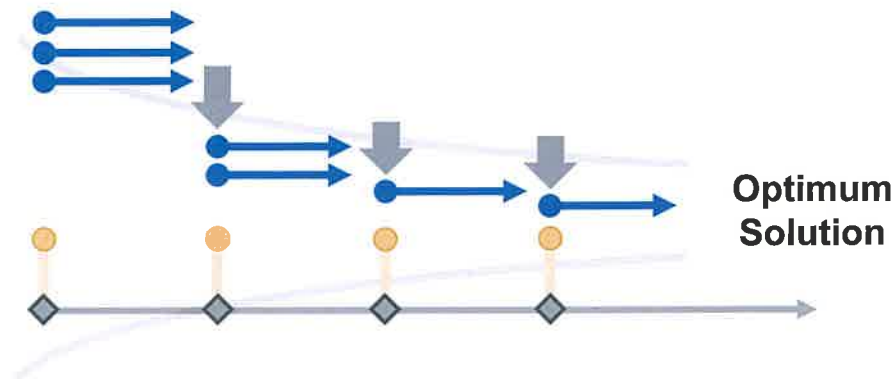


SCALED AGILE © Scaled Agile, Inc.

2-48

Iterate to the optimum Solution

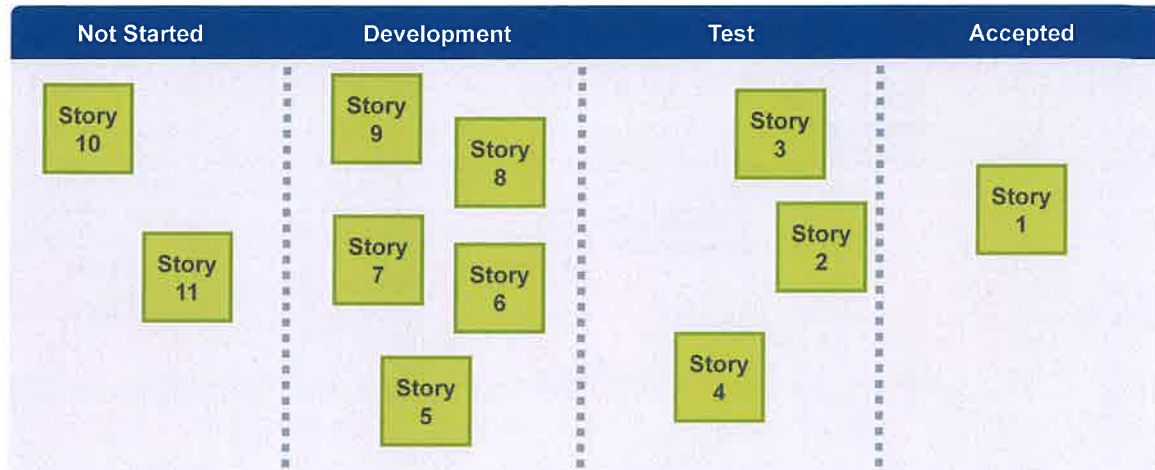
Objective Milestones facilitate learning and allow for continuous, cost-effective adjustments towards an optimum Solution.



#6 Visualize and limit WIP, reduce batch sizes, and manage queue lengths

An example from the field

How is this team doing? How do you know that?



SCALED AGILE © Scaled Agile, Inc.

2-51

KANBAN BOARD

Visualize to increase understanding

Now how do you think they are doing?



SCALED AGILE © Scaled Agile, Inc.

2-52

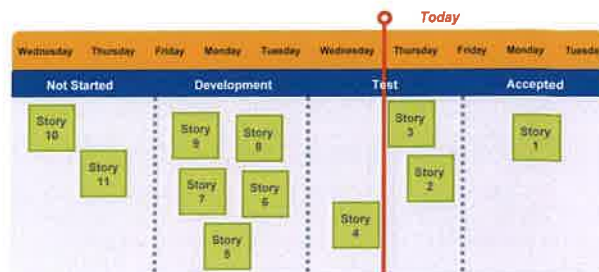
WORKING PROGRESS LIMIT = PROPOST MOST



Activity: WIP improvement opportunities



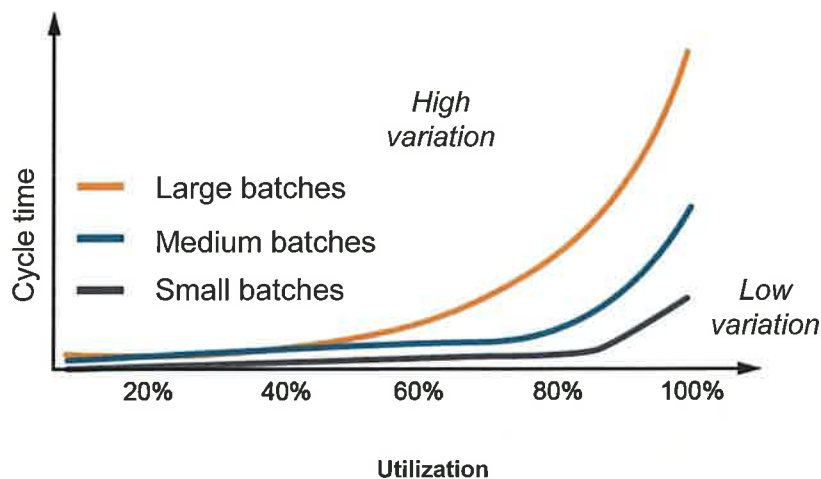
- **Step 1:** Referring to the *Team Board* example, discuss the effect of a three-story WIP constraint on Development and Test.
- **Step 2:** Consider this scenario: You're a developer. You just finished Story 6. What would you do if:
 - There is no WIP constraint
 - The three-Story WIP constraint is in place
- **Step 3:** Which scenario has the highest throughput?



SCALED AGILE © Scaled Agile, Inc.

2-53

Reduce batch size for higher predictability



Source: *Implementing Lean Software Development*, Poppendieck, Mary

SCALED AGILE © Scaled Agile, Inc.

2-54

STILENA BOLEST JE POLOVITELI :)



Activity: Experience a large batch size



- **Step 1:** Create groups of five people with 10 coins per group. Designate one person as the timekeeper. The remaining four people will be processing the coins.
- **Step 2:** Person by person process each coin.
- **Step 3:** Pass all coins at the same time to the next person, who repeats step two until all four people are done
- **Step 4:** The timekeeper stops the timer and records the total time



<https://bit.ly/Video-LargeBatchPart1>

Optional 1:20 video demonstrating this exercise: part 1 of 3

SCALED AGILE® © Scaled Agile, Inc.

2-55



Activity: Experience a small batch size



- **Step 1:** Ensure that the timekeeper is ready to start the timer
- **Step 2:** This time, each person processes one coin at a time and immediately passes each coin to the next person
- **Step 3:** The timekeeper will stop the timer when the last person flips the last coin and records the result



Optional 18 sec video demonstrating this exercise: part 2 of 3

<https://bit.ly/Video-SmallBatchPart2>



Optional 19 sec video demonstrating this exercise: part 3 of 3

<https://bit.ly/Video-BatchOverviewPart3>

SCALED AGILE® © Scaled Agile, Inc.

2-56

The importance of small batches

- ▶ Large batch sizes increase variability
 - ▶ High utilization increases variability
 - ▶ **Severe project slippage is the most likely result**
- ➔
- ▶ Small batches go through the system faster with lower variability
 - ▶ The most important batch is the handoff batch



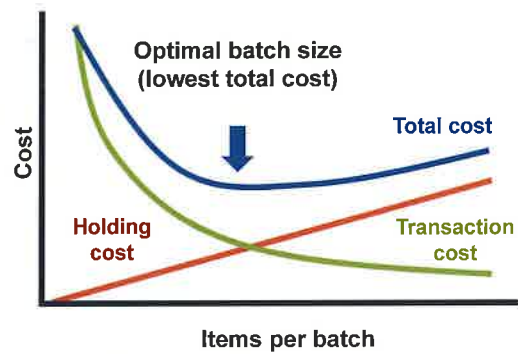
SCALED AGILE © Scaled Agile, Inc.

2-57

Finding optimal batch size

Optimal batch size is an example of a U-curve optimization.

- ▶ Total costs are the sum of holding costs and transaction costs
- ▶ Higher transaction costs make optimal batch size bigger
- ▶ Higher holding costs make optimal batch size smaller



Principles of Product Development Flow, Don Reinertsen

SCALED AGILE © Scaled Agile, Inc.

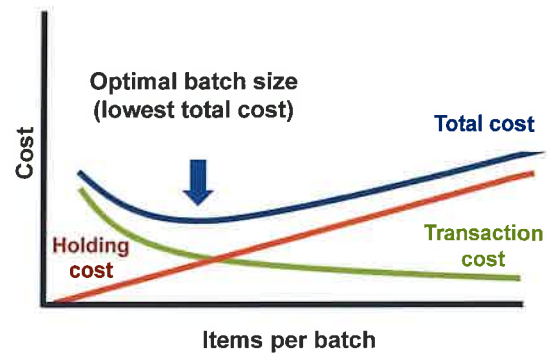
2-58

SAFE DE ROLL SYSTEM

Reducing optimal batch size ?

Reducing transaction costs reduces total costs and lowers optimal batch size.

- ▶ Reducing batch size:
 - Increases predictability
 - Accelerates feedback
 - Reduces rework
 - Lowers cost
- ▶ Batch size reduction probably saves **twice** what you would think



Principles of Product Development Flow, Don Reinertsen



Video: Formula 1 Pit Stops: 1950 and Today



<https://bit.ly/Video-Formula1PitStops>

Manage queue lengths

Email from a client service organization:

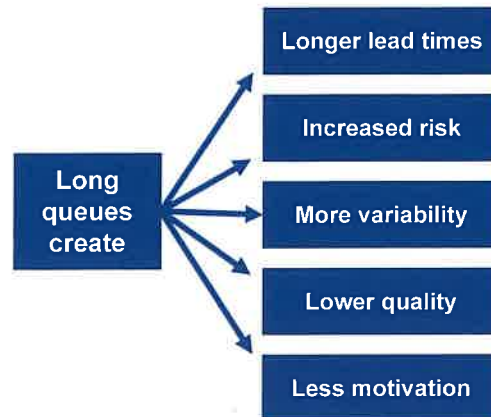
Thank you for contacting us.



We are experiencing increased volumes and apologize in advance for the delay.

Our goal is to contact you within...

Long queues: All bad



Principles of Product Development Flow, Don Reinertsen

Reduce queue lengths

- Understand Little's Law
 - Faster processing time decreases wait
 - Shorter queue lengths decrease wait
- Control wait times by controlling queue lengths:
 - WIP limits, small batches, defer commitments

$$W_q = \frac{L_q}{\lambda}$$

Average wait time = average queue length divided by average processing rate

Example – Given an average processing speed of 10 Features per quarter and a committed set of 30 Features, a new Feature will experience an approximate wait time of:

$$\frac{30 \text{ items}}{10 \text{ items/Quarter}} = 3 \text{ Quarters}$$

#7 Apply cadence, synchronize with cross-domain planning

SCALED AGILE © Scaled Agile, Inc.

2-63

Cadence and synchronization

Cadence

- ▶ Converts unpredictable events into predictable occurrences and lowers cost
- ▶ Makes waiting times for new work predictable
- ▶ Supports regular planning and cross-functional coordination
- ▶ Limits batch sizes to a single interval
- ▶ Controls injection of new work
- ▶ Provides scheduled integration points

Note: Delivering on cadence requires scope or capacity margin

SCALED AGILE © Scaled Agile, Inc.

Synchronization

- ▶ Causes multiple events to happen simultaneously
- ▶ Facilitates cross-functional trade-offs
- ▶ Provides routine dependency management
- ▶ Supports full system integration and assessment
- ▶ Provides multiple feedback perspectives

Note: To work effectively, design cycles must be synchronized

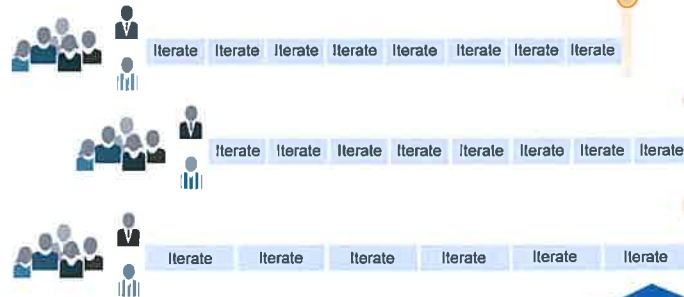
2-64

Cadence without synchronization is not enough

These teams
are iterating

Time spent thinking you are on track

When you discover you are not



Integrate
and slip!

The slowest component drags the train – still late discovery!

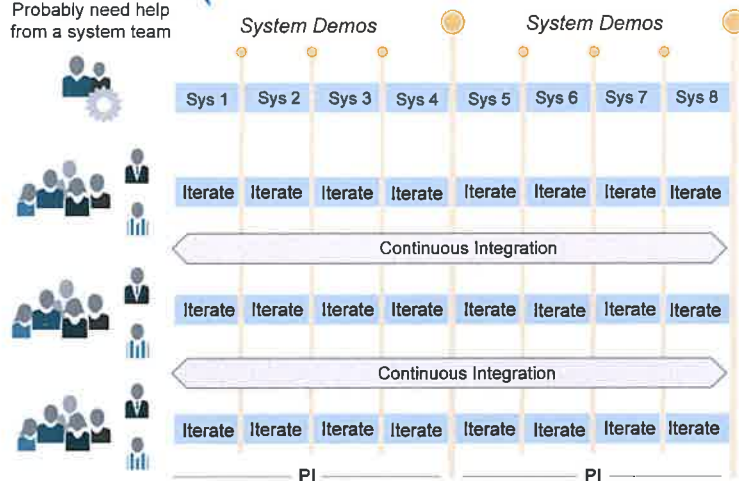
SCALED AGILE® © Scaled Agile, Inc.

2-65

Synchronize to assure delivery

This system is iterating

Probably need help
from a system team

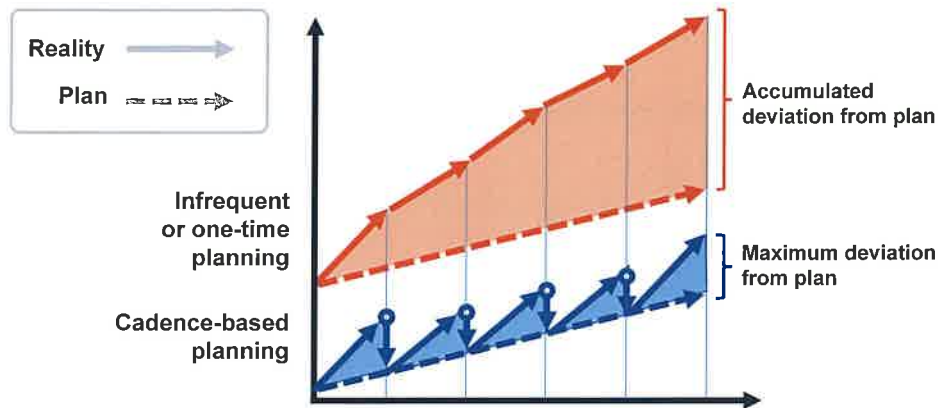


SCALED AGILE® © Scaled Agile, Inc.

2-66

Control variability with planning cadence

Cadence-based planning limits variability to a single interval.



SCALED AGILE © Scaled Agile, Inc.

2-67

Synchronize with cross-domain planning

Future product development tasks can't be predetermined. Distribute planning and control to those who can understand and react to the end results.

—Michael Kennedy, Product Development for the Lean Enterprise

- ▶ Everyone plans together at the same time
- ▶ Management sets the mission with minimum constraints
- ▶ Requirements and design emerge
- ▶ Important decisions are accelerated
- ▶ Teams take responsibility for their own plans



SCALED AGILE © Scaled Agile, Inc.

2-68

#8 Unlock the intrinsic motivation of knowledge workers

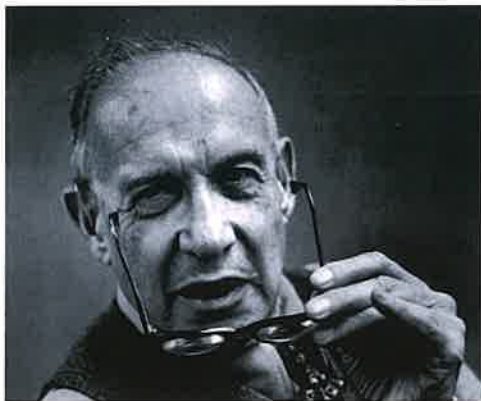
SCALED AGILE © Scaled Agile, Inc.

2-69

On managing knowledge workers

Workers are knowledge workers if they know more about the work they perform than their bosses.

—Peter Drucker



Used with permission from The Drucker Institute at Claremont Graduate University

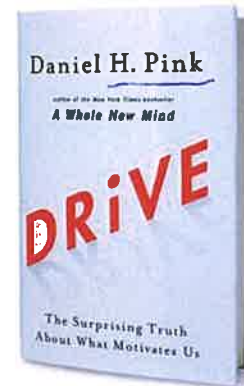
- ▶ Workers themselves are most qualified to make decisions about how to perform their work.
- ▶ The workers must be heard and respected for management to lead effectively.
- ▶ Knowledge workers must manage themselves. They need autonomy.
- ▶ Continuing innovation must be part of the work, the tasks, and the responsibilities of knowledge workers.

SCALED AGILE © Scaled Agile, Inc.

2-70

Unlocking intrinsic motivation with autonomy, mastery, and purpose

- ▶ **Autonomy** is the desire to be self-directed and have control over what we work on, how we do our work, and who we work with
- ▶ **Mastery** is the urge to get better at what we do and improve our personal and team skills
- ▶ **Purpose** is the desire to do something that matters and has meaning



#9 Decentralize decision-making



Video: Greatness by David Marquet

Duration



<https://bit.ly/Video-GreatnessMarquet>

SCALED AGILE © Scaled Agile, Inc.

2-73

Decentralize decision-making

Define the economic logic behind a decision; empower others to make the changes.

Centralize

- ▶ **Infrequent** – Not made very often and usually not urgent
(**Example:** Internationalization strategy)
- ▶ **Long-lasting** – Once made, highly unlikely to change
(**Example:** Common technology platform)
- ▶ **Significant economies of scale** – Provide large and broad economic benefit
(**Example:** Compensation strategy)

Decentralize everything else

- ▶ **Frequent** – Routine, everyday decisions
(**Example:** Team and Program Backlog)
- ▶ **Time critical** – High cost of delay
(**Example:** Point release to Customer)
- ▶ **Requires local information** – Specific and local technology or Customer context is required
(**Example:** Feature criteria)

SCALED AGILE © Scaled Agile, Inc.

2-74

ECONOMY OF SCALE - VÝHODA Z ROZSAHU



Activity: Decentralize decision-making

Prepare



Share



- ▶ **Step 1:** Consider three significant decisions you are currently facing. Write them in the table provided in your workbook.
- ▶ **Step 2:** Rate each decision based on the frequency, time criticality, and economies of scale, assigning a value of 0, 1, or 2.
- ▶ **Step 3:** Add the total values: 0 – 3 centralize and 4 – 6 decentralize.

Decision	Frequent? Y=2 N=0	Time-critical? Y=2 N=0	Economies of scale? Y=0 N=2	Total

Keys to practicing decentralized decision-making

- ▶ Openly discuss how decisions are made and explore opportunities to move authority for those decisions closer to where the work is performed.
- ▶ Establish a decision-making framework that equips knowledge workers with the information to make good decisions.
- ▶ Provide clarity on organizational objectives, coach effective problem-solving, and provide opportunities to exercise and cultivate decision-making abilities.
- ▶ Take responsibility for making and communicating strategic decisions—those that are infrequent, long lasting, and have significant economies of scale. Decentralize all other decisions.

Decentralize Decision-Making

Decision	Frequent? Y=2 N=0	Time- critical? Y=2 N=0	Economies of scale? Y=0 N=2	Total
				0
				0
				0

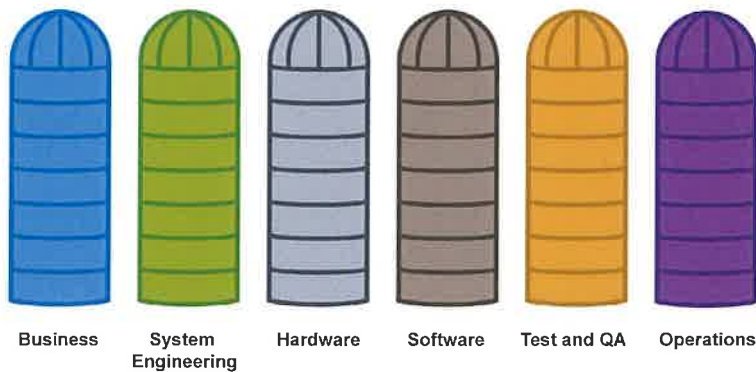
Notes

#10 Organize around value

SCALED AGILE © Scaled Agile, Inc.

2-77

Value doesn't follow silos



Management challenge: Connect the silos

- ▶ Value delivery is inhibited by handoffs and delays
- ▶ Political boundaries can prevent cooperation
- ▶ Silos encourage geographic distribution of functions
- ▶ Communication across silos is difficult

SCALED AGILE © Scaled Agile, Inc.

2-78

Instead, organize around Development Value Streams

The aim of development is in fact the creation of profitable operational value streams.

—Allen C. Ward

- ▶ Includes activities from recognizing an opportunity through release and validation
- ▶ Contains the steps, the flow of information and material, and the people who develop the Solutions used by the Operational Value Streams

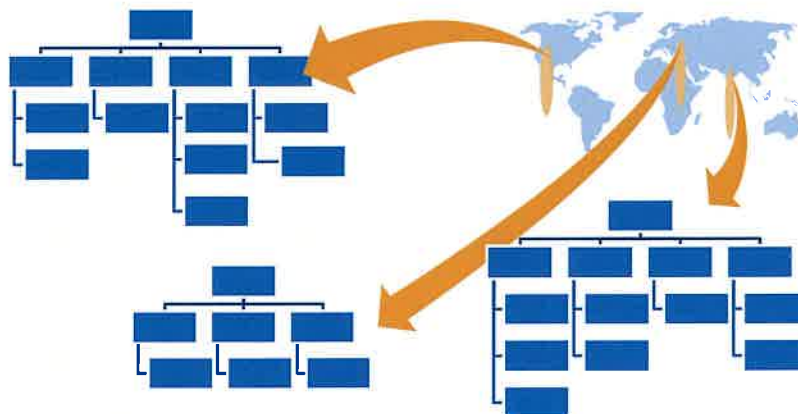


SCALED AGILE® © Scaled Agile, Inc.

2-79

2 TYPES VALUE STREAMS — OPERATIONAL
↳ DEV

Value at scale is distributed

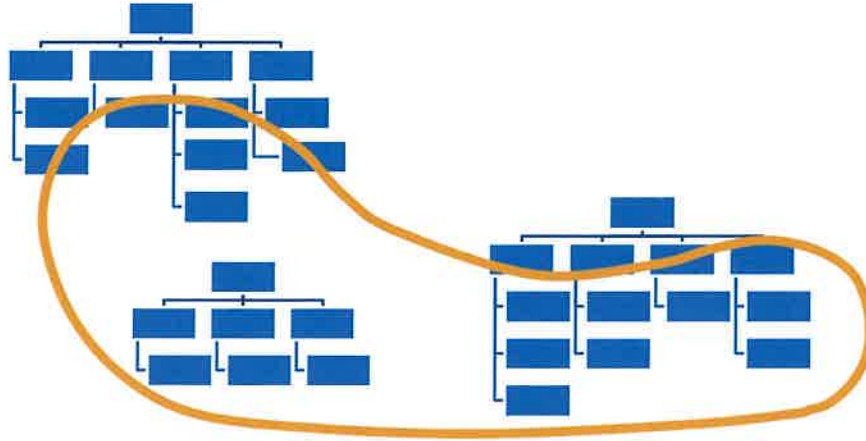


SCALED AGILE® © Scaled Agile, Inc.

2-80

Value flows across organizational boundaries

Identify the Value Streams within which to build one or more Agile Release Trains.



SCALED AGILE[®] © Scaled Agile, Inc.

2-81

Principles are great, but ...



*Clarity on how to **think**, without clarity on how to **act**, leaves people unmoved.*

—Daniel Pink

... it's time to put this thinking to work.

Let's start doing.

2-82



Action Plan: Advocating for SAFe Principles

Prepare



Share



- ▶ **Step 1:** Individually identify three actions you can take to model and advocate SAFe Principles in your Enterprise.
- ▶ **Step 2:** Write them down in your Action Plan.
- ▶ **Step 3:** In your group, share some of the insights you gained from SAFe Principles.



SCALED AGILE® © Scaled Agile, Inc.

2-83

Lesson review

In this lesson you:

- ▶ Explored the Lean-Agile Mindset
- ▶ Applied Lean and Agile at scale with the SAFe Principles

SCALED AGILE® © Scaled Agile, Inc.

2-84



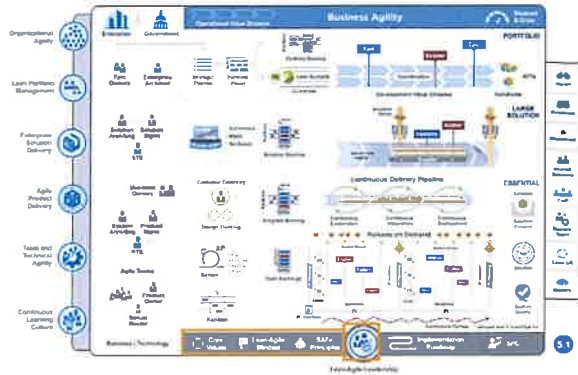
Action Plan

Advocating for SAFe Principles

Articles used in this lesson

Read these Framework articles to learn more about topics covered in this lesson

- ▶ “Core Values”
<https://www.scaledagileframework.com/safe-core-values/>
- ▶ “Lean-Agile Mindset”
<https://www.scaledagileframework.com/lean-agile-mindset/>
- ▶ “SAFe Principles”
<https://www.scaledagileframework.com/safe-lean-agile-principles/>
- ▶ “Lean-Agile Leadership”
<https://www.scaledagileframework.com/lean-agile-leadership/>



SCALED AGILE © Scaled Agile, Inc.

2-85

Continue your SAFe journey with the following resources

Review the *SAFe Core Values*
E-learning:
<https://bit.ly/Community-GettingStarted>

Review the *Lean-Agile Mindset*
E-Learning:
<https://bit.ly/Community-GettingStarted>

Review the *SAFe Lean-Agile Principles* E-Learning:
<https://bit.ly/Community-GettingStarted>

Watch this one-minute video, *How Batch Size Affects Delivery Speed*, which demonstrates how smaller batches enable faster delivery:
<https://bit.ly/Video-BatchandDeliverySpeed>

SCALED AGILE © Scaled Agile, Inc.

2-86

Lesson notes

Enter your notes below. If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 3

Establishing Team and Technical Agility

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



SCALED AGILE®

Why Team and Technical Agility

Agile Teams and teams of Agile Teams create and support the business Solutions that deliver value to the Enterprise's Customers. Consequently, an organization's ability to thrive in the digital age is entirely dependent on the ability of its teams to deliver Solutions that reliably meet a Customer's needs.



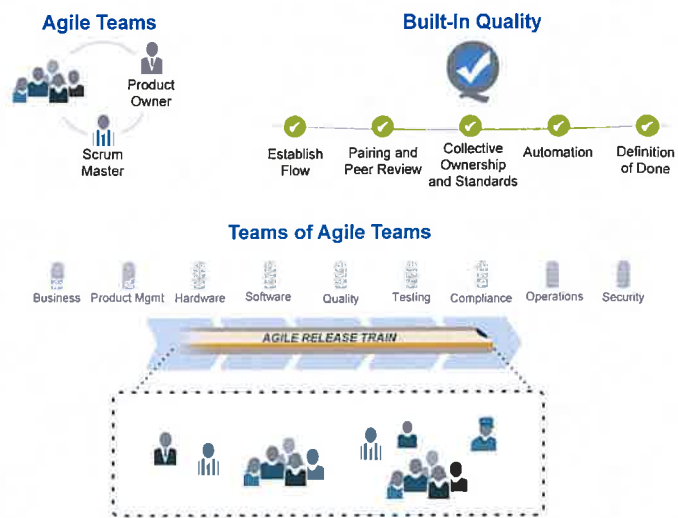
3-2

Lesson Topics

3.1 Forming cross-functional Agile Teams

3.2 Built-In Quality

3.3 Organizing Agile Release Trains around the flow of value



33

Learning objectives

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

- ▶ Prepare to form cross-functional Agile Teams
- ▶ Describe Built-in Quality practices
- ▶ Recommend organizing around value with Agile Release Trains (ARTs)

3.1 Forming cross-functional Agile Teams

SCALED AGILE © Scaled Agile, Inc.

3-6

Build cross-functional Agile Teams

Agile Teams are cross-functional, self-organizing entities that can define, build, test, and where applicable, deploy increments of value.

- ▶ Optimized for communication and delivery of value
- ▶ Deliver value every two weeks
- ▶ Contain two specialty roles:
 - Scrum Master
 - Product Owner

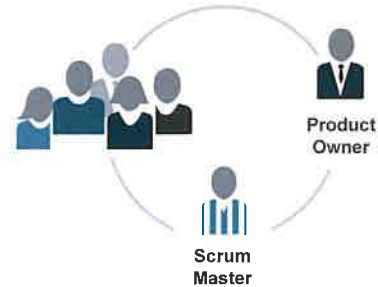


SCALED AGILE © Scaled Agile, Inc.

3-6

Responsibilities of the Agile Team

- 5 - 11
- ▶ Five to eleven team members
 - ▶ Create and refine Stories and acceptance criteria
 - ▶ Define, build, test and deploy Stories
 - ▶ Build quality in to each increment of the solution.
 - ▶ Develop and commit to team PI Objectives and Iteration Goals



Agile Teams have two speciality roles



Scrum Master

- Coaches the Agile Team in self-management
- Helps the team focus on creating increments of value each Iteration
- Facilitates the removal of impediments to the team's progress
- Ensures that all team events take place, are productive and kept within the timebox

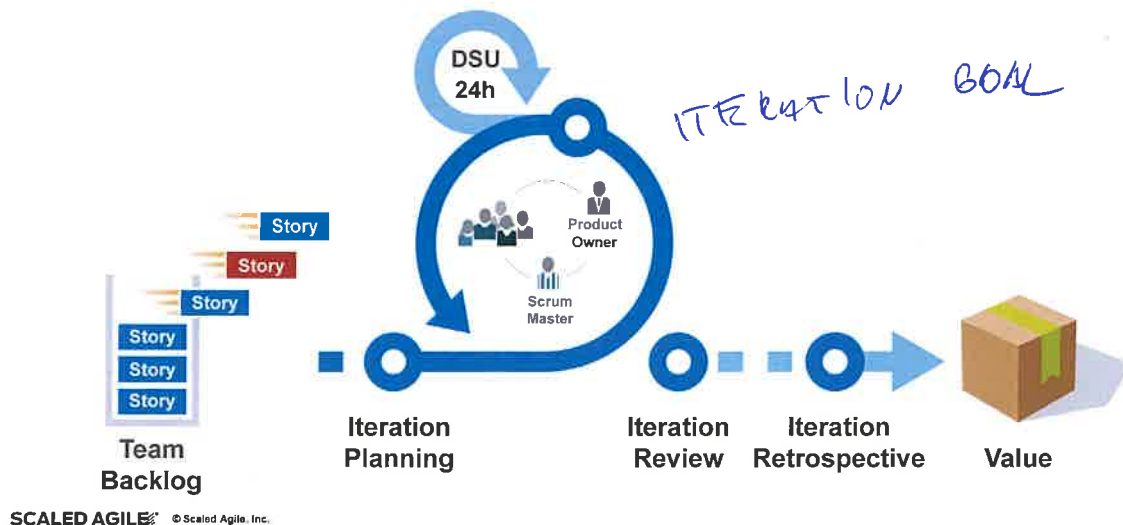


Product Owner

- Contributes to the Vision and Roadmap
- Acts as the Customer for team questions
- Creates, clearly communicates and accepts Stories
- Prioritizes the Team Backlog

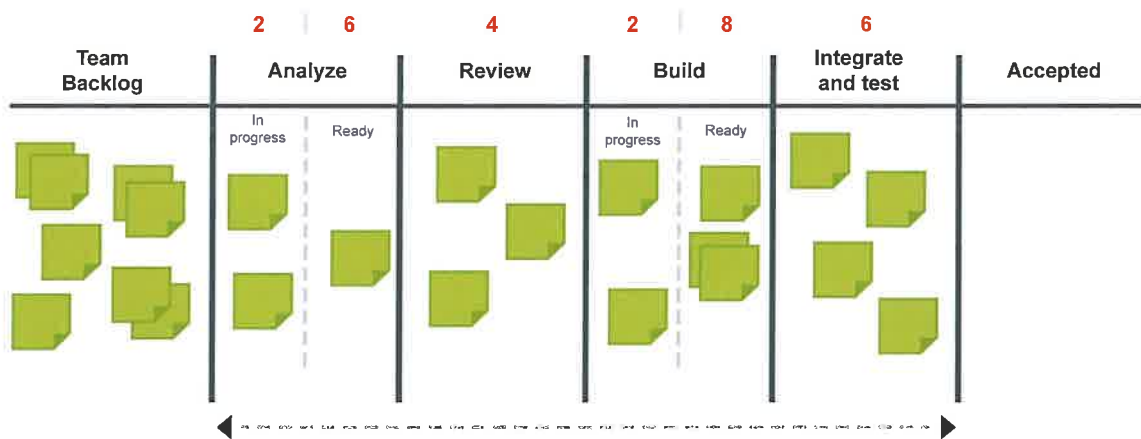
Teams execute Iterations with Scrum

Scrum is built on transparency, inspection, adaptation, and short learning cycles.



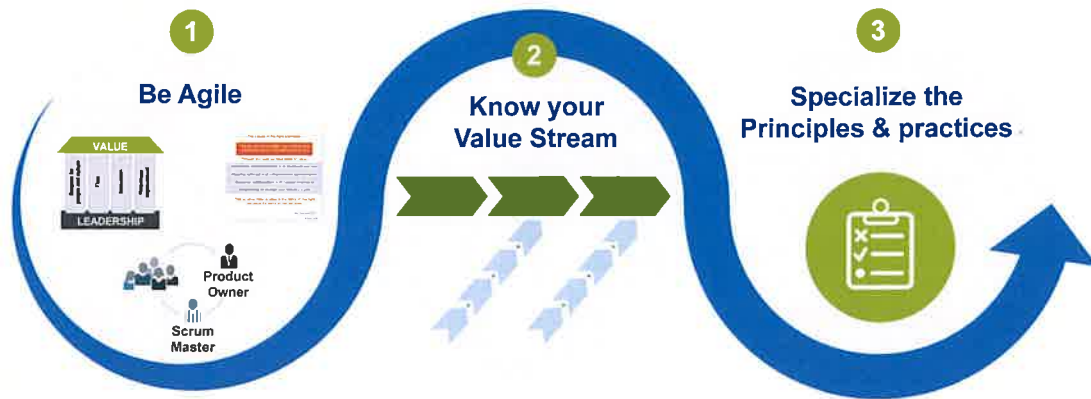
Teams visualize flow with Kanban

Kanban visualizes and optimizes the flow of work through the system.



Average WIP and duration are measured from the point work is pulled from the backlog until it is accepted.

Extend into the business with Agile business teams



Agile Team maturity cycle

<https://www.scaledagileframework.com/business-and-technology/>

SCALED AGILE® © Scaled Agile, Inc.

3-11



Activity: Identify team names and roles

Duration



- ▶ **Step 1:** Your team is your group. Create a team name
- ▶ **Step 2:** Select a Scrum Master for your team
- ▶ **Step 3:** Select a Product Owner for your team
- ▶ **Step 4:** Make sure the team name and the names of the people selected are visible to all other teams
 - **Note:** In the next lesson, your team will experience PI Planning



SCALED AGILE® © Scaled Agile, Inc.

3-12

3.2 Built-in Quality

SCALED AGILE © Scaled Agile, Inc.

3-13

Build quality in

You can't scale crappy code (or hardware, or anything else). //
– Dean Leffingwell

- ▶ Ensures that every increment of the Solution reflects quality standards
- ▶ Is required for high, sustainable development velocity
- ▶ Agile quality practices apply to every team, whether business or technology:
 - Establish flow *KANBAN*
 - Peer review and pairing
 - Collective ownership and standards
 - Automation
 - Definition of done



SCALED AGILE © Scaled Agile, Inc.

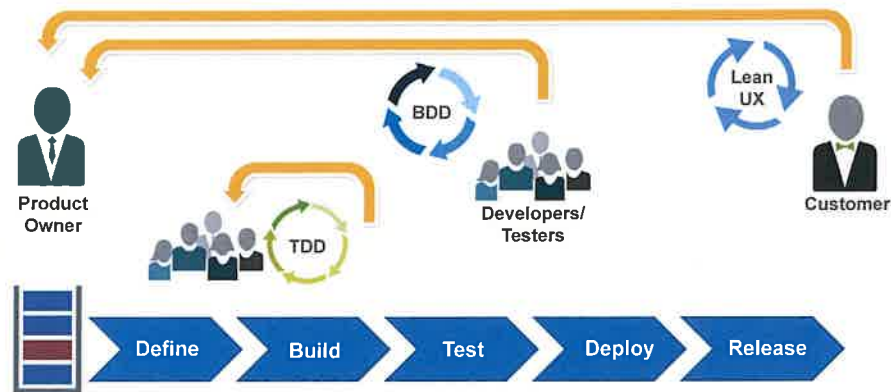
3-14

FEATURE
FLAGS

TIME

Built-in Quality practices for software teams

Include software quality practices (most inspired by XP) like, Agile testing, behavior-driven development, test-driven development, refactoring, code quality, and Agile architecture.



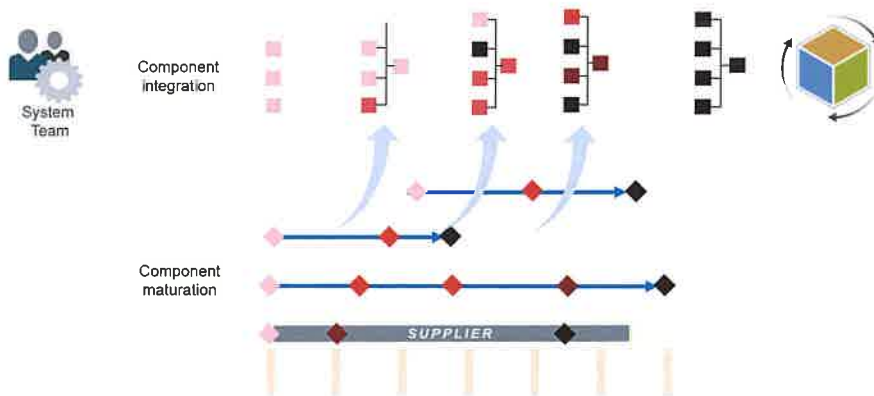
SCALED AGILE © Scaled Agile, Inc.

3-15

TEST DRIVEN DEV. BDD

Built-in Quality practices for hardware teams

Support hardware quality with exploratory, early iterations, frequent system-level integration, design verification, Model-Based Systems Engineering (MBSE), and set-based design.



SCALED AGILE © Scaled Agile, Inc.

3-16

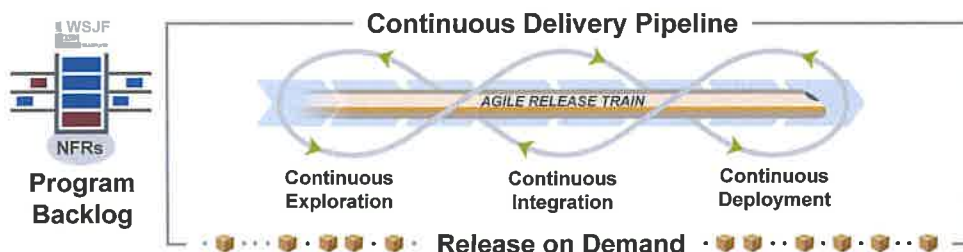
3.3 Organizing Agile Release Trains around the flow of value

SCALED AGILE © Scaled Agile, Inc.

3-17

Agile Release Trains (ARTs)

- ▶ A virtual organization of 5 – 12 teams (50 – 125+ individuals)
- ▶ Synchronized on a common cadence, a Program Increment (PI)
- ▶ Aligned to a common mission via a single Program Backlog



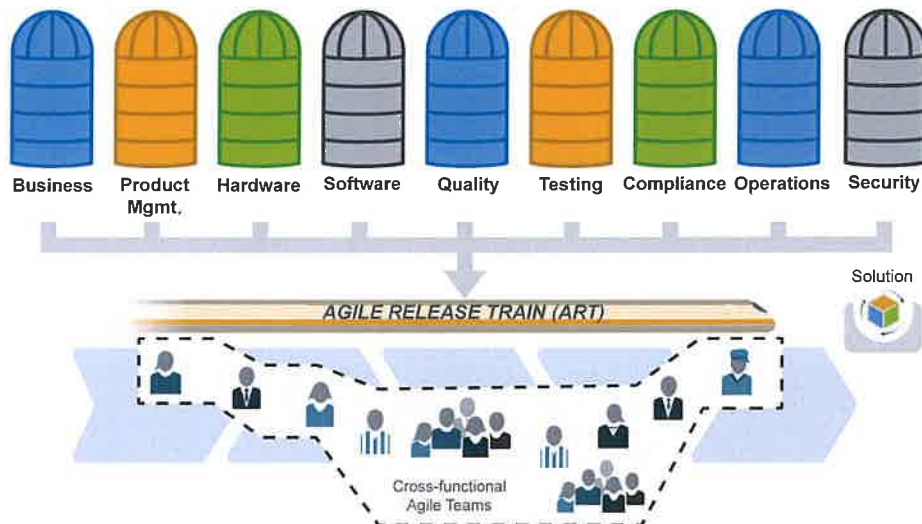
SCALED AGILE © Scaled Agile, Inc.



PRODUCT
MANAGER

3-18





ARTs are cross-functional



SCALED AGILE® © Scaled Agile, Inc.

3-19

Teams on the ART are organized for flow

-  **Stream-aligned team** – organized around the flow of work and has the ability to deliver value directly to the Customer or end user.
-  **Complicated subsystem team** – organized around specific subsystems that require deep specialty skills and expertise.
-  **Platform team** – organized around the development and support of platforms that provide services to other teams.
-  **Enabling team** – organized to assist other teams with specialized capabilities and help them become proficient in new technologies.



More information in the Advanced Topic Article:

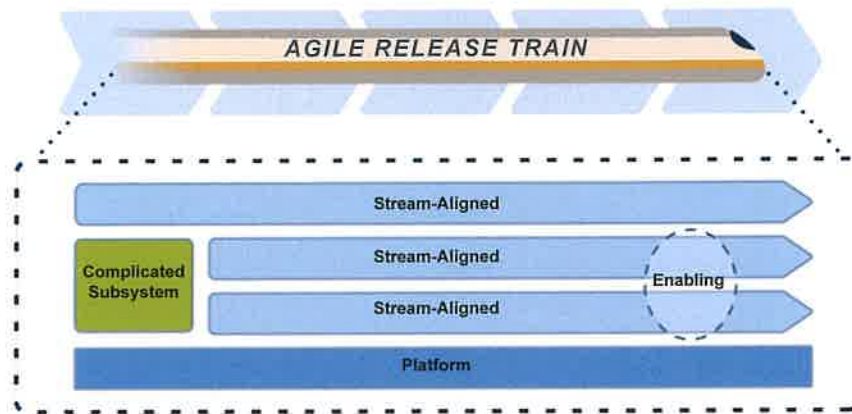
<https://www.scaledagileframework.com/organizing-agile-teams-and-arts-team-topologies-at-scale/>

SCALED AGILE® © Scaled Agile, Inc.

3-20

ARTs are organized to deliver value continuously

Consider the necessary interactions between the teams and organize to maximize flow.



SCALED AGILE © Scaled Agile, Inc.

3-21

Roles on the Agile Release Train



Release Train Engineer acts as the Chief Scrum Master for the train.



Product Management owns, defines, and prioritizes the Program Backlog.



System Architect/Engineering provides architectural guidance and technical enablement to the teams on the train.



The **System Team** provides processes and tools to integrate and evaluate assets early and often.



Business Owners are key stakeholders on the Agile Release Train.



SCALED AGILE © Scaled Agile, Inc.

3-22

Lesson review

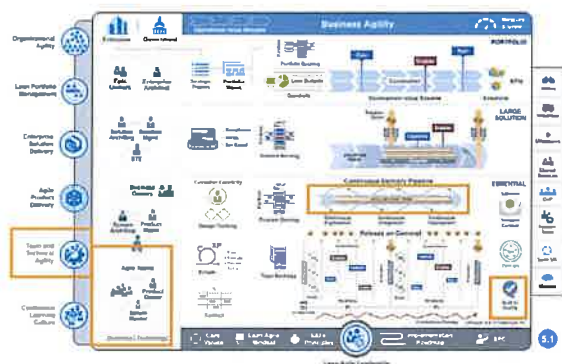
In this lesson you:

- ▶ Discussed how to form cross-functional Agile Teams
- ▶ Reviewed built-in quality practices
- ▶ Explored how to organize Agile Release Trains (ARTs) around the flow of value

Articles used in this lesson

Read these Framework articles to learn more about topics covered in this lesson

- ▶ “Team and Technical Agility”
<https://www.scaledagileframework.com/team-and-technical-agility/>
- ▶ “Built-In Quality”
<https://www.scaledagileframework.com/built-in-quality/>
- ▶ “Agile Teams”
<https://www.scaledagileframework.com/agile-teams/>
- ▶ “Agile Release Train”
<https://www.scaledagileframework.com/agile-release-train/>



Continue your SAFe journey with the following resources

Review the <i>Agile Basics</i> E-Learning: https://bit.ly/Community-GettingStarted	Run an <i>Agile Team Charter Workshop</i> from the Team Formation Toolkit to create the foundation for successful teams in SAFe: https://bit.ly/Community-ToolkitsandTemplates
Apply the guidance from the advanced topic article, "Organizing Agile Teams and ARTs": https://www.scaledagileframework.com/organizing-agile-teams-and-arts-team-topologies-at-scale/	Review the Built-in Quality technical practices in the <i>Agile Software Engineering Vlog</i> series: https://bit.ly/Playlist-SoftwareEngineering
Facilitate effective <i>Team Events</i> using the following tools and guidance: https://bit.ly/Community-SAFeARTandTeamEvents	Run a <i>Team and Technical Agility Assessment</i> to identify improvement opportunities: https://bit.ly/Community-MeasureAndGrow

Lesson notes

Enter your notes below. If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 4

Building Solutions with Agile Product Delivery

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



SCALED AGILE®

Why Agile Product Delivery?

In order to achieve Business Agility, Enterprises must rapidly increase their ability to deliver innovative products and services. To be sure that the Enterprise is creating the right Solutions for the right Customers at the right time, they must balance their execution focus with a Customer focus.



BUSINESS AGILITY PRODUJE DŠME SE ODBOJILI
OD ZAKAZNIKA

Lesson Topics

4.1 Customer Centricity and Design Thinking

4.2 Prioritizing the Program Backlog

4.3 PI Planning

4.4 Develop on Cadence; Release on Demand

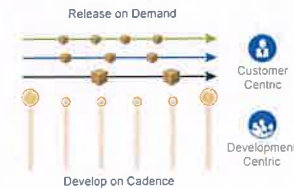
4.5 Building a Continuous Delivery Pipeline with DevOps

Customer Centricity



Design Thinking

Develop on Cadence, Release on Demand



DevOps and the Continuous Delivery Pipeline



© Scaled Agile, Inc.

4-3

Learning objectives

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

- ▶ Express the benefits of a Customer-centric culture
- ▶ Practice applying Design Thinking
- ▶ Prioritize the Program Backlog with weighted shortest job first (WSJF)
- ▶ Participate in a PI Planning event
- ▶ Explain the need to Develop on Cadence; Release on Demand
- ▶ Justify the need to build and maintain a Continuous Delivery Pipeline with DevOps

SCALED AGILE® © Scaled Agile, Inc.

4-4

4.1 Customer Centricity and Design Thinking

SCALED AGILE © Scaled Agile, Inc.

4-5



Discussion: Customer Centricity



- ▶ **Step 1:** Discuss as a group:
 - Why is it important to maintain focus on the Customer?
 - What are some of the characteristics of a Customer-centric Enterprise?
- ▶ **Step 2:** Be prepared to share with the class.



SCALED AGILE © Scaled Agile, Inc.

4-6

Why Customer Centricity?

Customer-centric Enterprises deliver whole-product Solutions that are designed with a deep understanding of Customer needs.



SCALED AGILE © Scaled Agile, Inc.

4-7

Customer Centricity is a mindset

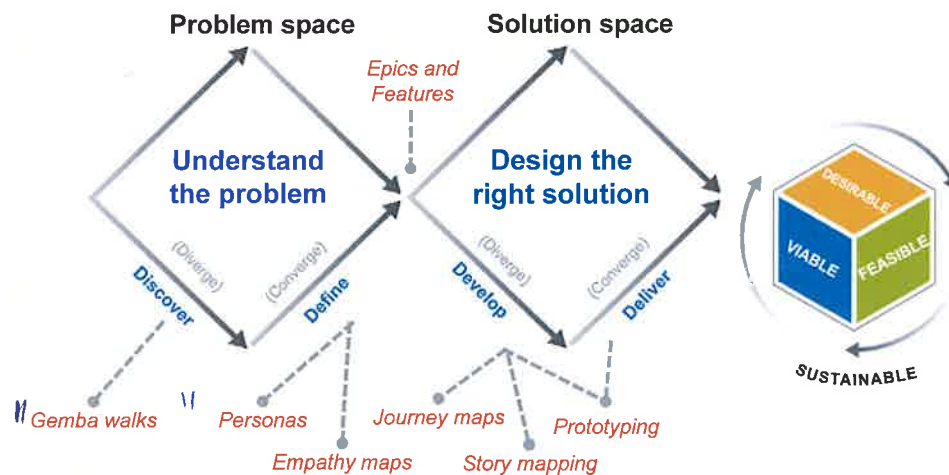
Whenever a customer-centric Enterprise makes a decision, it fully considers the effect it will have on its end users.



SCALED AGILE © Scaled Agile, Inc.

4-8

What is Design Thinking?



Design Thinking is a clear and continuous understanding of the target market, Customers, the problems they are facing, and the jobs to be done.

SCALED AGILE © Scaled Agile, Inc.

4-9

GEMBA - μίστο όπου

Use personas to understand Customers

Personas are fictional characters that represent the different people who might use your product. Personas:

- Convey the problems they're facing in context and key triggers for using the product
- Capture rich, concise information that inspires great products without unnecessary details



Cary the Consumer

Age: 36

Location: Reno, Nevada, USA

Time in App: 10 minutes

"I'm a working dad with three children ages 3, 6, and 10. I'm also in a band, which means I want to spend as much time as possible with my kids and my band. I need my package delivered on time so that I can maximize time with my family."

I like technology! I have an iPhone, iPad, and nice home Wi-Fi setup

I am not home on some weekends

I'd rather order online than dial the phone and talk to somebody

My wife also works during the week, so she doesn't have much spare time to help

Text is my favorite form of communication with suppliers

I don't own a computer, only tablets and phones.

SCALED AGILE © Scaled Agile, Inc.

4-10

Use empathy maps to identify with Customers

- ▶ The empathy map is a tool that helps teams develop deep, shared understanding and empathy for the Customer
- ▶ Use it to design better user experiences and Value Streams



SCALED AGILE © Scaled Agile, Inc.

4-11



Activity: Empathy mapping

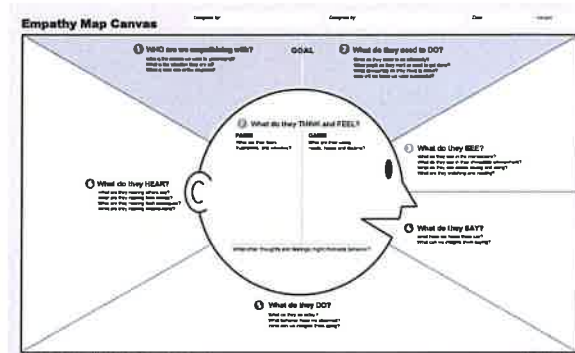
Prepare



Share



- ▶ **Step 1:** In your group, create an empathy map using the example in your workbook or the template provided.
- ▶ **Step 2:** Select a user or Customer of a product or service from one of the companies at your table.
- ▶ **Step 3:** Following the sequence of numbers, fill in each section of the empathy map.
- ▶ **Step 4:** Discuss with your group how the empathy map can inform Solution development. Be prepared to share your insights with the class.



SCALED AGILE © Scaled Agile, Inc.

4-12

Empathy Map Canvas

Designed for:

Designed by:

Date:

Version:

Empathy Map Canvas

1 WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

GOAL

2 What do they need to DO?
What do they need to do differently?
What job(s) do they want to get done?
What decision(s) do they need to make?
How will we know they were successful?

3 What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

6 What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

What do they THINK and FEEL?

7 PAINS
What are their fears, frustrations, and anxieties?

GAINS
What are their wants, needs, hopes and dreams?

What other thoughts and feelings might motivate their behavior?

4 What do they SAY?
What have we heard them say?
What can we imagine them saying?

5 What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

Last updated on 16 July 2017. Download a copy of this canvas at <http://gamestorming.com/empathy-map/>

© 2017 Dave Gray, spide.com

Empathy Mapping

Instructions:

Step 1: Select a user customer of a product or service from one of your companies in your group.

Step 2: Following the sequence of numbers, fill in each section of the empathy map in the spaces below.

Step 3: Discuss with your group how the empathy map can inform Solution development. Be prepared to share with the class.



#1

GOAL: WHO are we empathizing with?

Who is the person we want to understand?

What is the situation they are in?

What is their role in the situation?

Empathy Mapping

#2

GOAL: What do they need to *DO*?

What do they need to do differently?

What job(s) do they want or need to get done?

What decision(s) do they need to make?

How will we know we were successful?

#3

What do they *SEE*?

What do they see in the marketplace?

What do they see in their immediate environment?

What do they see others saying and doing?

What are they watching and reading?

Empathy Mapping

#4

What do they *SAY*?

What have we heard them say?

What can we imagine them saying?

#5

What do they *DO*?

What do they do today?

What behavior have we observed?

What can we imagine them doing?