Microsoft

WorkshopPLUS: DevOps Fundamentals

Module 1: Planning

Conditions and Terms of Use

Microsoft Confidential

This training package is proprietary and confidential and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet website references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

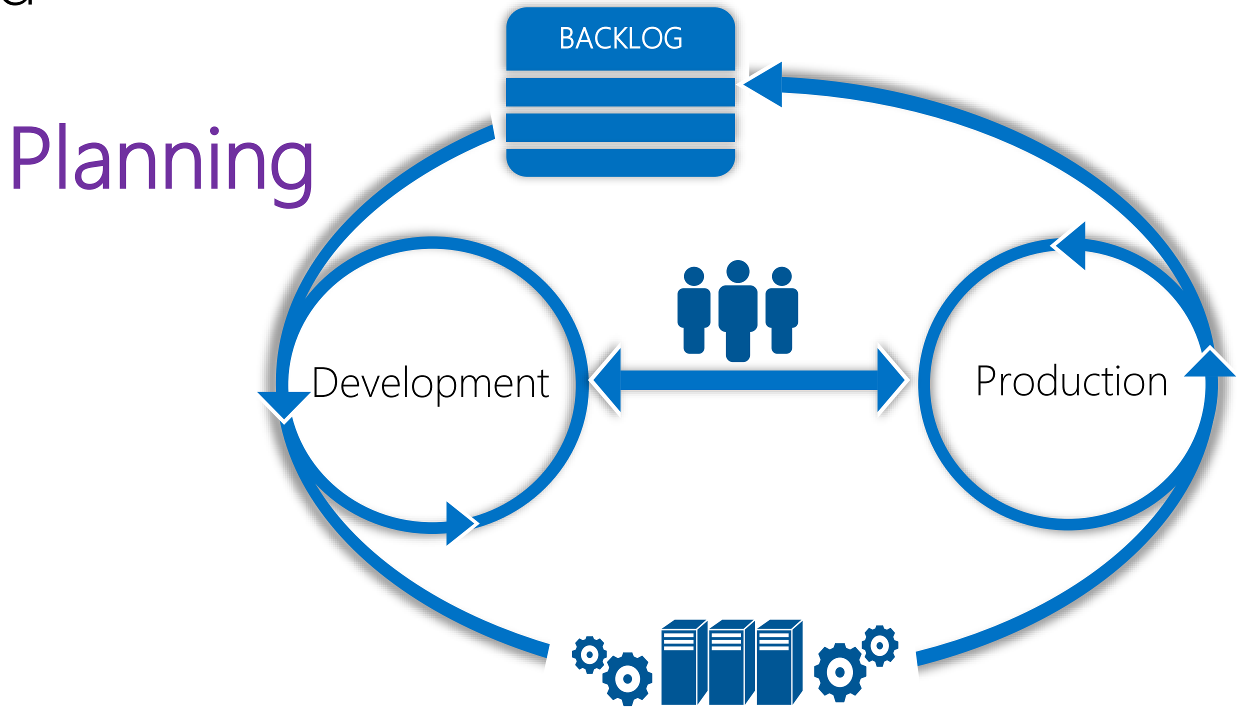
Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at
https://www.microsoft.com/en-us/legal/intellectualproperty/permissions/default.aspx

Microsoft®, Internet Explorer®, Outlook®, SkyDrive®, Windows Vista®, Zune®, Xbox 360®, DirectX®, Windows Server® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Module Objectives

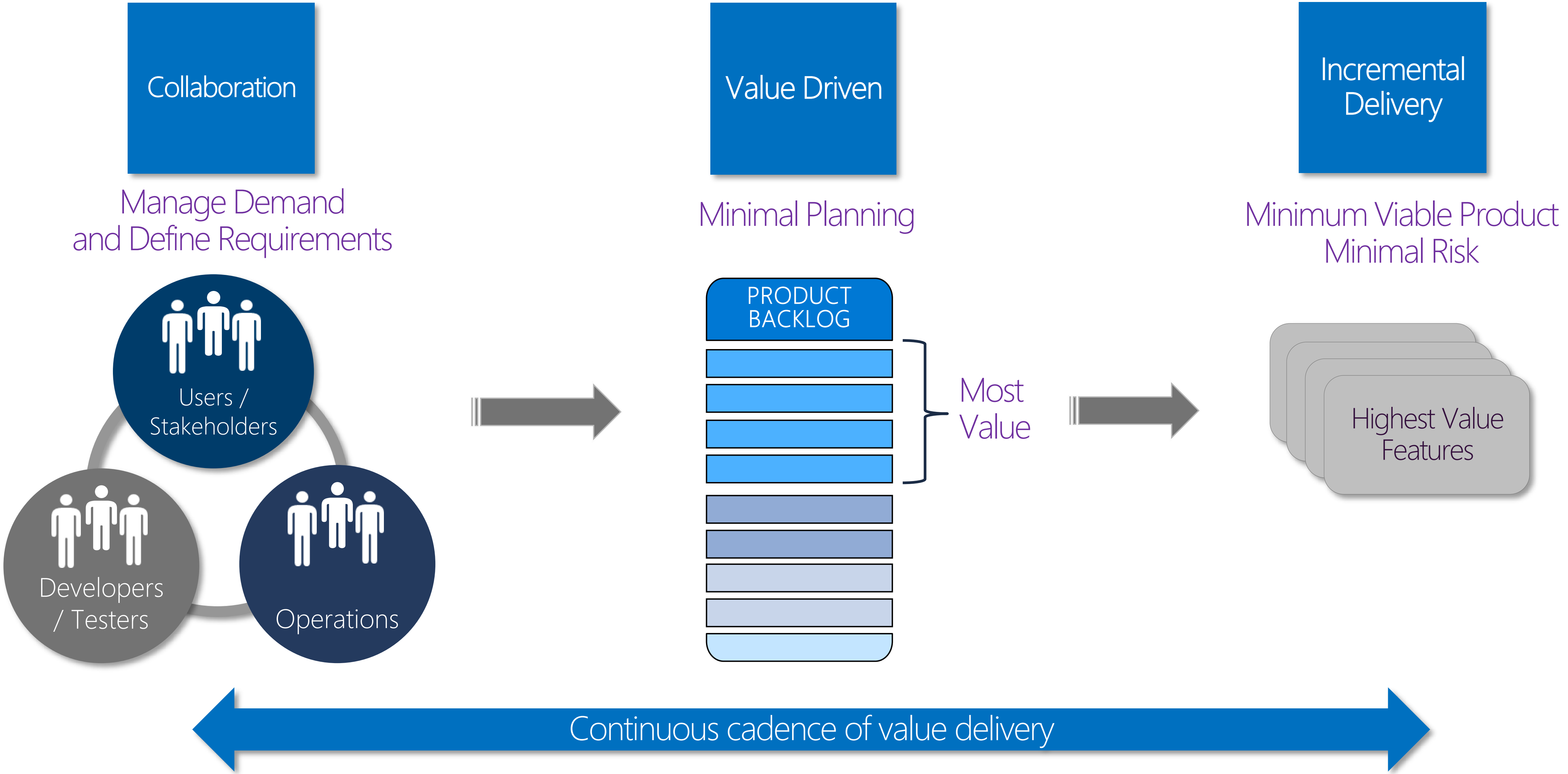
- Understand how Agile planning is a crucial pillar of DevOps, and
- Learn how to scale Agile practices throughout the enterprise.



“Plans are worthless, but planning is everything”

Dwight Eisenhower

Planning for Continuous Delivery of Value



Collaboration

Manage Demand and Define Requirements

Value Driven

Minimal Planning

Incremental Delivery

Minimum Viable Product
Minimal Risk

But what is Agile?

(not a methodology, but a set of values and principles)

Agile Manifesto

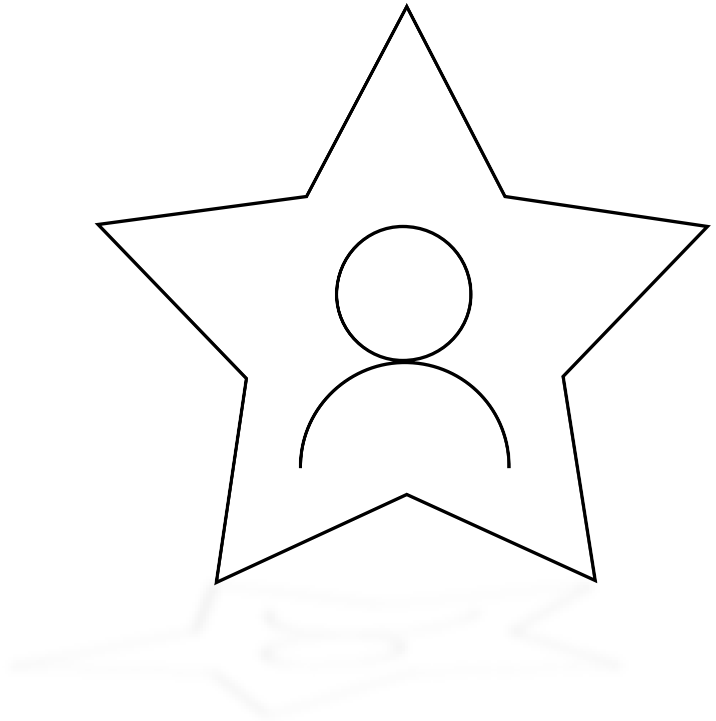
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and Interactions *over* processes and tools
- Working Software *over* comprehensive documentation
- Customer Collaboration *over* contract negotiation
- Responding to Change *over* following a plan

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

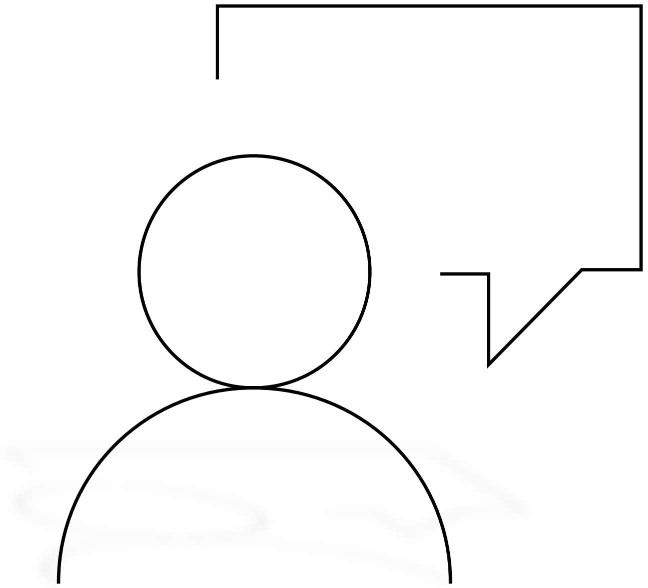
Agile Principle #1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.



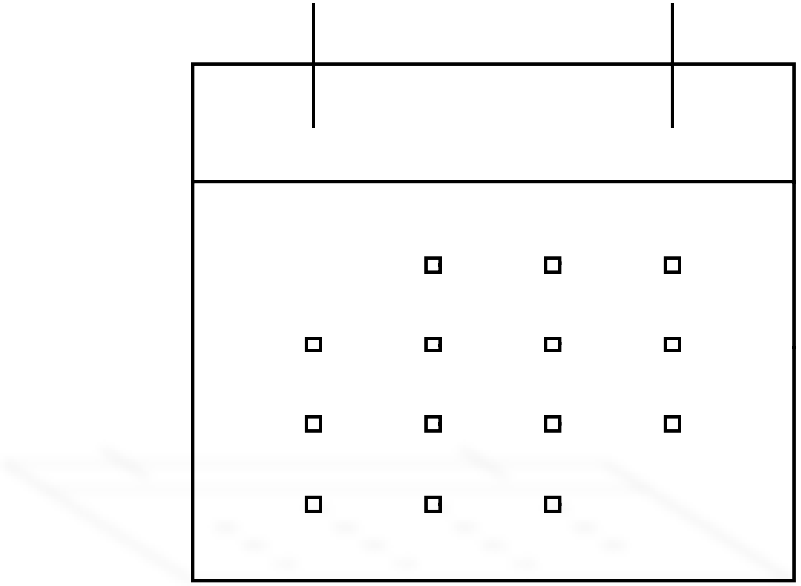
Agile Principle #2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.



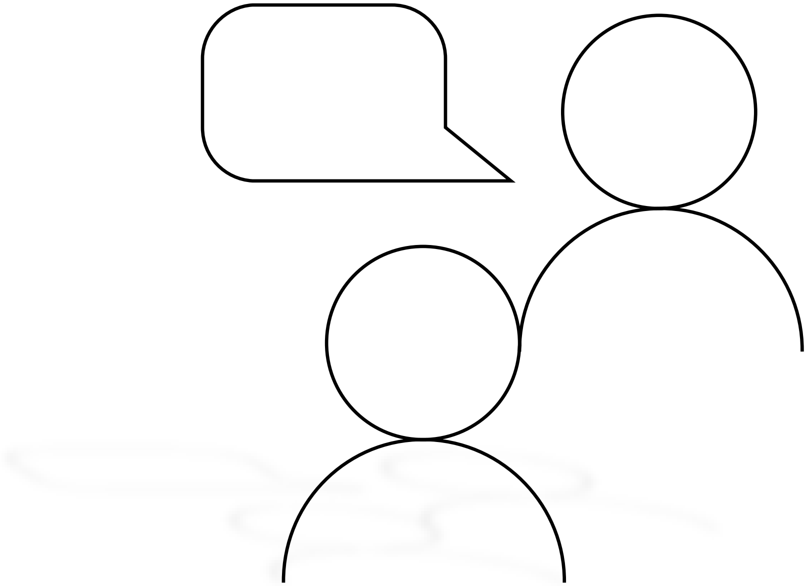
Agile Principle #3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.



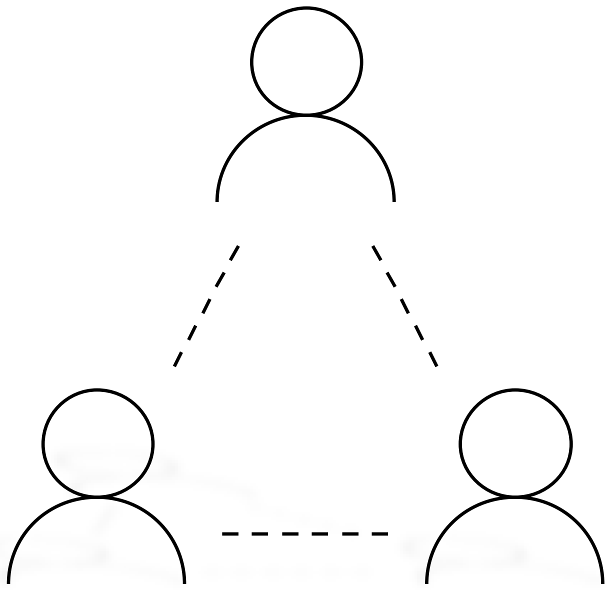
Agile Principle #4

Businesspeople and developers must work together daily throughout the project.



Agile Principle #5

Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.



Agile Principle #6

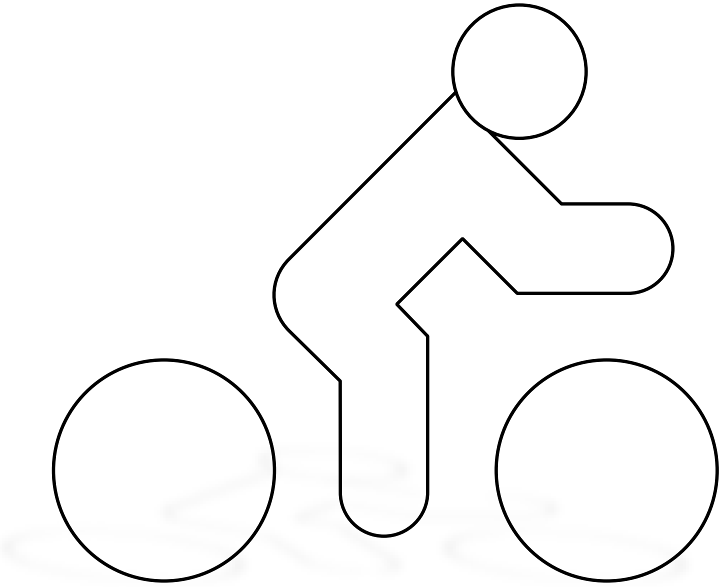
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Principle #7

Working software is the primary measure of progress.

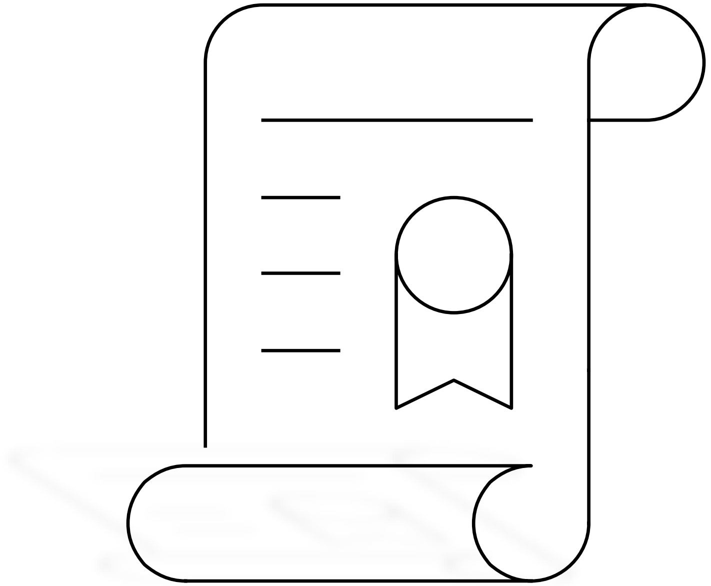
Agile Principle #8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.



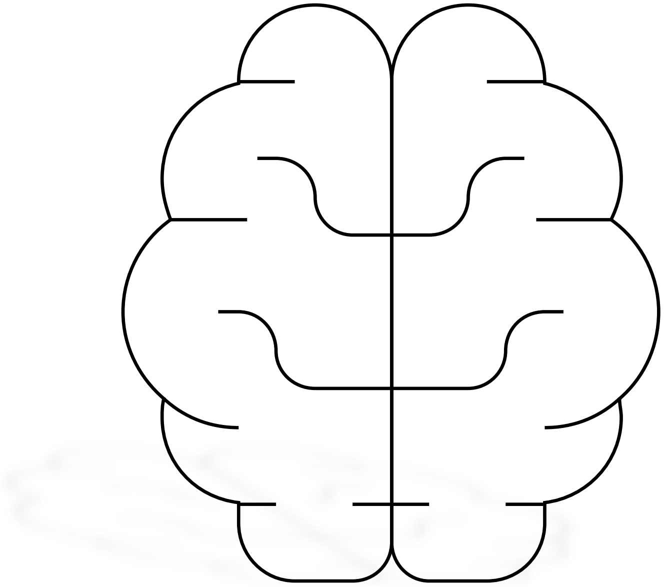
Agile Principle #9

Continuous attention to technical excellence and good design enhances agility.



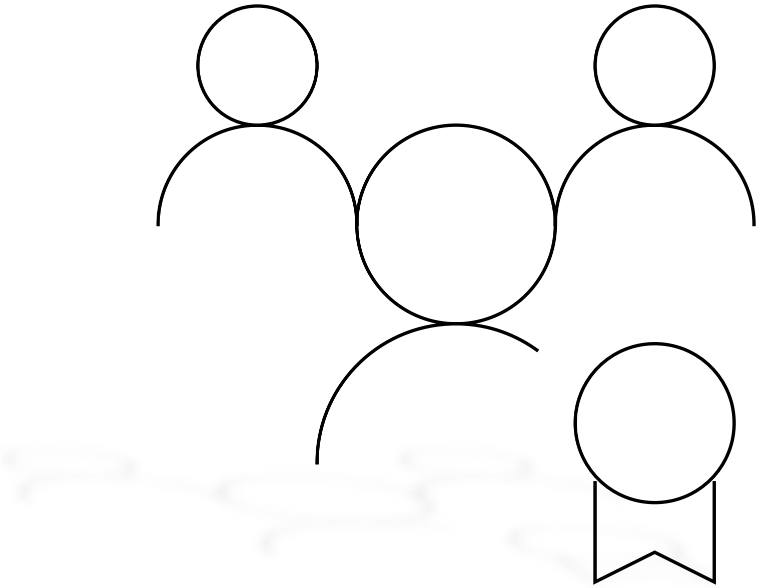
Agile Principle #10

Simplicity - the art of maximizing the amount of work *not done* is essential.



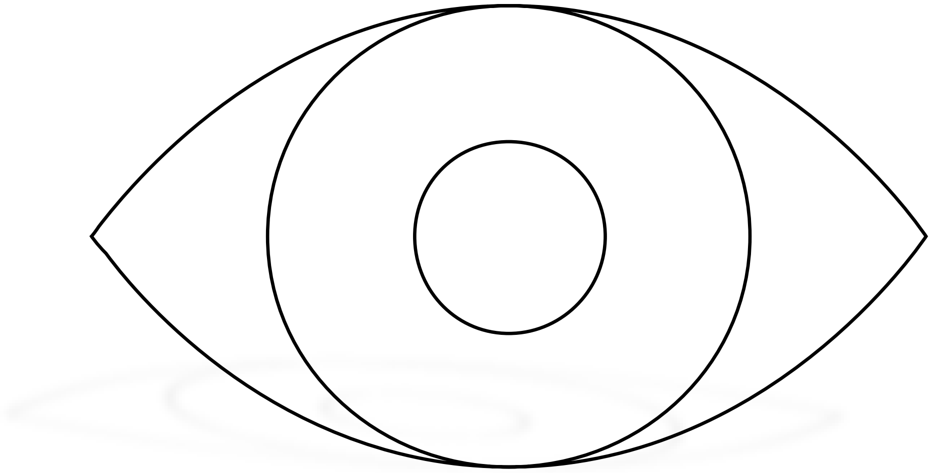
Agile Principle #11

The best architectures, requirements, and designs emerge from self-organizing teams.



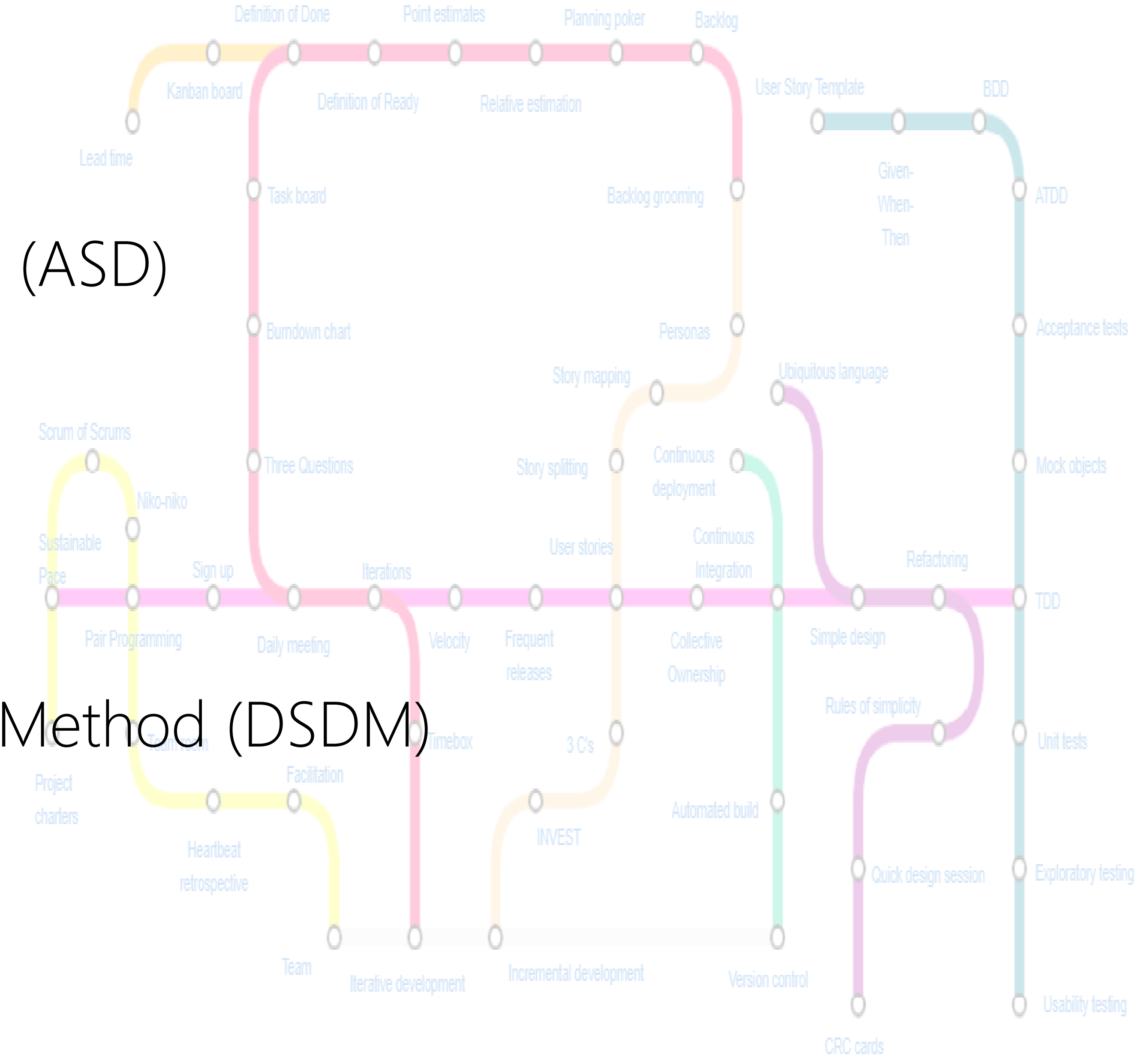
Agile Principle #12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



What about the methodology?

Agile Methodologies



- Adaptive Software Development (ASD)
- Scrum
- eXtreme Programming (XP)
- Lean Software Development
- Scaled Agile Framework (SAFe)
- Dynamic Systems Development Method (DSDM)
- Agile Unified Process (AUP)
- Kanban
- … *others*

Choose the methodology that works for you but stay true to its principles!

Image from https://www.agilealliance.org/agile101/subway-map-to-agile-practices/

Why Lean?

Some Lean concepts are built into Azure DevOps Server, and Azure DevOps Services such as Cycle Time, Lead Time, Kanban

Lean Methodology

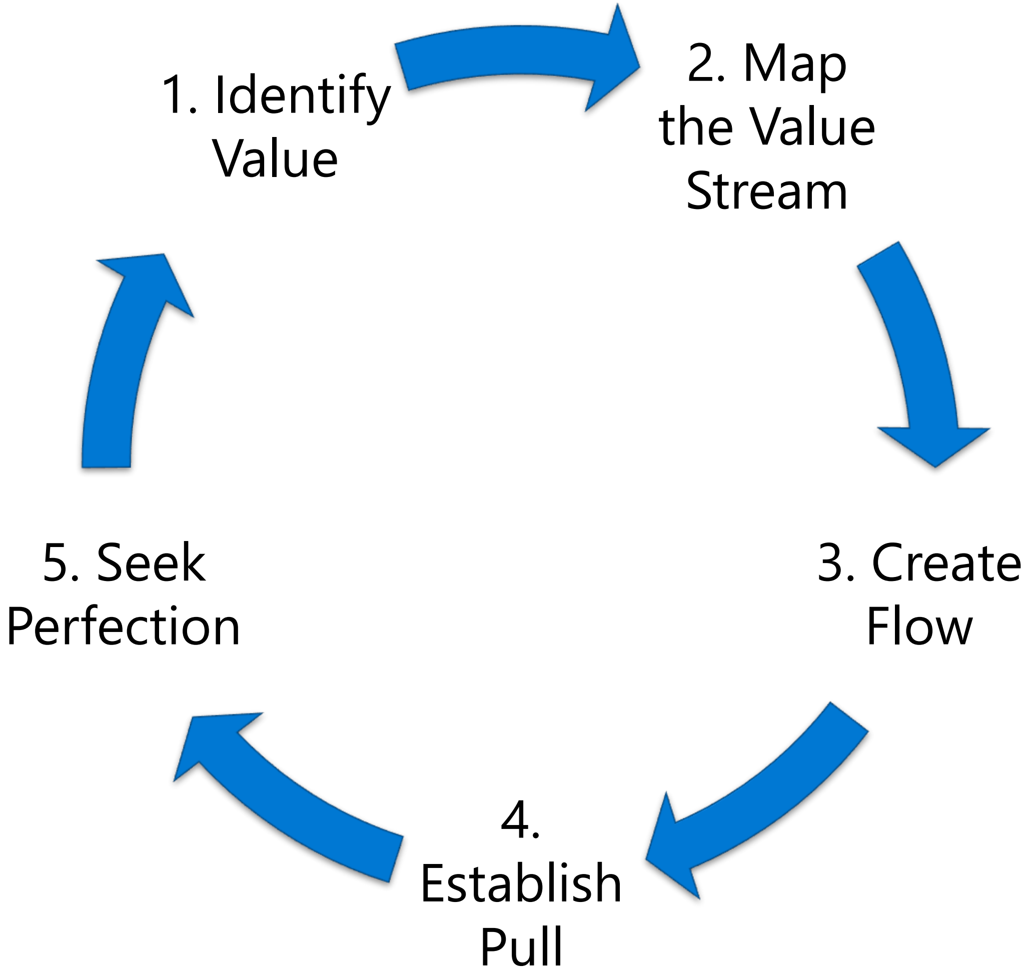
- Based on the Toyota Production System
- The core idea is to maximize customer value while minimizing waste - creating more value for customers with fewer resources.
- To accomplish this, lean thinking changes the focus of management from optimizing separate technologies, assets, and vertical departments to optimizing the flow of products and services through entire value streams that flow horizontally across technologies, assets, and departments to customers.

Toyota Production System

"A production system based on the philosophy of achieving the complete elimination of all waste in pursuit of the most efficient methods." *Toyota Production System | Vision & Philosophy | Company | Toyota Motor Corporation Official Global Website*. [online] Toyota Motor Corporation Official Global Website. Available at: <https://global.toyota/en/company/vision-and-philosophy/production-system/> [Accessed 13 October 2020].

- Continuous improvement
- Respect for people
- Long-term philosophy
- The right process will produce the right results
- Add value to the organization by developing your people and partners
- Continuously solving problems drives organization learning

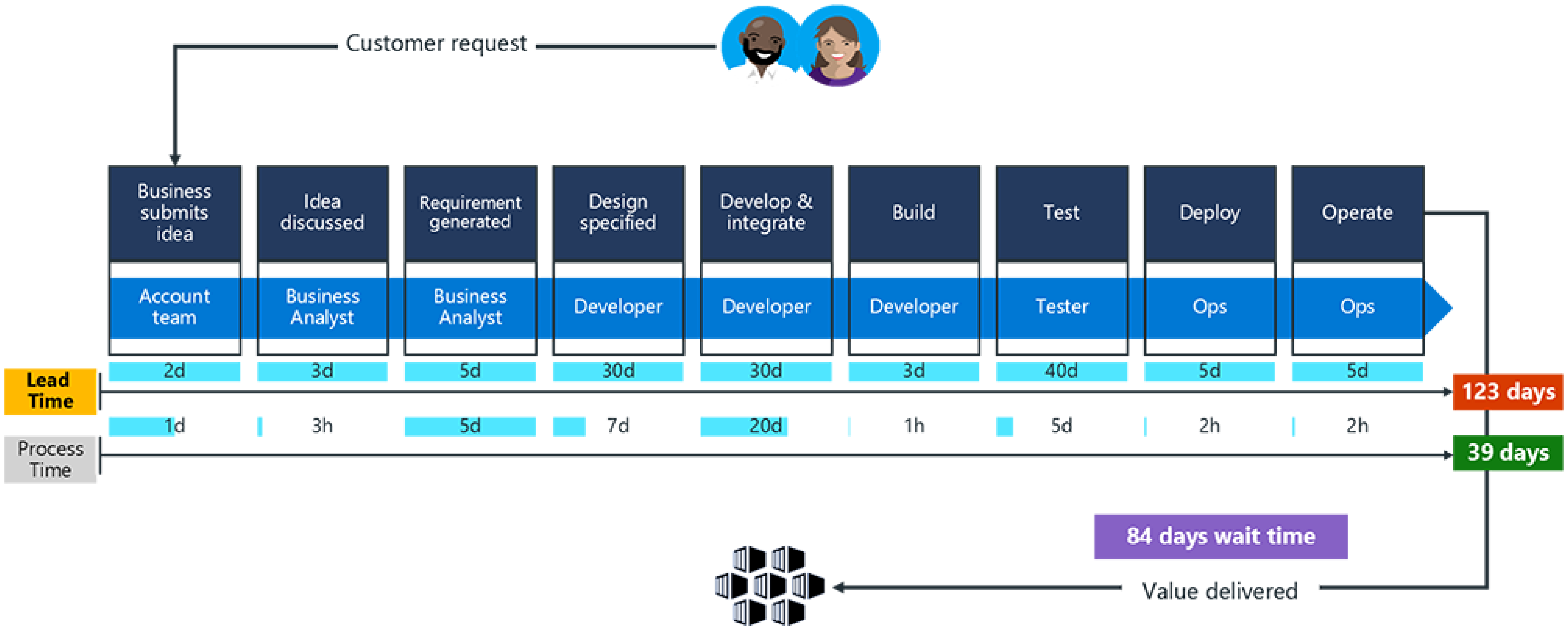
Steps in the Lean Process



1. Identify Value – Always defined by the customer's needs for a specific product.
2. Map the Value Stream – Once the value (end goal) has been determined, map all the steps and processes from raw material to customer delivery
3. Flow – After waste is removed, be sure remaining steps flow smoothly with no interruptions, delays, or bottlenecks
4. Pull – With improved flow, time to customer can be dramatically improved. Can be delivered Just-In-Time or customer can "pull" from you as needed
5. Seek Perfection – Adopt Lean thinking and process improvement in your culture

Lead Time

- The time elapsed between a customer placing an order and receiving the product ordered
- Process time is a measure of the time a product is being worked on in a machine or by an employee in a work area



Customer request

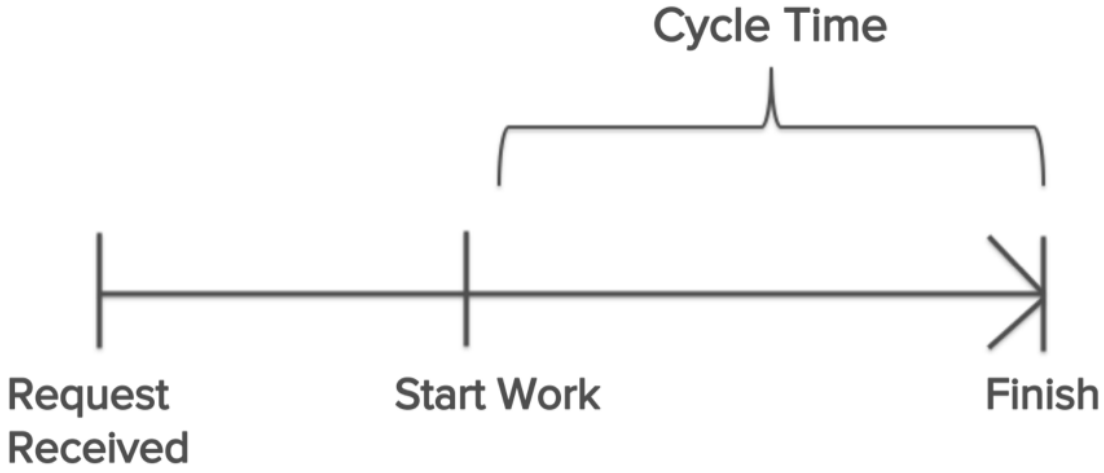
	Business submits idea	Idea discussed	Requirement generated	Design specified	Develop & integrate	Build	Test	Deploy	Operate	
	Account team	Business Analyst	Business Analyst	Developer	Developer	Developer	Tester	Ops	Ops	
Lead Time	2d	3d	5d	30d	30d	3d	40d	5d	5d	123 days
Process Time	1d	3h	5d	7d	20d	1h	5d	2h	2h	39 days

84 days wait time

Value delivered

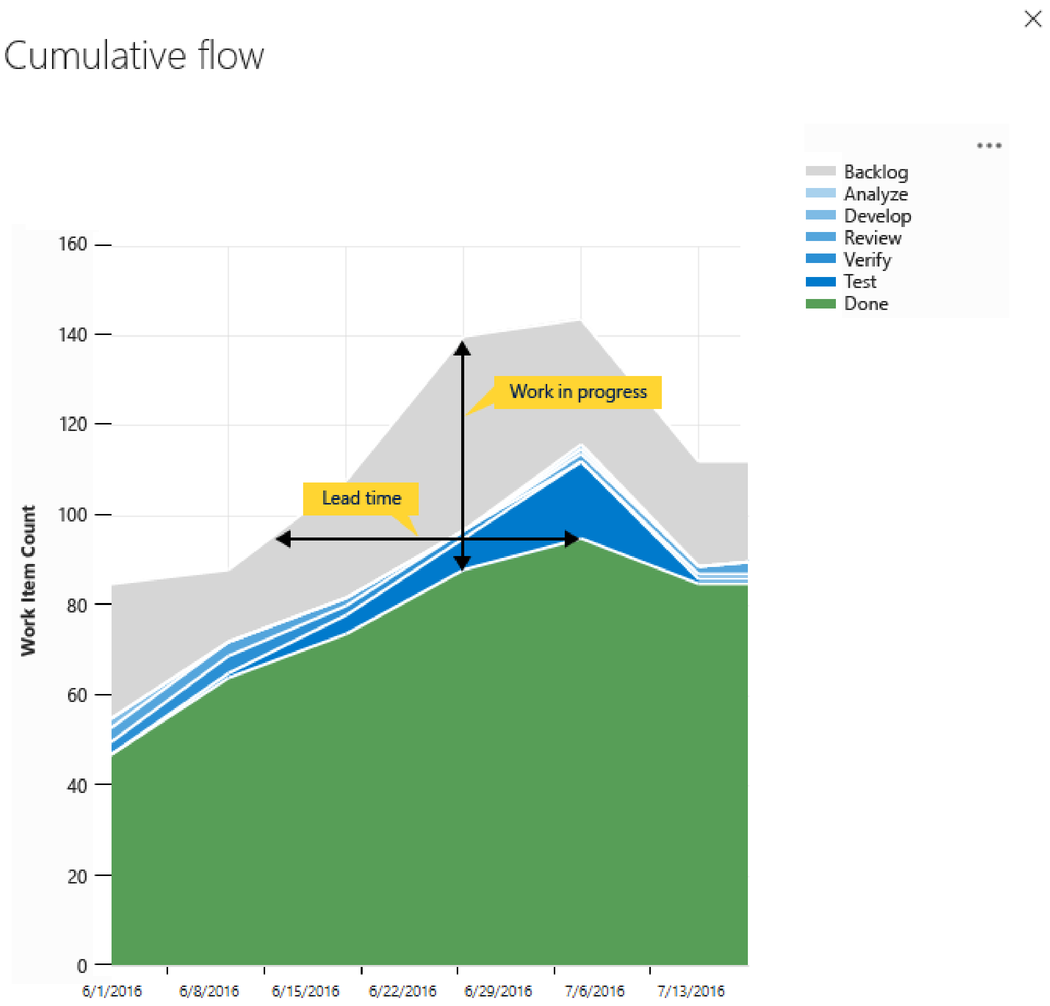
Cycle Time

- Cycle Time is the total time from the beginning to the end of your process, as defined by you and your customer.
- Cycle Time includes process time, during which a unit is acted upon to bring it closer to an output, and delay time, during which a unit of work is spent waiting to take the next action.
- Cycle Time is measured by the amount of time per unit (minutes/customer, hours/part, etc.…)



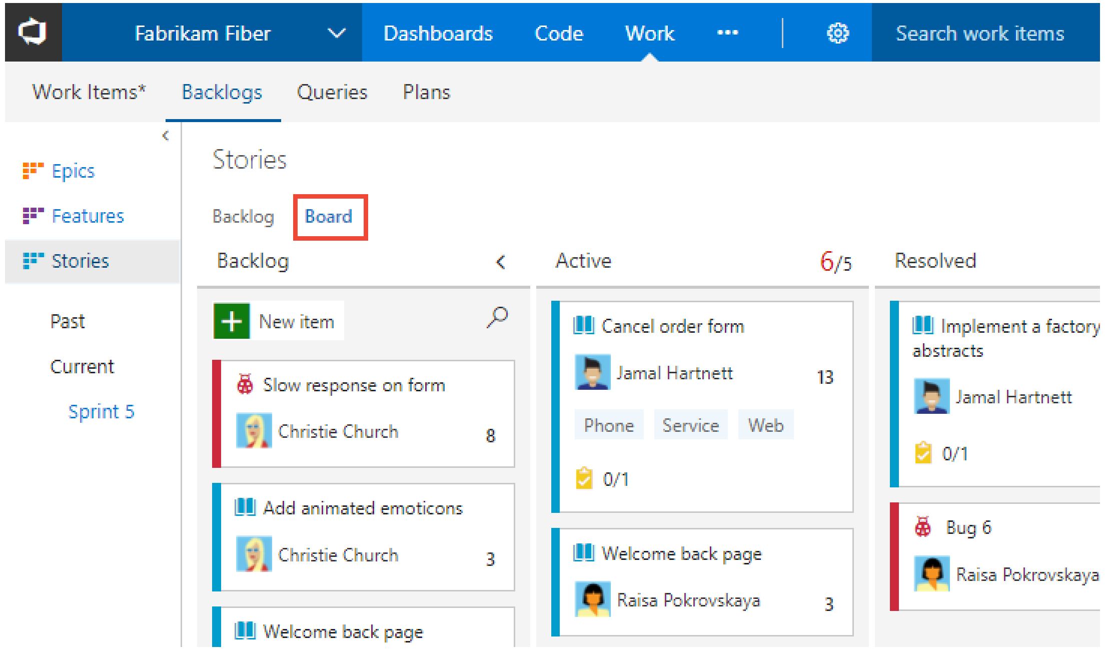
Cumulative Flow

- Shows the amount of work in each queue and process block over time



Kanban

- Visualize your workflow using a board showing work in progress
- Set WIP (work in progress) limits



Backlog

- Slow response on form — Christie Church — 8
- Add animated emoticons — Christie Church — 3
- Welcome back page

Active 6/5

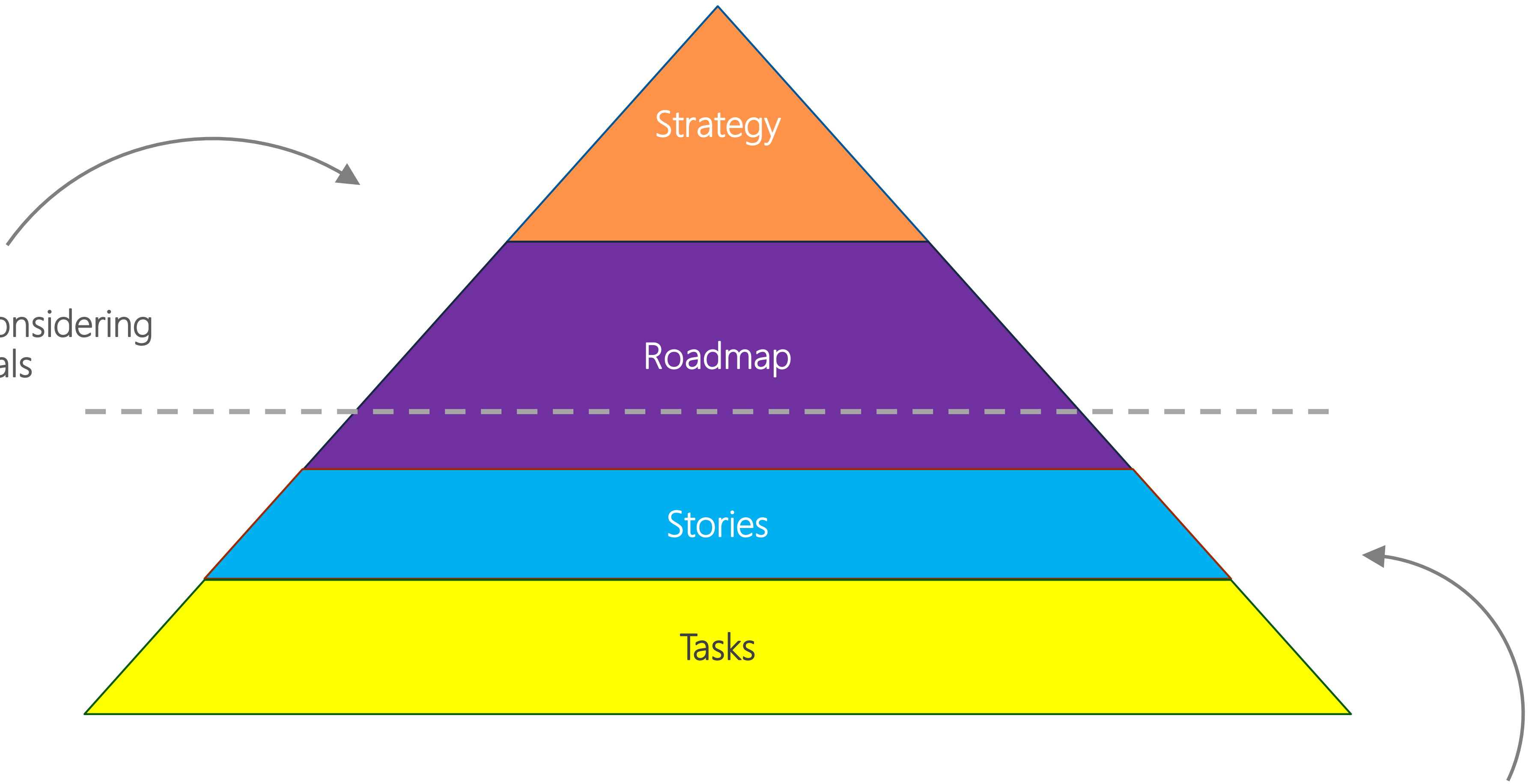
- Cancel order form — Jamal Hartnett — 13 — Phone, Service, Web — 0/1
- Welcome back page — Raisa Pokrovskaya — 3

Resolved

- Implement a factory abstracts — Jamal Hartnett — 0/1
- Bug 6 — Raisa Pokrovskaya

How do we scale Agile to the overall organization?

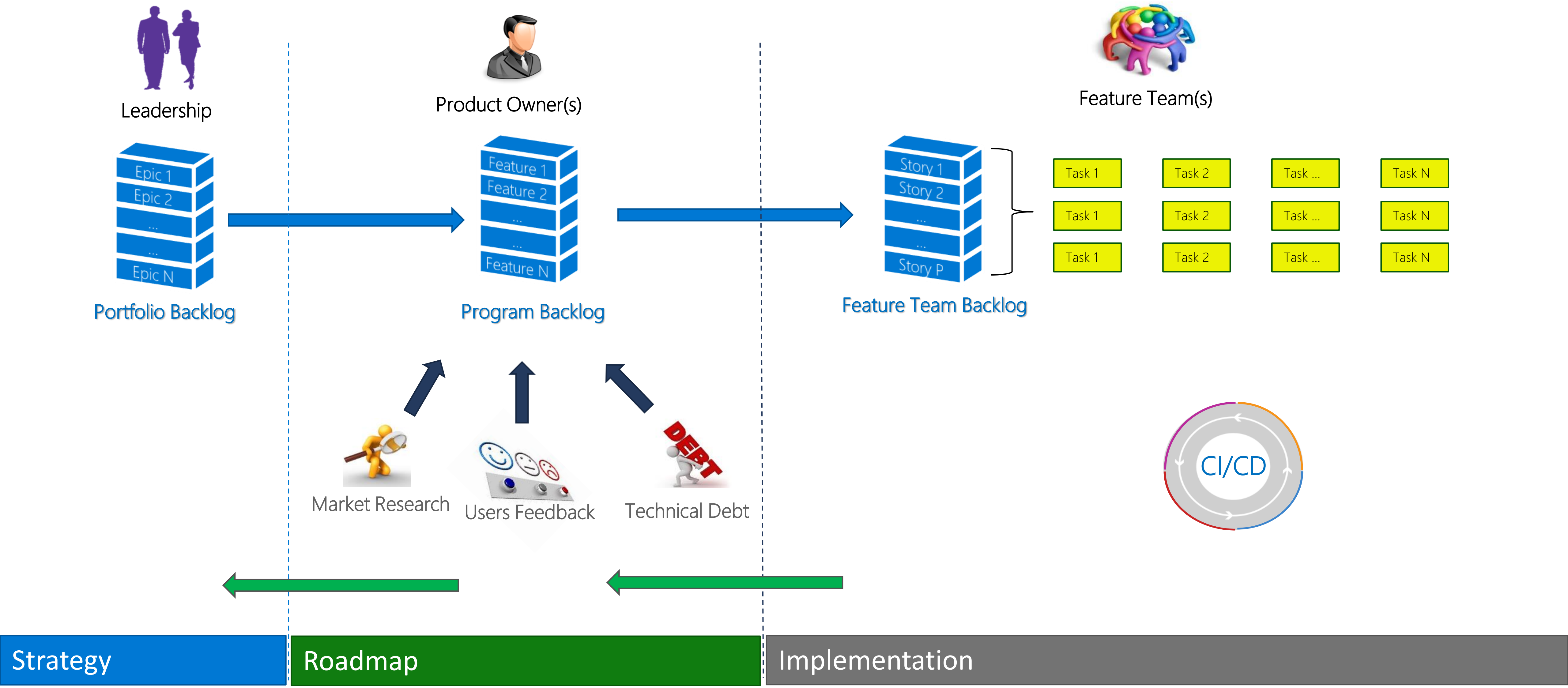
Aligned Autonomy



Alignment
The big picture considering your business goals

Autonomy
The detail about what your teams will deliver to achieve our business goals

Agile at Scale – Hierarchical Backlogs



Leadership

Epic 1
Epic 2
...
...
Epic N

Portfolio Backlog

Product Owner(s)

Feature 1
Feature 2
...
...
Feature N

Program Backlog

Market Research

Users Feedback

Technical Debt

Feature Team(s)

Story 1
Story 2
...
...
Story P

Feature Team Backlog

Task 1	Task 2	Task ...	Task N
Task 1	Task 2	Task ...	Task N
Task 1	Task 2	Task ...	Task N

CI/CD

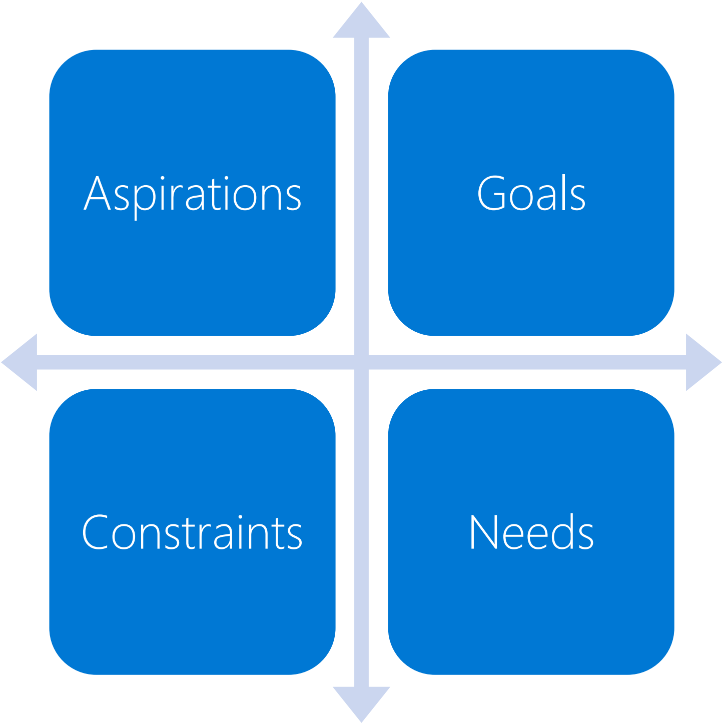
Strategy

Roadmap

Implementation

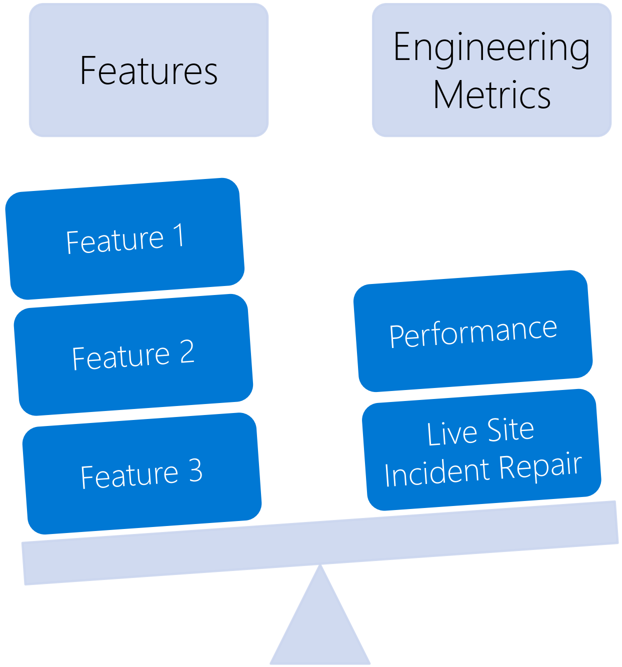
Portfolio Planning

- Sets the mid-term vision
- Trusts in its programs and teams with *HOW* to achieve the vision
  - Alignment and Autonomy
- Continuously seeks feedback

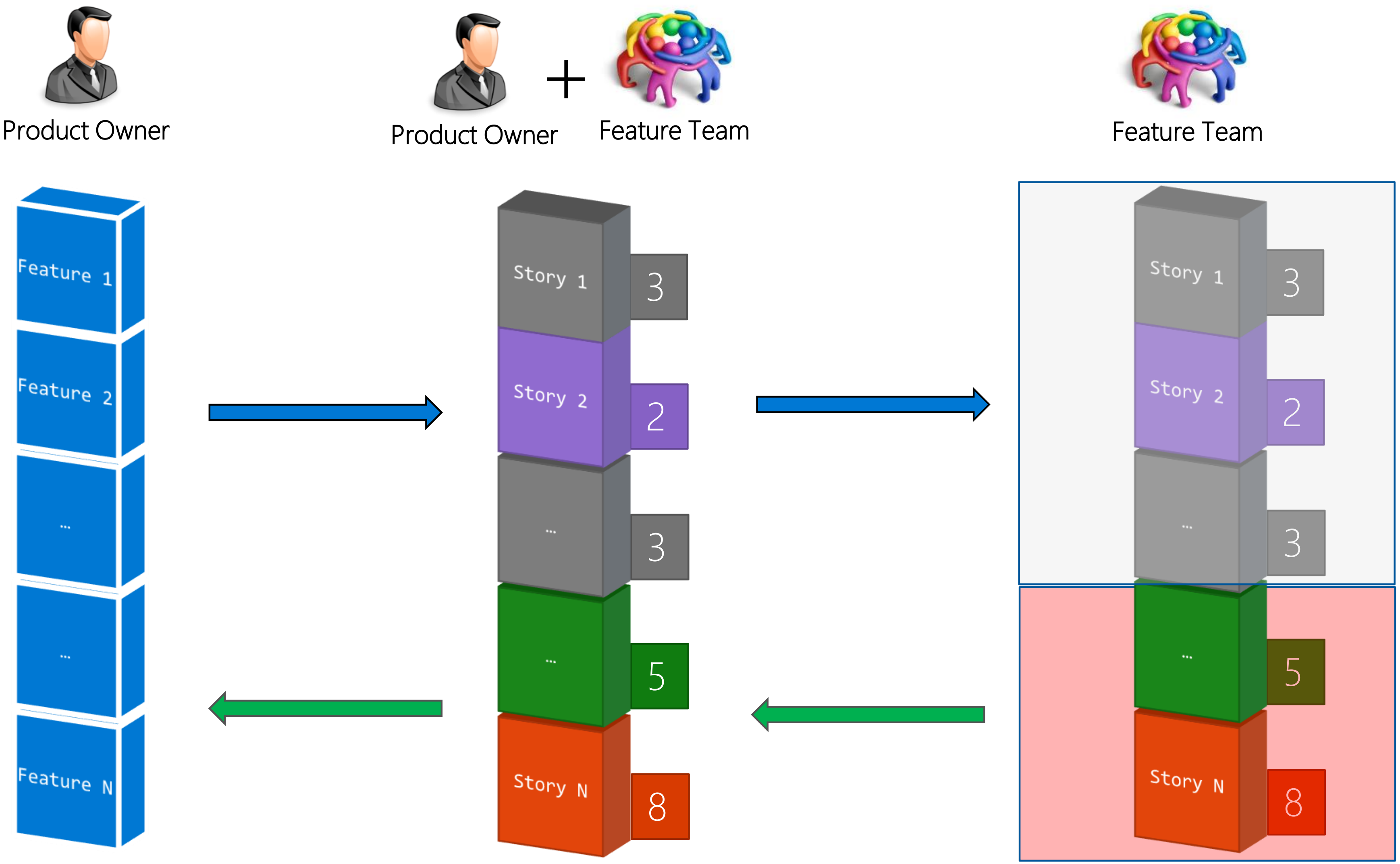


Program Roadmap

- Bring engineering health and debt measures into planning
- Measure what you want to move
- Consider alignment of metric to desired action carefully



Agile Methodologies – Scrum Feature Team



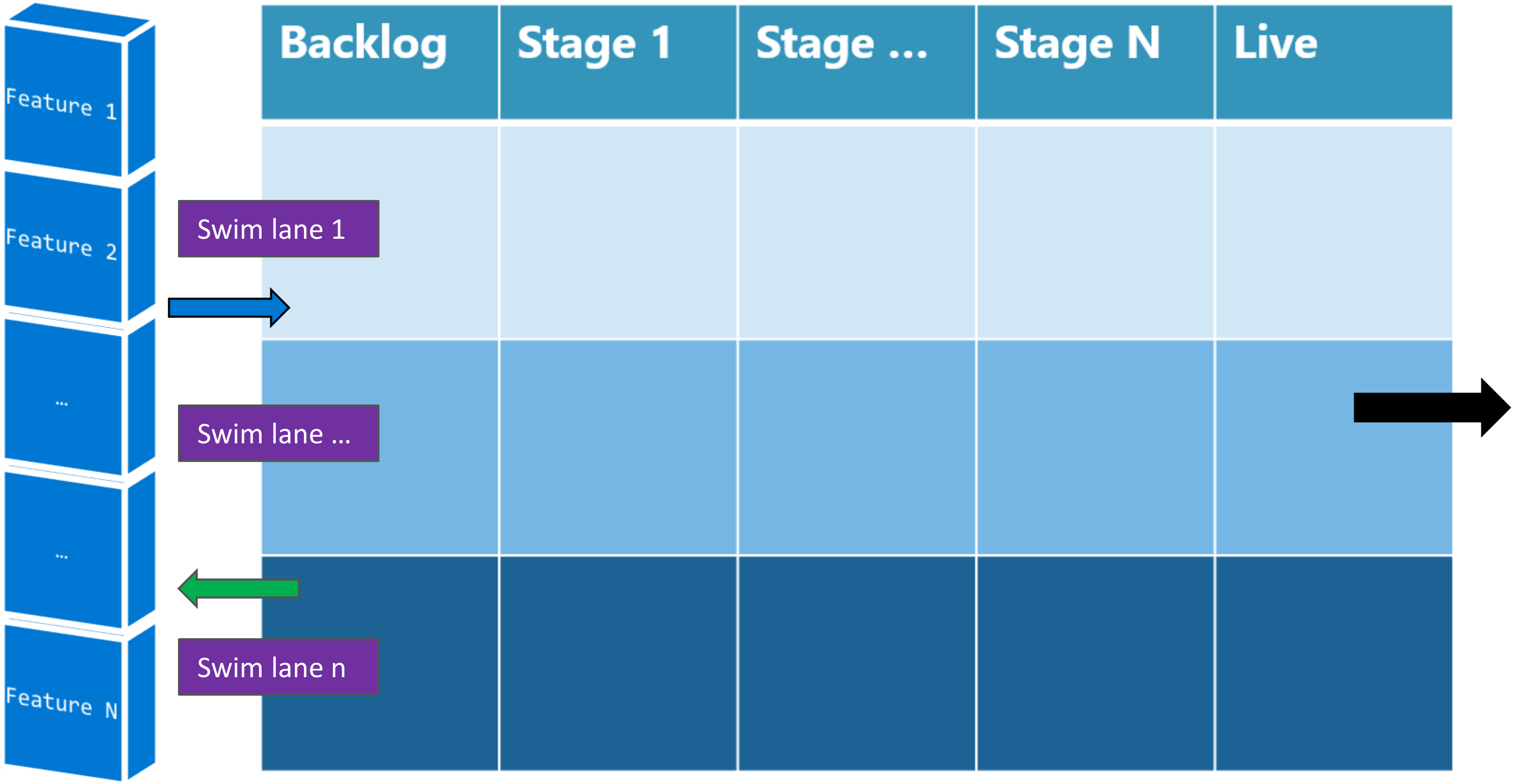
Sprint backlog chosen based on velocity history in order to maintain a constant cadence and based on business priorities

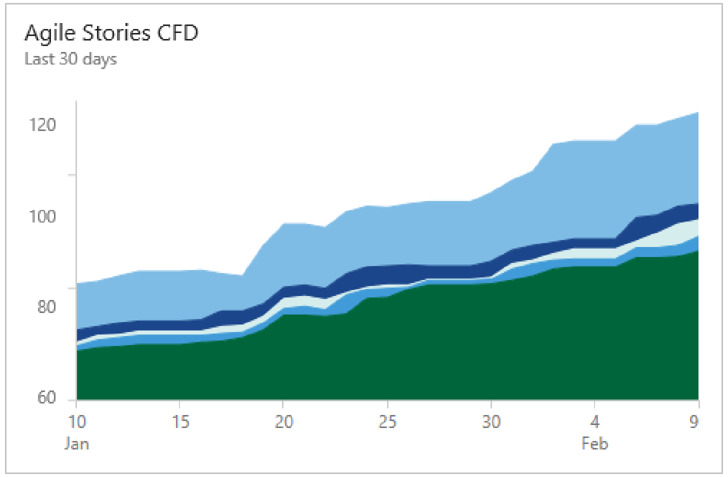
Set Priorities | Estimated and Prioritized backlog | Committed backlog for next Sprint

Agile Methodologies – Kanban Feature Team

Product Owner

Feature Team





Ensures that the team is continuously delivering the same amount of value at anytime for any stage or any swim lane.

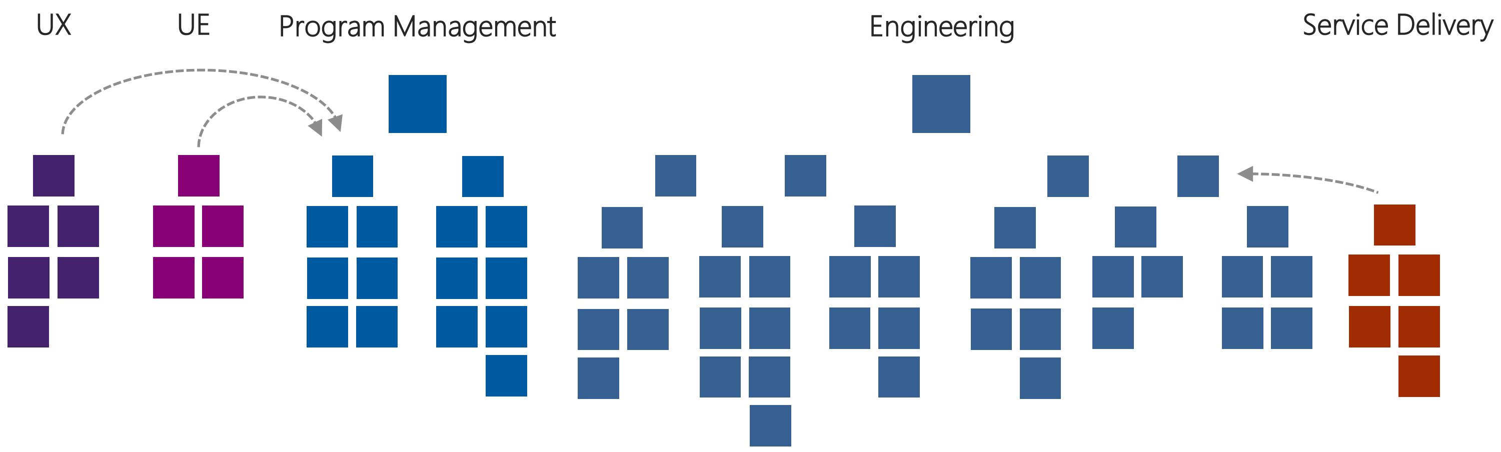
Set Priorities | Define Swim Lanes and Process | Pull Next Item from Backlog

Testimonial #3: Agile Planning at Scale

People

(Roles & Team Structure)

Roles

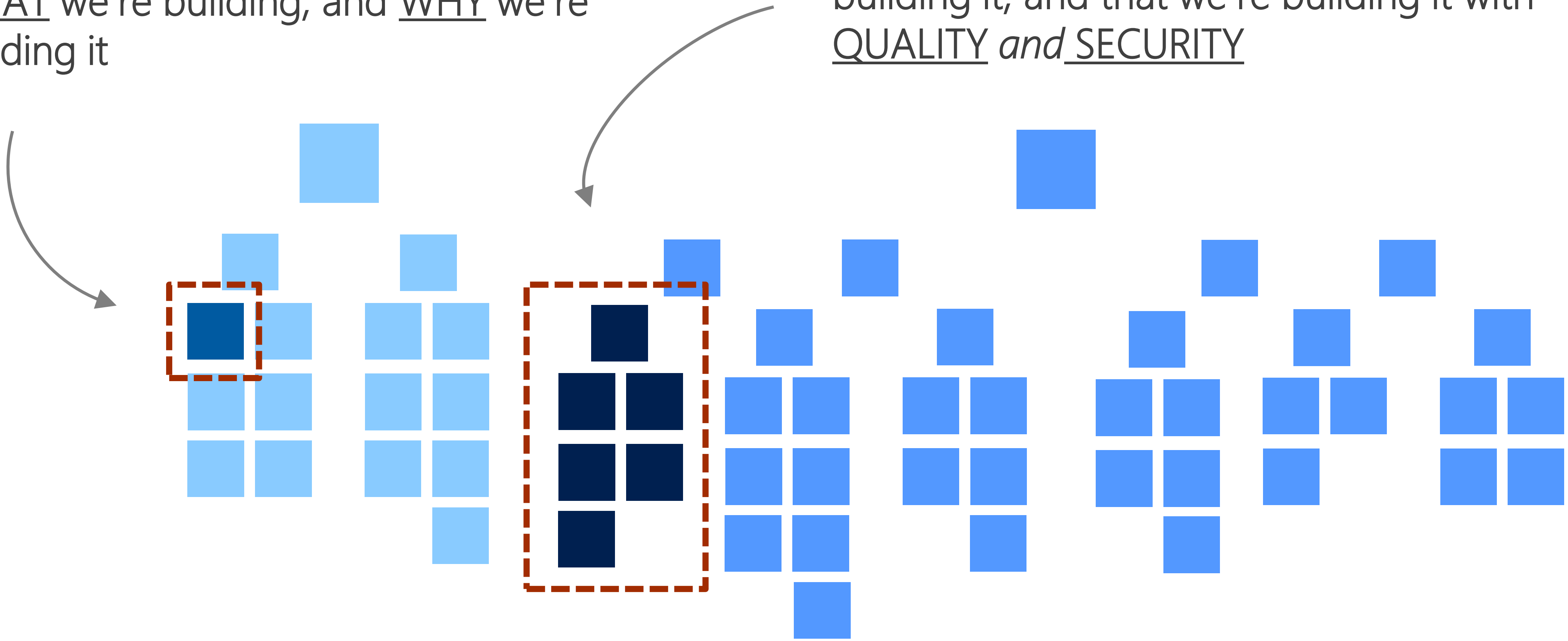


Service Delivery is integrated directly into our organization.

Teams

Program Management is responsible for: WHAT we're building, and WHY we're building it

Engineering is responsible for HOW we're building it, and that we're building it with QUALITY *and* SECURITY



Shielding Distractions: L-Team

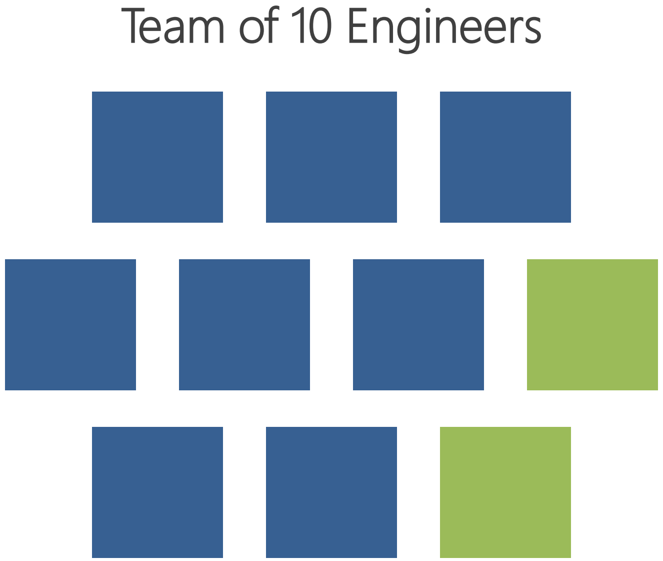
- A strategy adopted by our teams to provide focus and assist with an interrupt culture.

- The team self-organizes each sprint into two distinct sub-teams: Features and Shield

🟦 **F-Team**
(feature team)
Works on committed features (new work)

🟩 **L-Team**
(live site team)
Deals with all live-site issues and interruptions

- Rotating role established by the team
- L-Team size may vary based on live site debt or demand

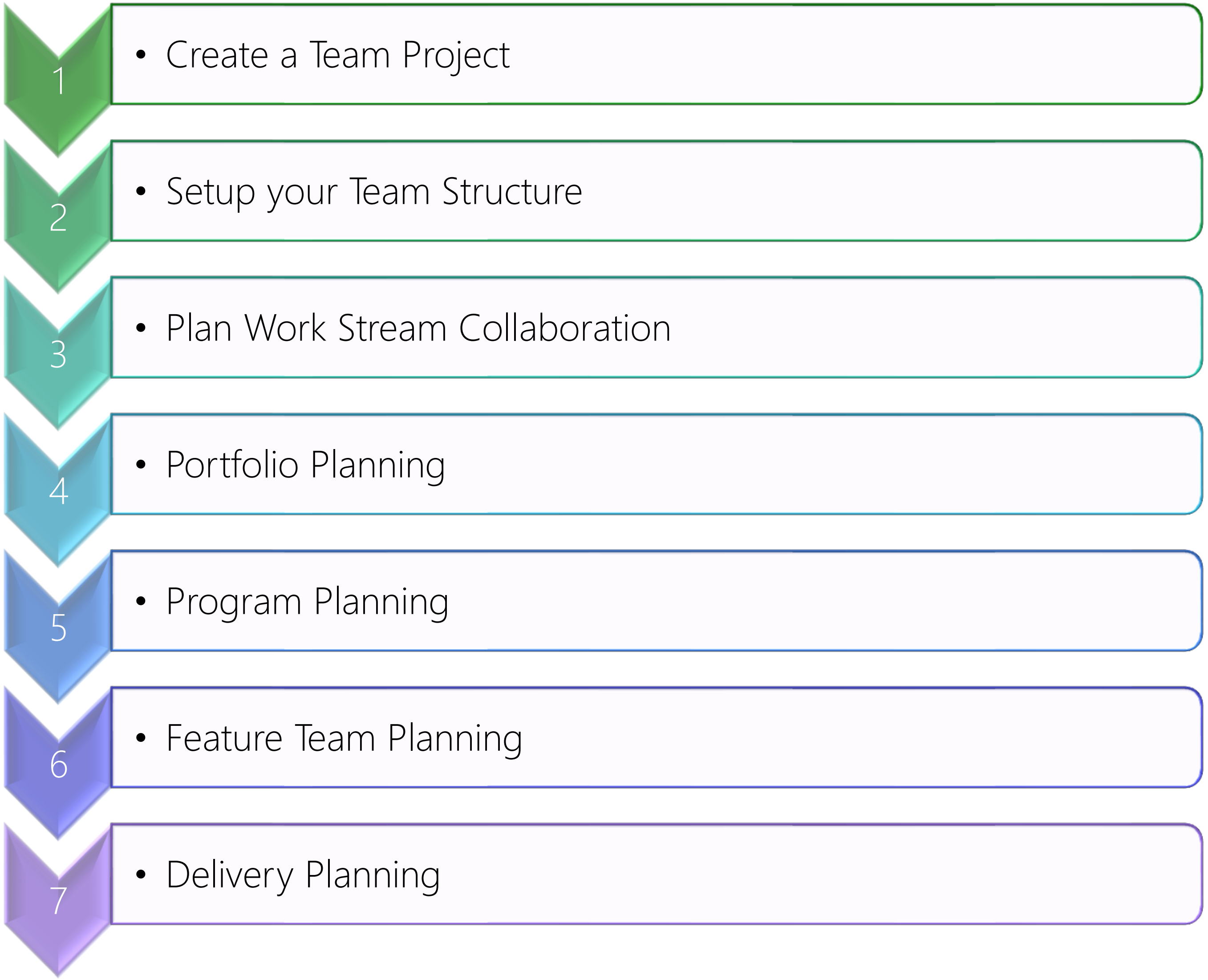


Team of 10 Engineers

Demonstration: Agile at Scale using Azure DevOps Services

We will use Azure DevOps Services to showcase how your organization can scale agile practices.

Demonstration Review



1
- Create a Team Project

2
- Setup your Team Structure

3
- Plan Work Stream Collaboration

4
- Portfolio Planning

5
- Program Planning

6
- Feature Team Planning

7
- Delivery Planning

Module 1: Planning

Lab 1: Plan – Agile at Scale

Exercise 1: Create a Team Project
Exercise 2: Setup your Team Structure
Exercise 3: Plan Work Streams Collaboration Flow
Exercise 4: Portfolio Backlog Planning
Exercise 5: Program Team Planning
Exercise 6: Feature Team Planning
Exercise 7: Delivery Planning

Lab Time: 120 minutes (about 2 hours)

Knowledge Check

Question #1: Why is Agile planning important in DevOps?

We actively respond to change and continuous development rather than follow a big up-front master plan.

Question #2: How do you measure progress?

By the value delivered, the quality, and security of the product

Question #3: What is Cycle Time?

The amount of time to delivery a product. Think of it as the amount of time to create a thing, like a cup.

Microsoft