# Delivering Products Using Azure DevOps and Scrum

Accentient™

ADSvcs | 2021.06

# ACCENTIENT EDUCATION SERIES

**Committed to training success**

www.accentient.com

# Delivering Products Using Azure DevOps and Scrum

| | |
|---|---|
| **Course Number:** | **DPADS** |
| **Version:** | **2021.6** |
| **Software version:** | **AzDOSvc VS 16.10** |

## Delivering Products Using Azure DevOps and Scrum

# Course Introduction

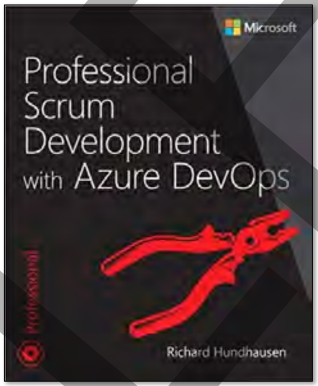
## Accentient

- A leader in ALM, DevOps, and Scrum knowledge
- Helped thousands of teams and individuals understand and implement Azure DevOps/VSTS/TFS and Scrum successfully
- Has a close working relationship with Microsoft
- Course creator and steward for Scrum.org
- Has trainers that are Microsoft MVPs, Professional Scrum Developers, Professional Scrum Trainers, and authors

www.accentient.com | @accentient

## Course Creator: Richard Hundhausen

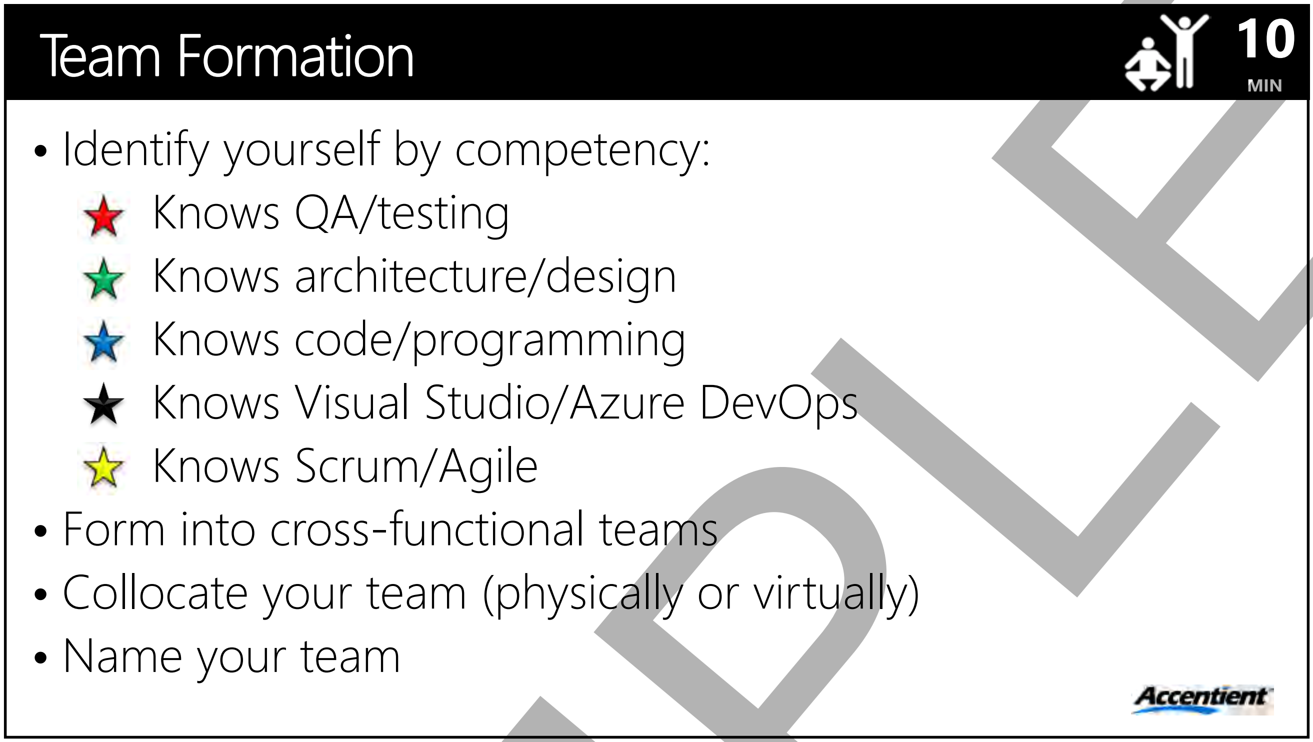
- President of Accentient
- Author of software development books
- First Microsoft TFS/ALM/DevOps MVP
- Professional Scrum Developer
- Professional Scrum Trainer
- Co-creator of Nexus scaled Scrum Fx
- richard@accentient.com
- @rhundhausen



http://bit.ly/PSDAzDo

## Prerequisites

- Familiar with software development lifecycle
- Familiar with team-based development
  - Product Owner, Scrum Master, or Developer
- Familiar with the Scrum framework
- Have used a modern version of Visual Studio

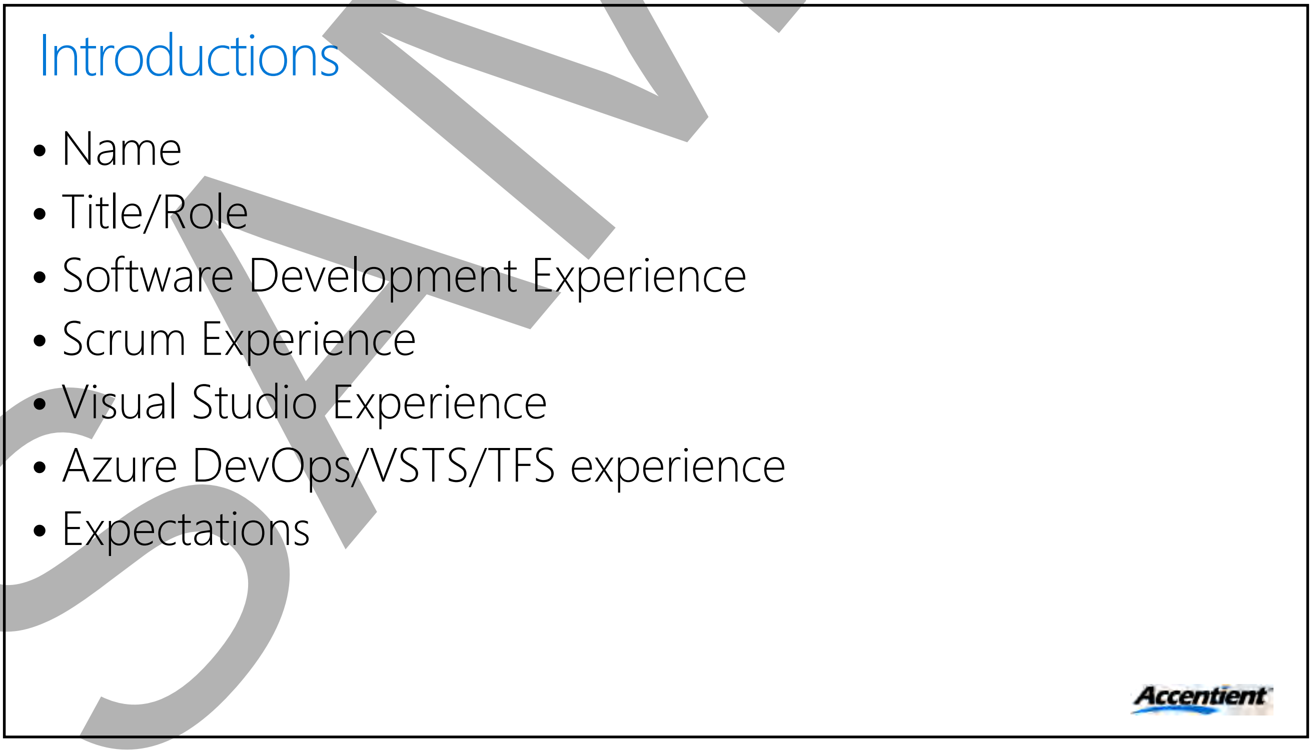


## Team Formation

10 MIN

- Identify yourself by competency:
  - ★ Knows QA/testing
  - ★ Knows architecture/design
  - ★ Knows code/programming
  - ★ Knows Visual Studio/Azure DevOps
  - ★ Knows Scrum/Agile
- Form into cross-functional teams
- Collocate your team (physically or virtually)
- Name your team

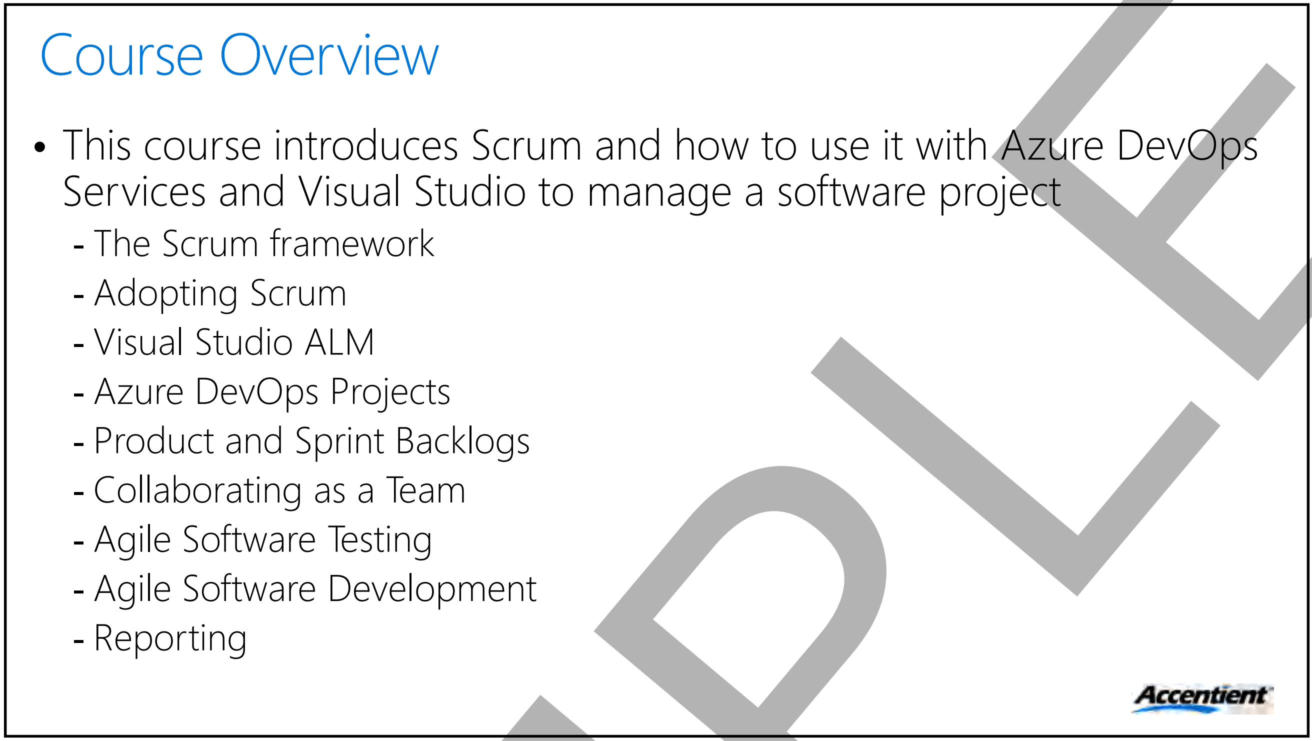
Accentient



## Introductions

- Name
- Title/Role
- Software Development Experience
- Scrum Experience
- Visual Studio Experience
- Azure DevOps/VSTS/TFS experience
- Expectations

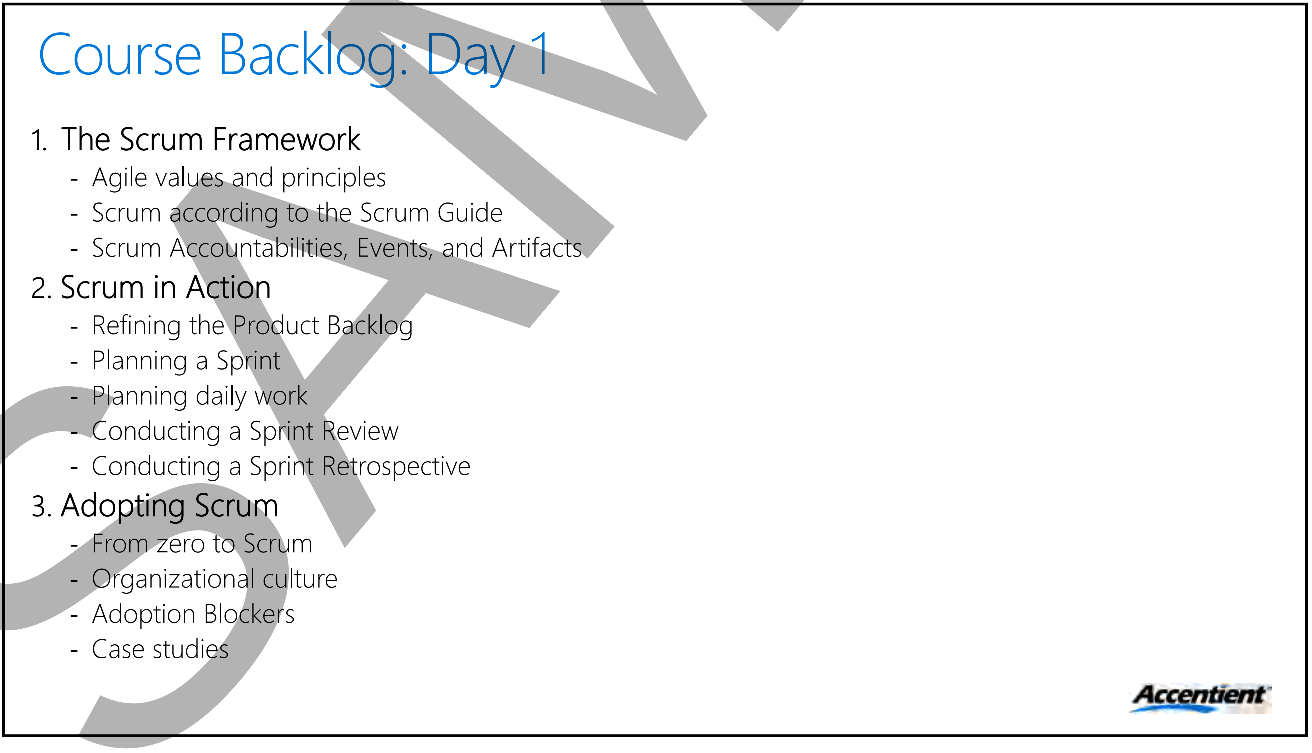
Accentient



## Course Overview

- This course introduces Scrum and how to use it with Azure DevOps Services and Visual Studio to manage a software project
  - The Scrum framework
  - Adopting Scrum
  - Visual Studio ALM
  - Azure DevOps Projects
  - Product and Sprint Backlogs
  - Collaborating as a Team
  - Agile Software Testing
  - Agile Software Development
  - Reporting

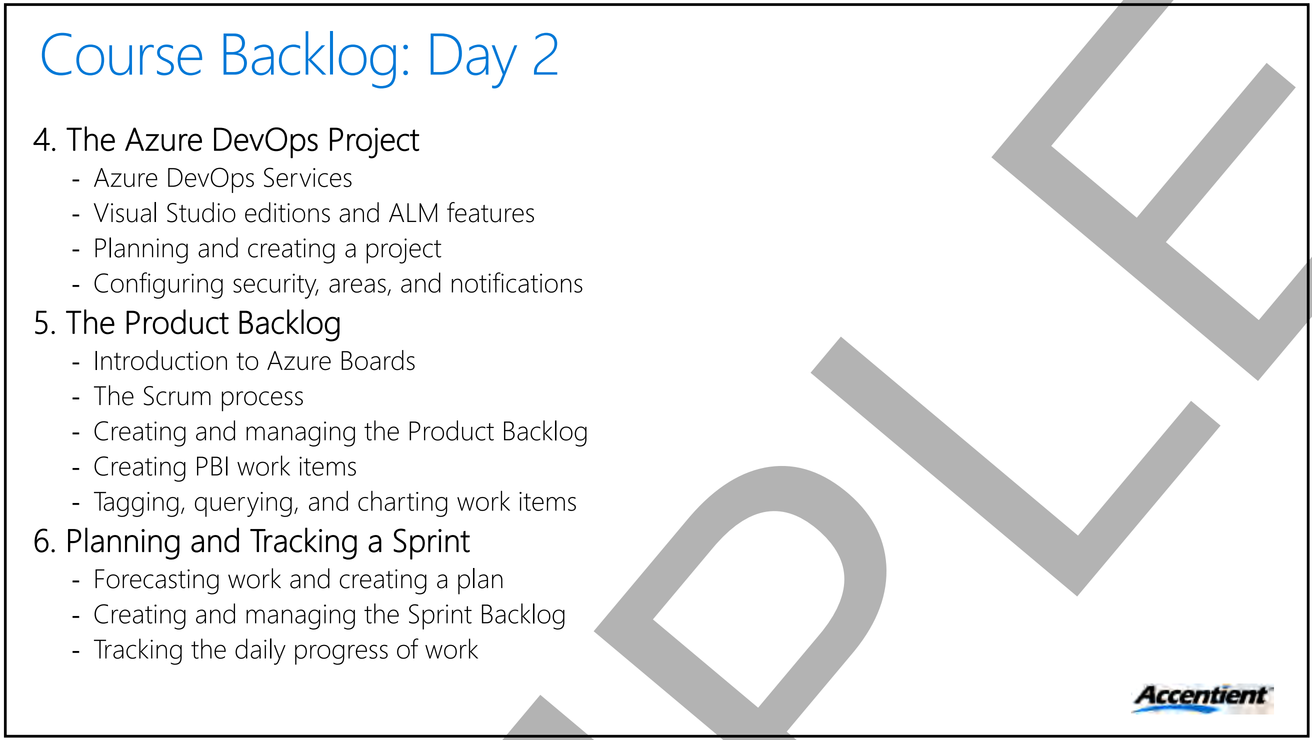
Accentient



## Course Backlog: Day 1

1. The Scrum Framework
   - Agile values and principles
   - Scrum according to the Scrum Guide
   - Scrum Accountabilities, Events, and Artifacts
2. Scrum in Action
   - Refining the Product Backlog
   - Planning a Sprint
   - Planning daily work
   - Conducting a Sprint Review
   - Conducting a Sprint Retrospective
3. Adopting Scrum
   - From zero to Scrum
   - Organizational culture
   - Adoption Blockers
   - Case studies

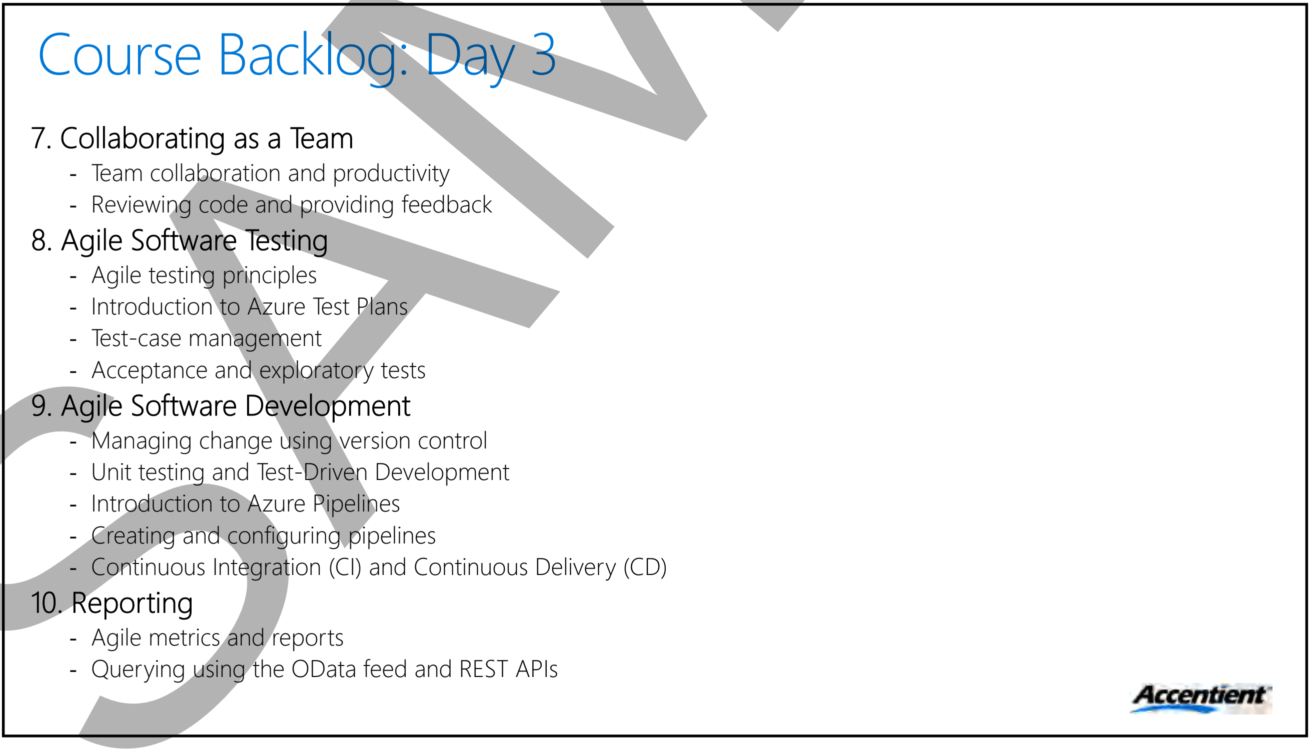
Accentient



## Course Backlog: Day 2

4. The Azure DevOps Project
   - Azure DevOps Services
   - Visual Studio editions and ALM features
   - Planning and creating a project
   - Configuring security, areas, and notifications
5. The Product Backlog
   - Introduction to Azure Boards
   - The Scrum process
   - Creating and managing the Product Backlog
   - Creating PBI work items
   - Tagging, querying, and charting work items
6. Planning and Tracking a Sprint
   - Forecasting work and creating a plan
   - Creating and managing the Sprint Backlog
   - Tracking the daily progress of work

Accentient



## Course Backlog: Day 3

7. Collaborating as a Team
   - Team collaboration and productivity
   - Reviewing code and providing feedback
8. Agile Software Testing
   - Agile testing principles
   - Introduction to Azure Test Plans
   - Test-case management
   - Acceptance and exploratory tests
9. Agile Software Development
   - Managing change using version control
   - Unit testing and Test-Driven Development
   - Introduction to Azure Pipelines
   - Creating and configuring pipelines
   - Continuous Integration (CI) and Continuous Delivery (CD)
10. Reporting
   - Agile metrics and reports
   - Querying using the OData feed and REST APIs

Accentient

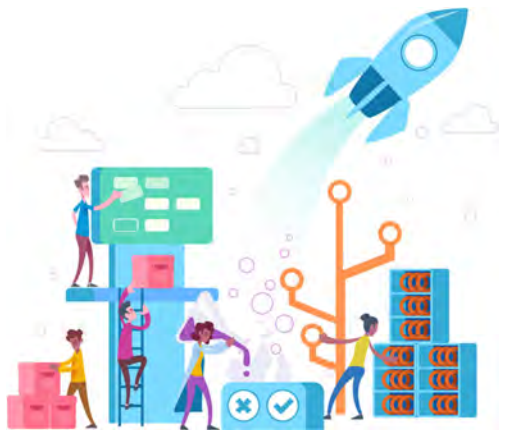
## Our Desktop Environment

- Visual Studio 2019 Enterprise Edition
  - All features installed
  - Current updates
- SQL Server 2016 (or newer)
  - Any edition, including Express/LocalDB
- Microsoft Office 2016 Professional (or newer)

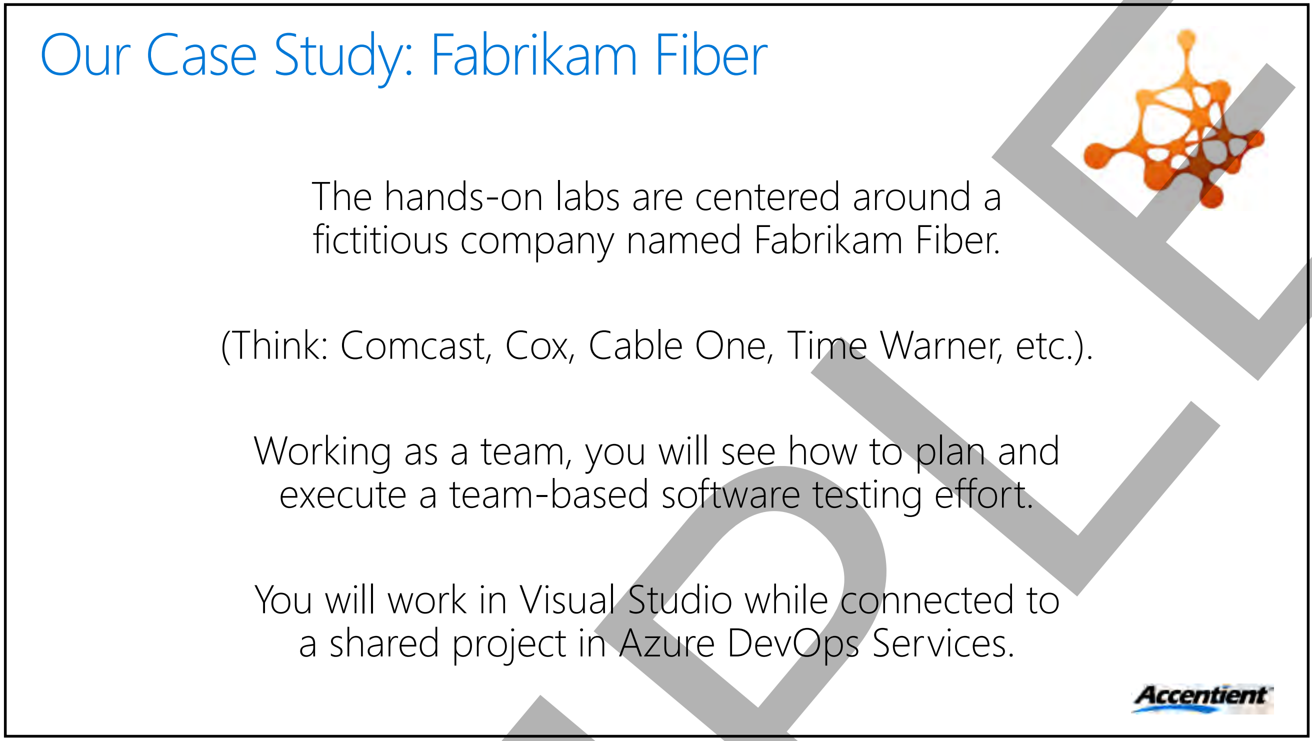
Accentient

## Our Azure DevOps Services Environment

- We will be using a shared instance of Azure DevOps Services
- Each team will ...
  - Be collocated (physically or virtually)
  - Have its own Azure DevOps project
  - Collaborate on all work in this class
- Each team member will ...
  - Need a Microsoft Account



Accentient



## Our Case Study: Fabrikam Fiber

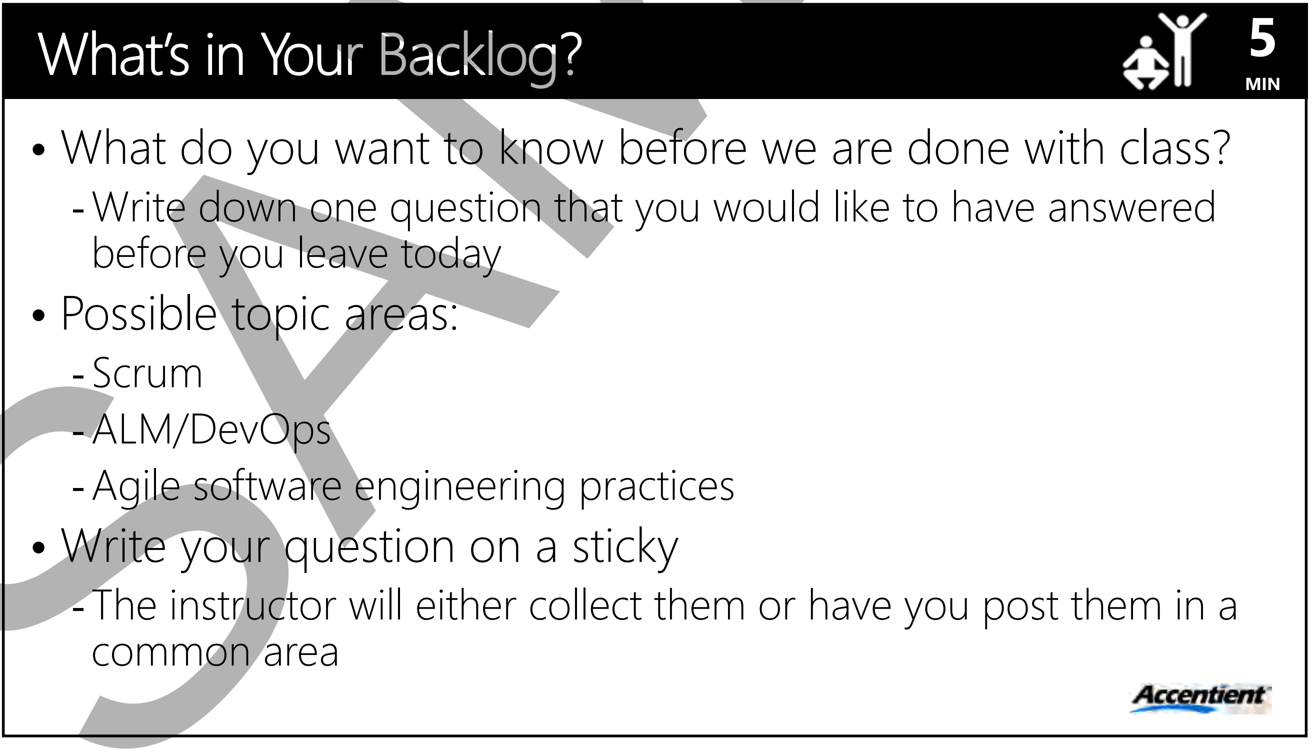
The hands-on labs are centered around a fictitious company named Fabrikam Fiber.

(Think: Comcast, Cox, Cable One, Time Warner, etc.).

Working as a team, you will see how to plan and execute a team-based software testing effort.

You will work in Visual Studio while connected to a shared project in Azure DevOps Services.

Accentient

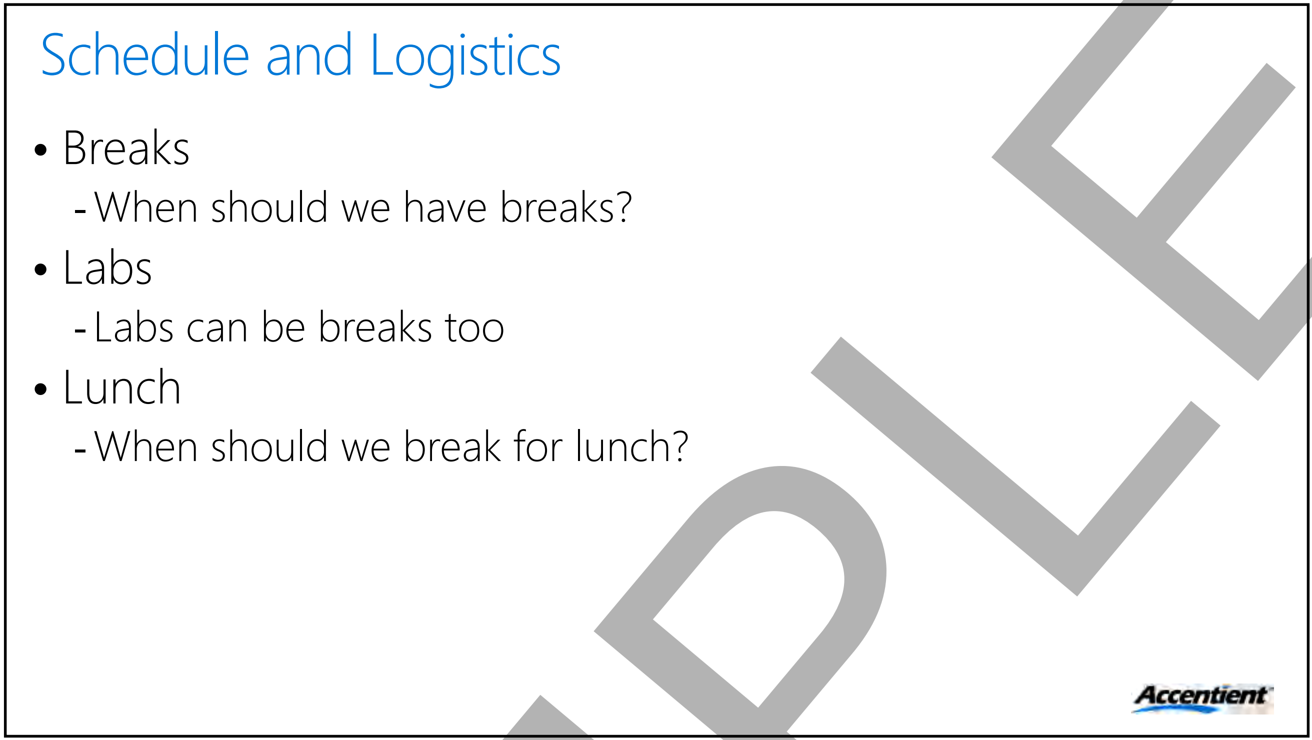


## What's in Your Backlog?

5 MIN

- What do you want to know before we are done with class?
  - Write down one question that you would like to have answered before you leave today
- Possible topic areas:
  - Scrum
  - ALM/DevOps
  - Agile software engineering practices
- Write your question on a sticky
  - The instructor will either collect them or have you post them in a common area

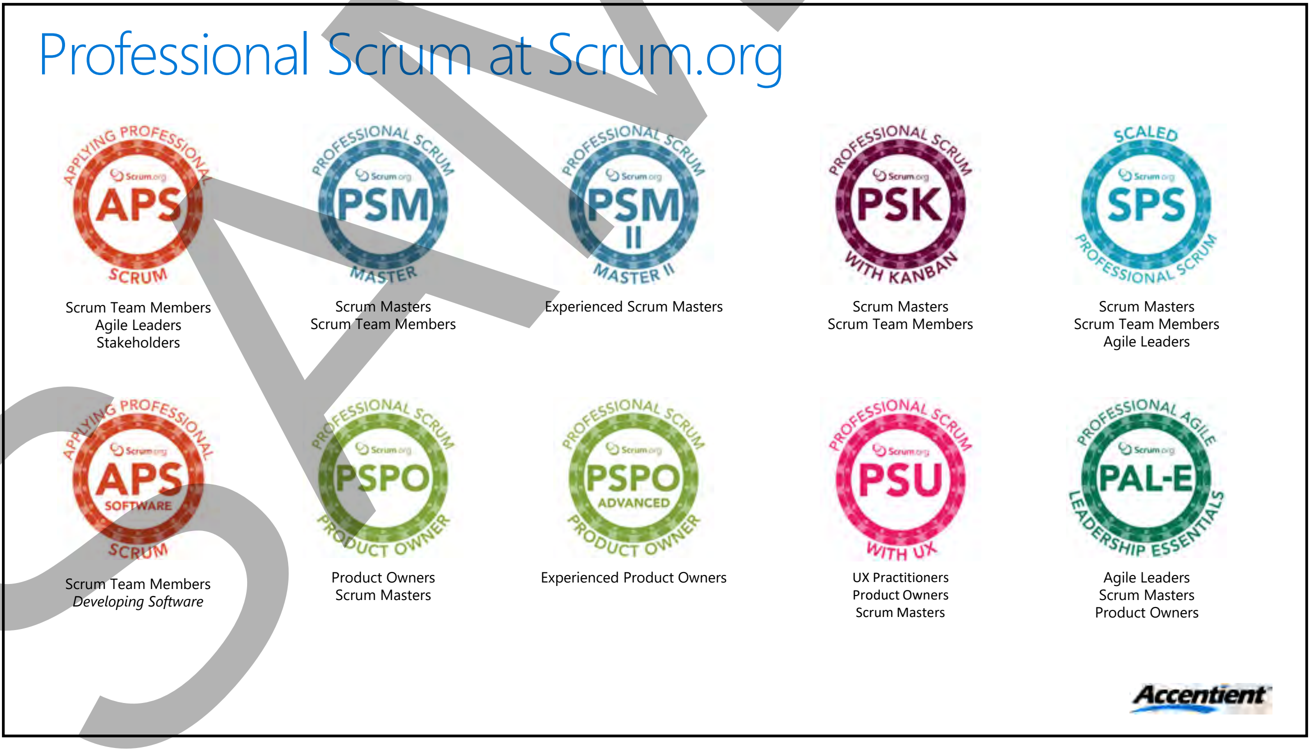
Accentient



## Schedule and Logistics

- Breaks
  - When should we have breaks?
- Labs
  - Labs can be breaks too
- Lunch
  - When should we break for lunch?

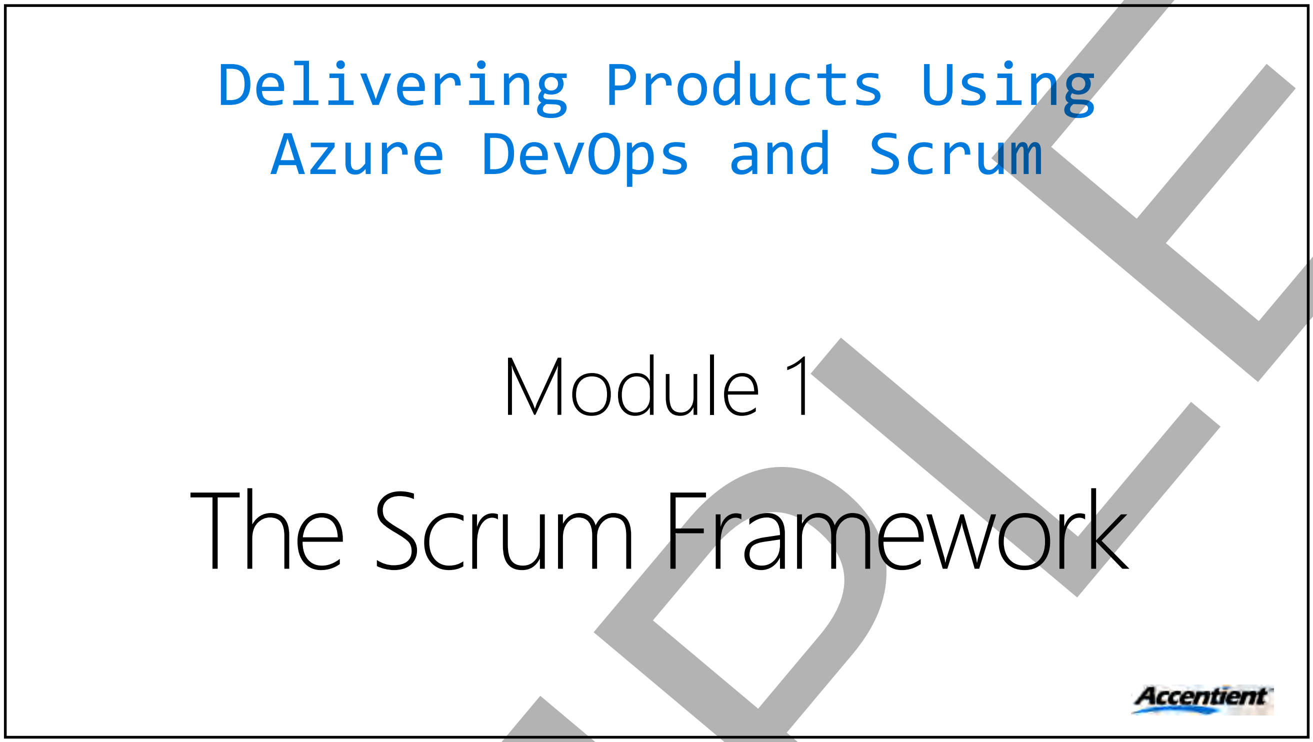
Accentient



## Professional Scrum at Scrum.org

| | | | | |
|---|---|---|---|---|
| APPLYING PROFESSIONAL SCRUM — APS | PROFESSIONAL SCRUM MASTER — PSM | PROFESSIONAL SCRUM MASTER — PSM II | PROFESSIONAL SCRUM WITH KANBAN — PSK | SCALED PROFESSIONAL SCRUM — SPS |
| Scrum Team Members<br>Agile Leaders<br>Stakeholders | Scrum Masters<br>Scrum Team Members | Experienced Scrum Masters | Scrum Masters<br>Scrum Team Members | Scrum Masters<br>Scrum Team Members<br>Agile Leaders |
| APPLYING PROFESSIONAL SCRUM — APS SOFTWARE | PROFESSIONAL SCRUM PRODUCT OWNER — PSPO | PROFESSIONAL SCRUM PRODUCT OWNER — PSPO ADVANCED | PROFESSIONAL SCRUM WITH UX — PSU | PROFESSIONAL AGILE LEADERSHIP ESSENTIALS — PAL-E |
| Scrum Team Members<br>*Developing Software* | Product Owners<br>Scrum Masters | Experienced Product Owners | UX Practitioners<br>Product Owners<br>Scrum Masters | Agile Leaders<br>Scrum Masters<br>Product Owners |

Accentient



Delivering Products Using
Azure DevOps and Scrum

Module 1

# The Scrum Framework

## Module Backlog

- The Agile Manifesto
- The Scrum Framework
  - Accountabilities
  - Events
  - Artifacts
  - Commitments

## Why Scrum?
## First, Why Agile?

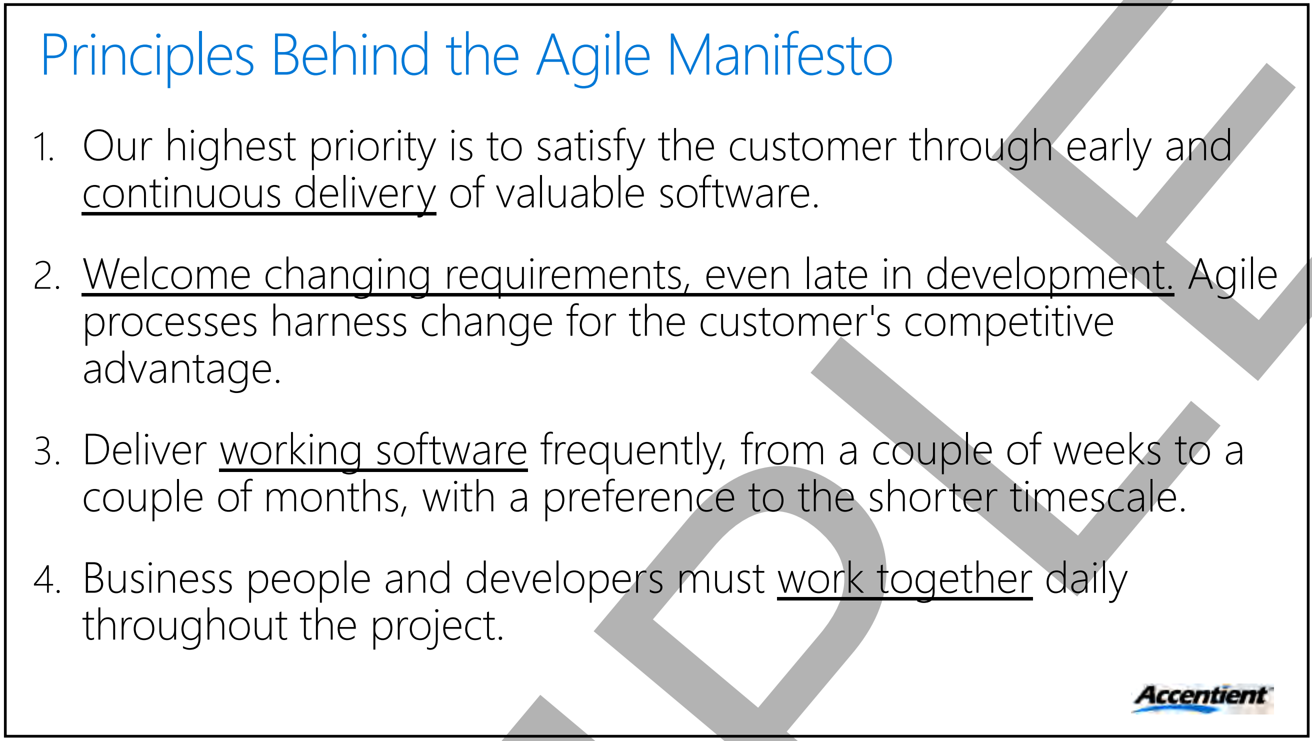
## The Agile Manifesto Says it All …

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

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

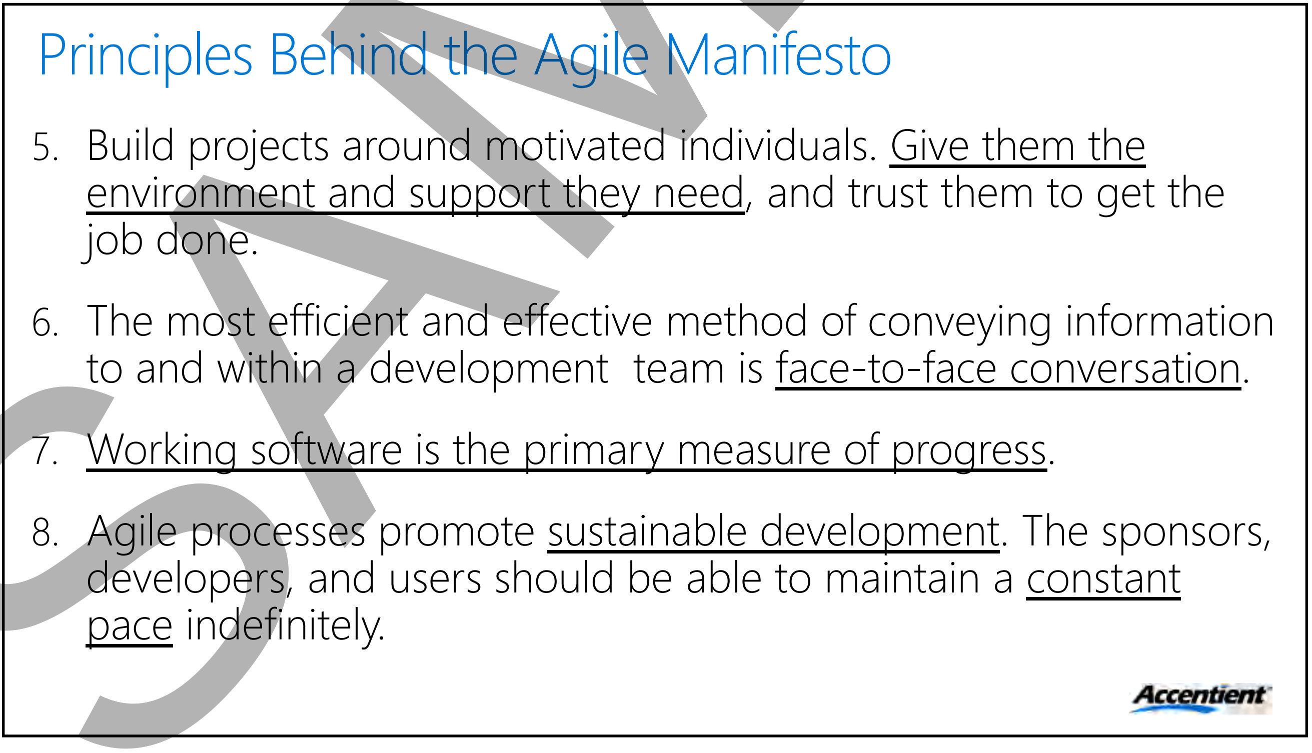
http://agilemanifesto.org



## Principles Behind 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 to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

Accentient



## Principles Behind the Agile Manifesto

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.

Accentient

## Principles Behind the Agile Manifesto

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.

## Agility == Adaptiveness

- Agility provides the capacity and capability of rapidly and efficiently adapting to change
- No silver bullet here, everyone involved must work and share in the risk to realize better, faster, cheaper
- Trivia: *Agile* software development was almost called *Adaptive* software development back in 2001

## How Would You Maintain 72 Degrees?

5 MIN

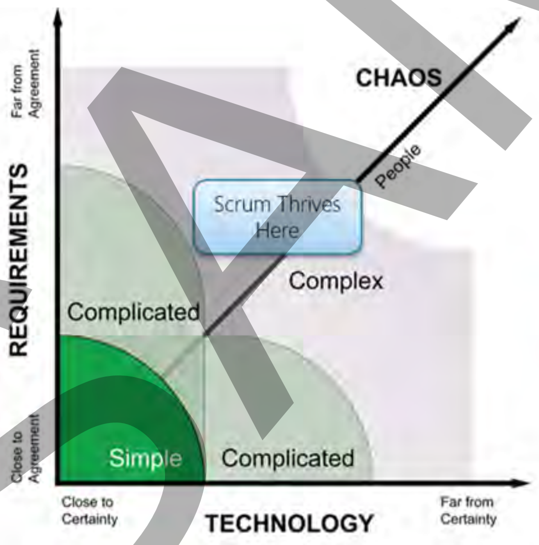
1. Inspect the room you are in right now
(Length, width, height, heating & cooling systems)

2. What are the variables you'd have to take into account in order to maintain a comfortable temperature of 68°F (20° C) in this room?

Tip: Knowledge of thermodynamics is helpful

Accentient

## The Complexity Of Software Development



**Simple**
Everything is known

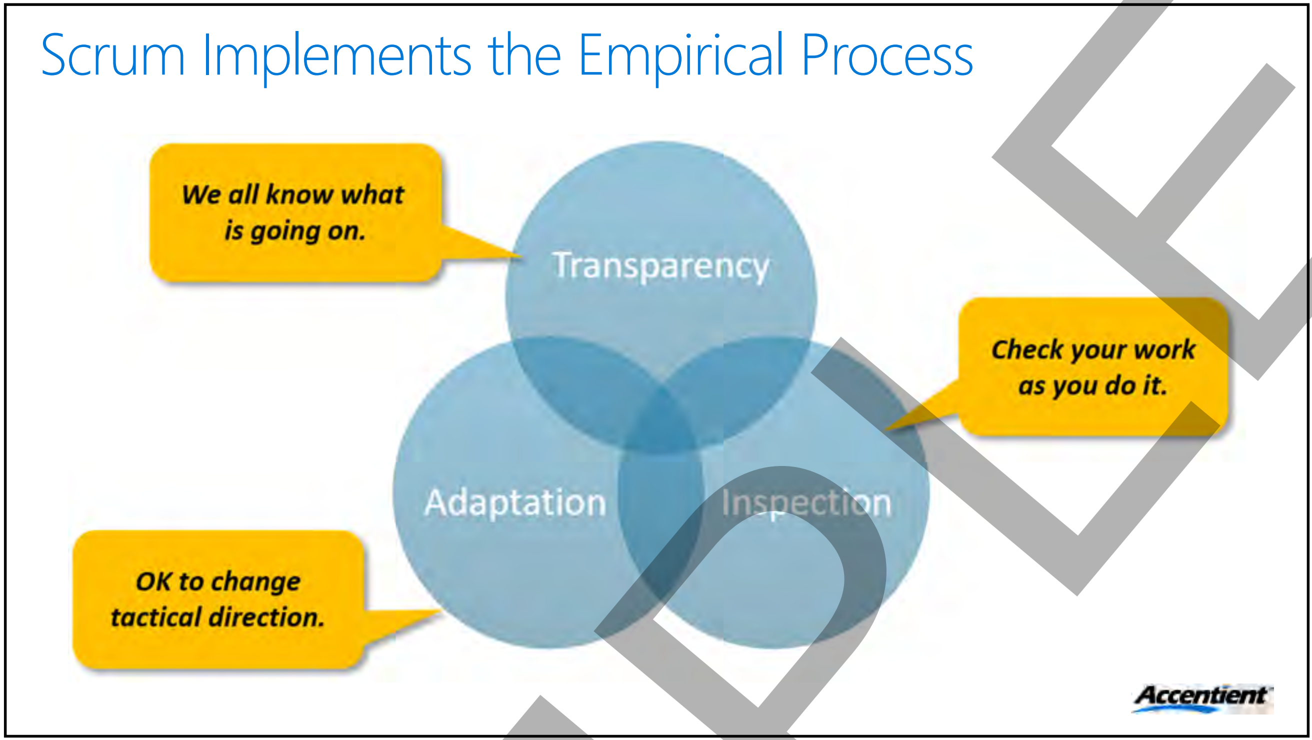
**Complicated**
More is known than unknown

**Complex**
More is unknown than known

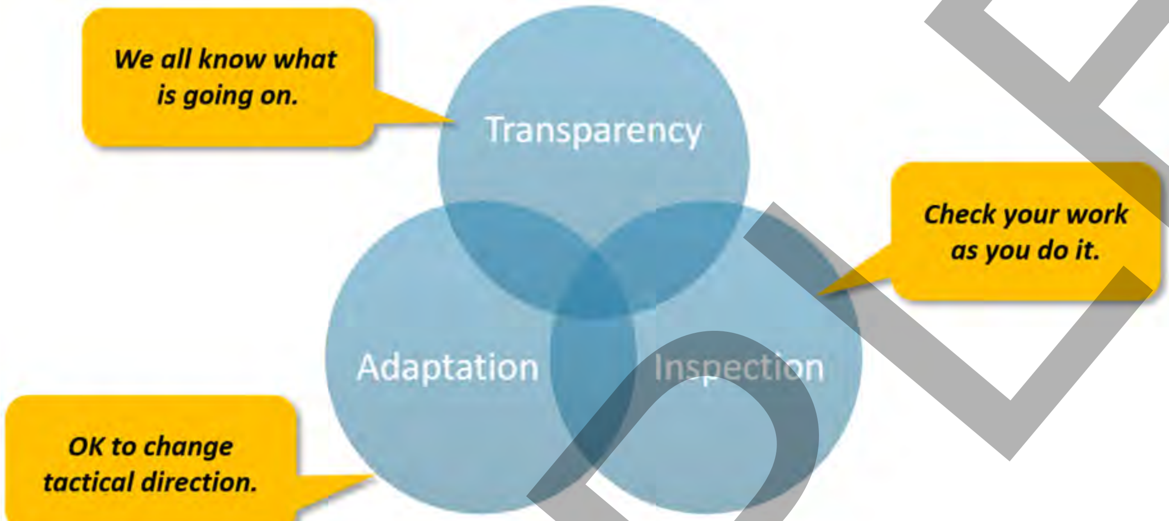
**Chaotic**
Very little is known

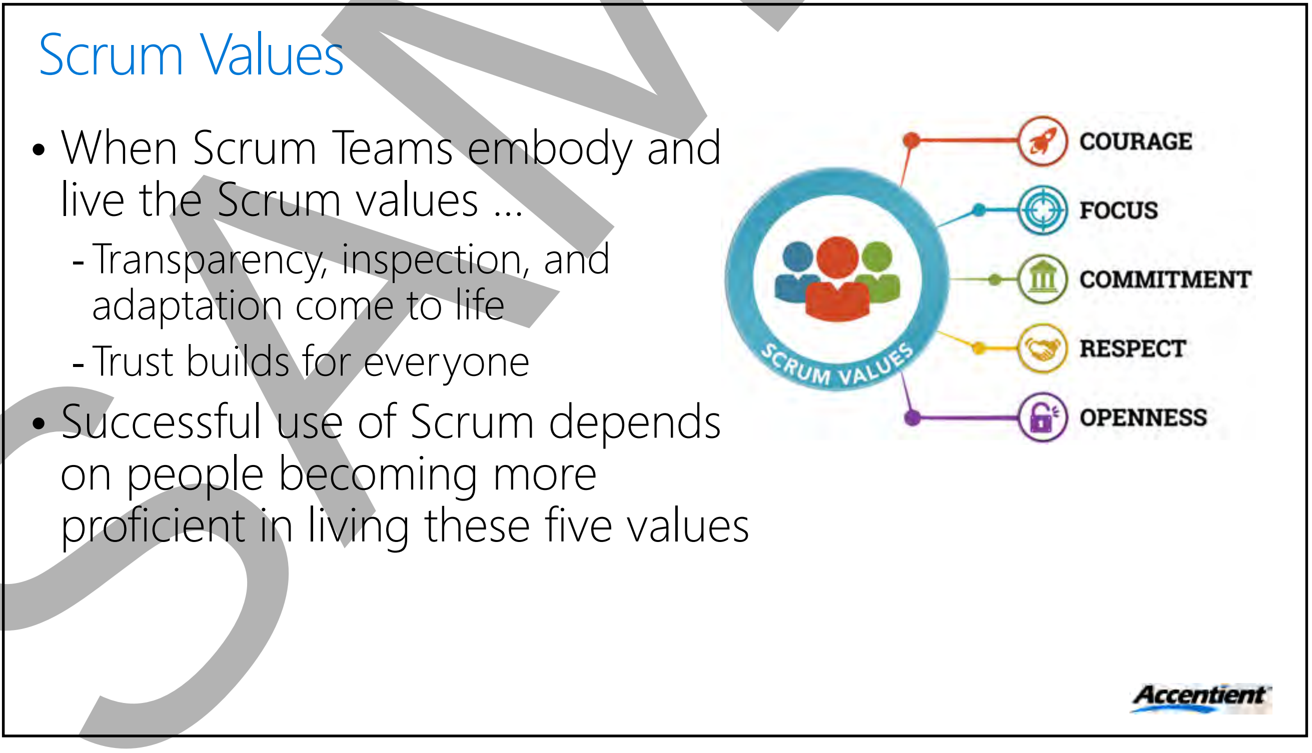
Source: Ralph Stacey, University of Hertfordshire

Accentient



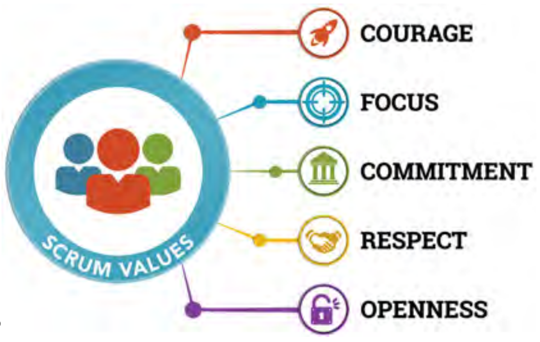
## Scrum Implements the Empirical Process


We all know what is going on.
Transparency
Check your work as you do it.
Adaptation
Inspection
OK to change tactical direction.



## Scrum Values

- When Scrum Teams embody and live the Scrum values ...
  - Transparency, inspection, and adaptation come to life
  - Trust builds for everyone
- Successful use of Scrum depends on people becoming more proficient in living these five values



## How Can You Enact the Scrum Values

10 MIN

As a team, list some practices or behaviors that embody each of the Scrum values:

Courage, Focus, Commitment, Respect, Openness

## Scrum is …

- A framework for developing and sustaining complex products
  - Example: Software development

## Scrum is Very Popular

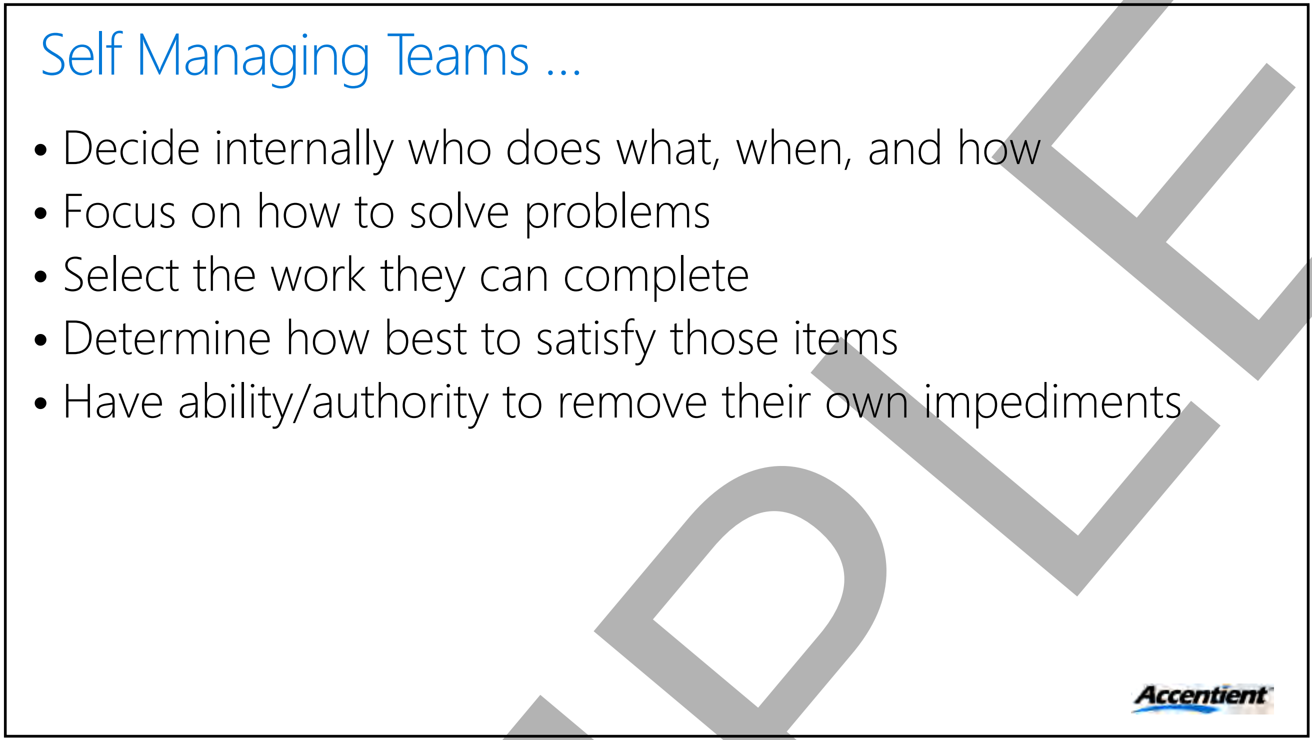
- It's the most popular way to be Agile today
- Scrum is not new
  - In use since the early 90s
  - Presented at OOPSLA 95
  - Scrum's birthday is celebrated every November

Accentient

## A Scrum Team ...

- Is accountable for creating a valuable, useful Increment every Sprint
- Commits to achieving its goals and to supporting each other
- Is expected to adapt the moment it learns anything new through inspection
- Is accountable for their success or failure
- Is self-managing

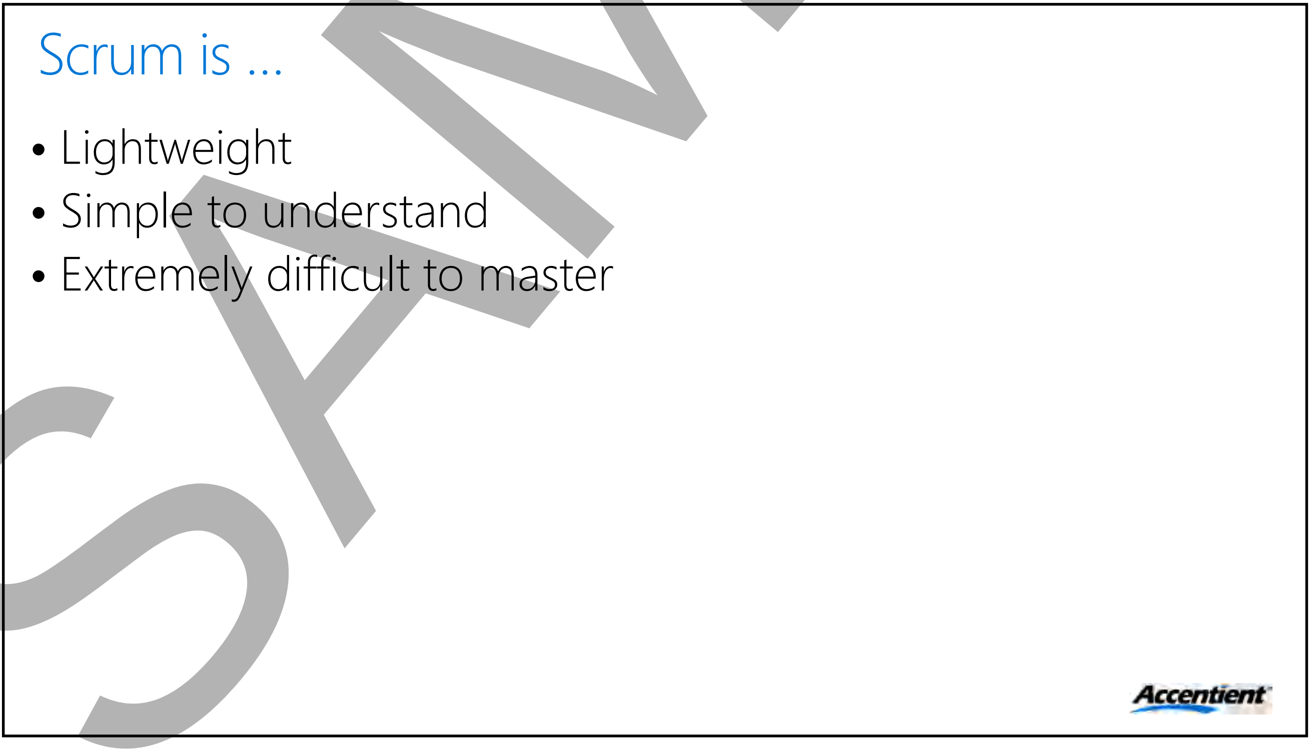
Accentient



## Self Managing Teams …

- Decide internally who does what, when, and how
- Focus on how to solve problems
- Select the work they can complete
- Determine how best to satisfy those items
- Have ability/authority to remove their own impediments

Accentient



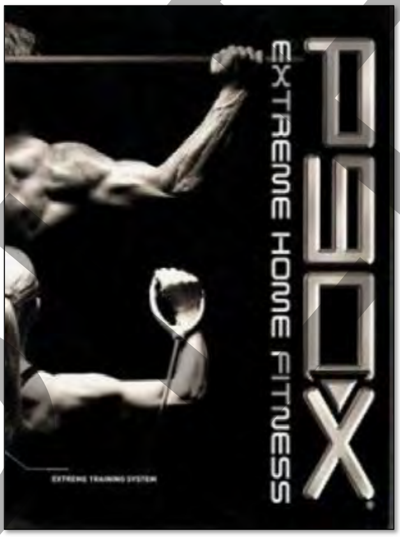
## Scrum is …

- Lightweight
- Simple to understand
- Extremely difficult to master

Accentient

## Scrum is to Agile …

- As P90X® is to "getting healthy"
- Both have core principles:
  - Eat better and exercise
- P90X® is a framework:
  - Daily menus and exercises
- In both cases, you are still the one that has to do it every day!



## So, Maybe Scrum Isn't for You

- Because implementing Scrum correctly will be difficult, don't consider it unless the pain of *not adopting* it is greater
- Only the courageous – motivated by insight, vision, and need – should consider using Scrum

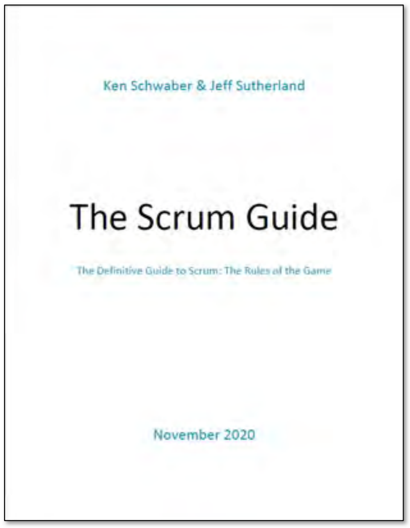
## The Scrum Framework

Accentient

## The Scrum Guide

- Documents the Scrum framework
  - Official rules of Scrum
- Maintained by Ken Schwaber and Jeff Sutherland
- Available at:

  http://scrumguides.org



Ken Schwaber & Jeff Sutherland

The Scrum Guide

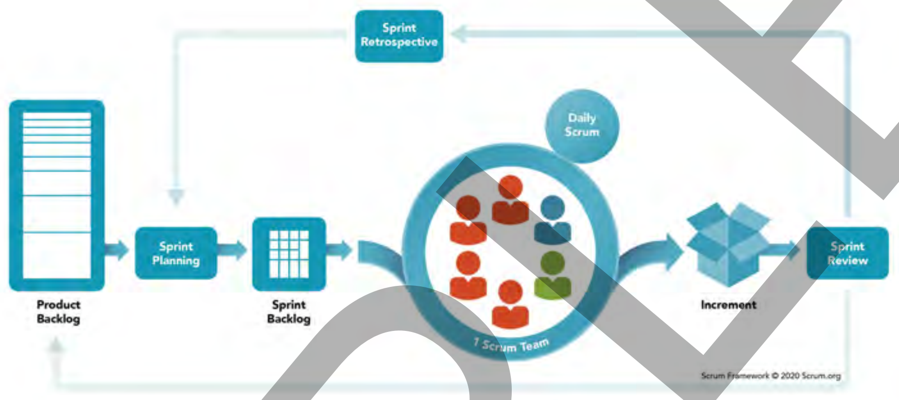
The Definitive Guide to Scrum: The Rules of the Game

November 2020

Accentient

## The Scrum Framework

- Accountabilities
  - Product Owner
  - Developers
  - Scrum Master
- Events
  - Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective
- Artifacts
  - Product Backlog
  - Sprint Backlog
  - Increment
- Commitments
  - Product Goal
  - Sprint Goal
  - Definition of Done



Accentient

## Product Owner

Accountability

- The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team.
  - How this is done may vary widely across organizations, Scrum Teams, and individuals
  - The Product Owner is also accountable for effective Product Backlog management

Accentient

## Developers

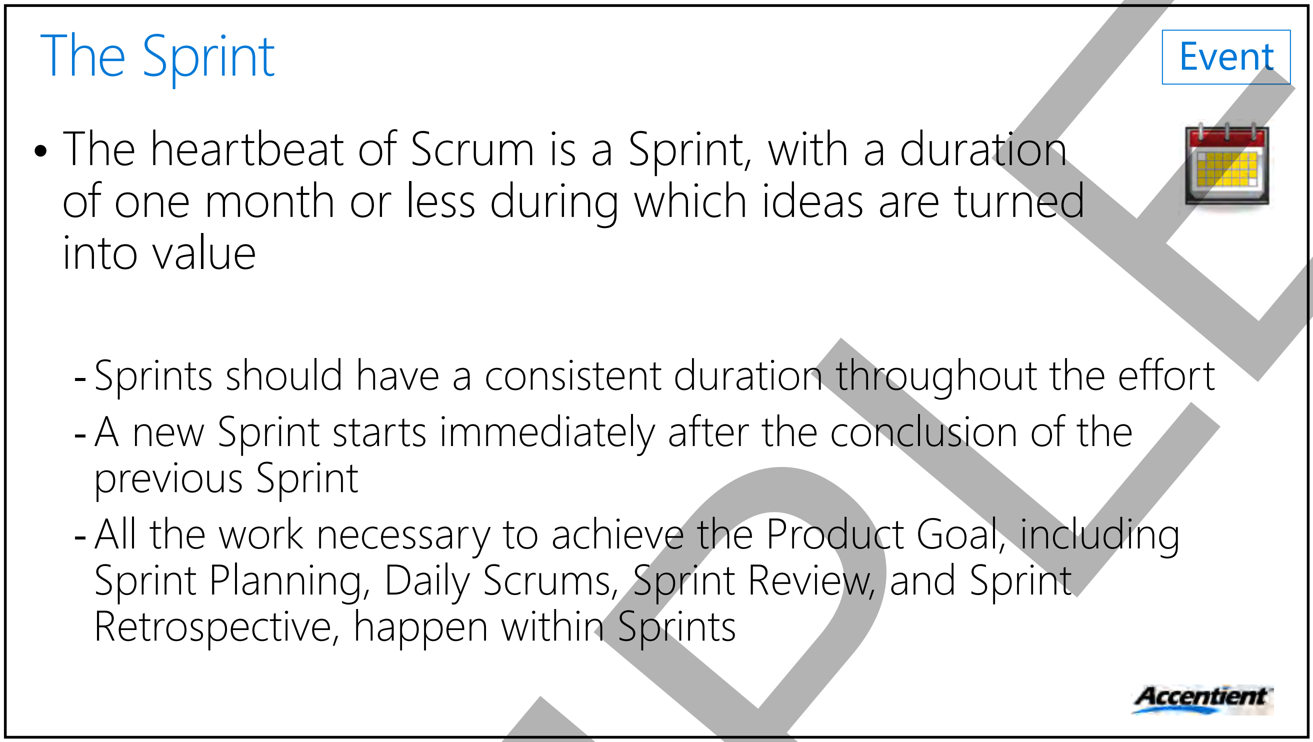
Accountability

- Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint
  - Only the Developers create the Increment

## Scrum Master

Accountability

- The Scrum Master is responsible for ensuring Scrum is understood and enacted
  - Scrum Masters do this by ensuring that the Scrum Team adheres to Scrum theory, practices, and rules
  - The Scrum Master is a servant-leader for the Scrum Team

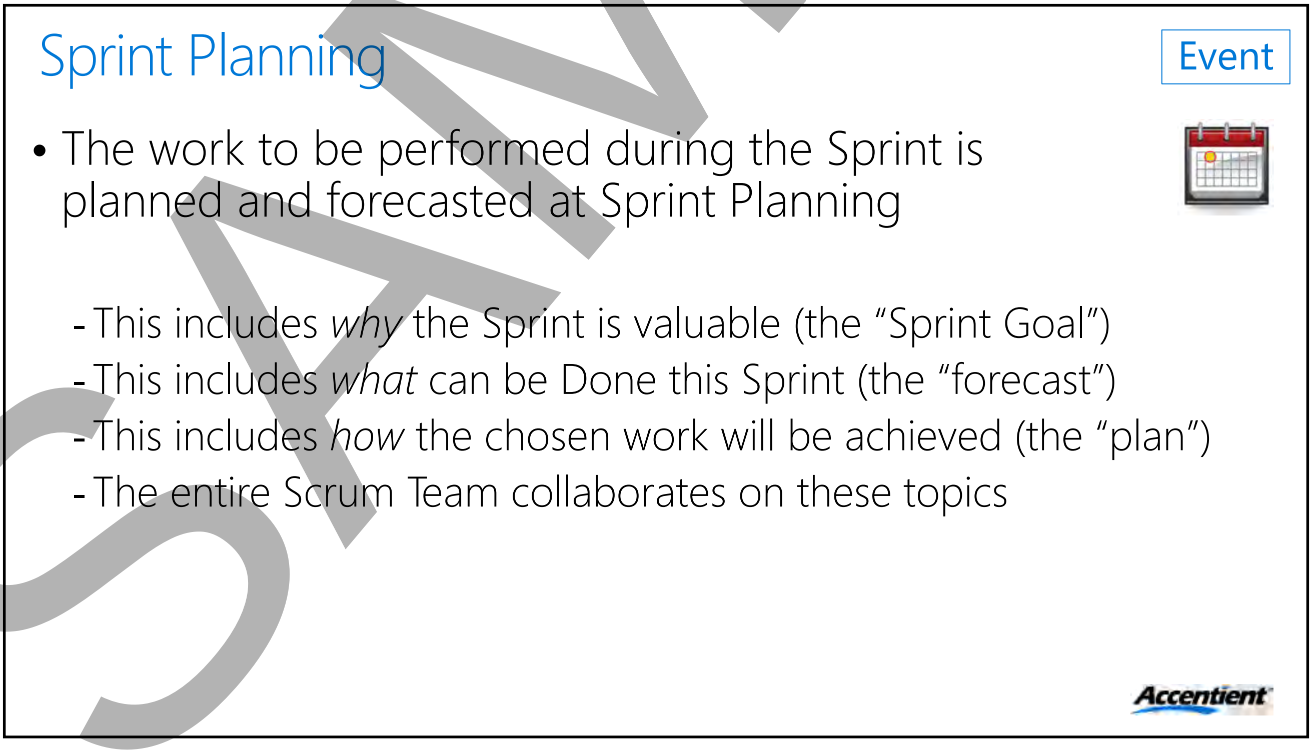


## The Sprint

Event

- The heartbeat of Scrum is a Sprint, with a duration of one month or less during which ideas are turned into value
  - Sprints should have a consistent duration throughout the effort
  - A new Sprint starts immediately after the conclusion of the previous Sprint
  - All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints

Accentient



## Sprint Planning

Event

- The work to be performed during the Sprint is planned and forecasted at Sprint Planning
  - This includes *why* the Sprint is valuable (the “Sprint Goal”)
  - This includes *what* can be Done this Sprint (the “forecast”)
  - This includes *how* the chosen work will be achieved (the “plan”)
  - The entire Scrum Team collaborates on these topics

Accentient

## Daily Scrum

Event

- The Daily Scrum is a 15-minute time-boxed to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.
  - This is done by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one
  - The Daily Scrum is held at the same time and same place each day to reduce complexity

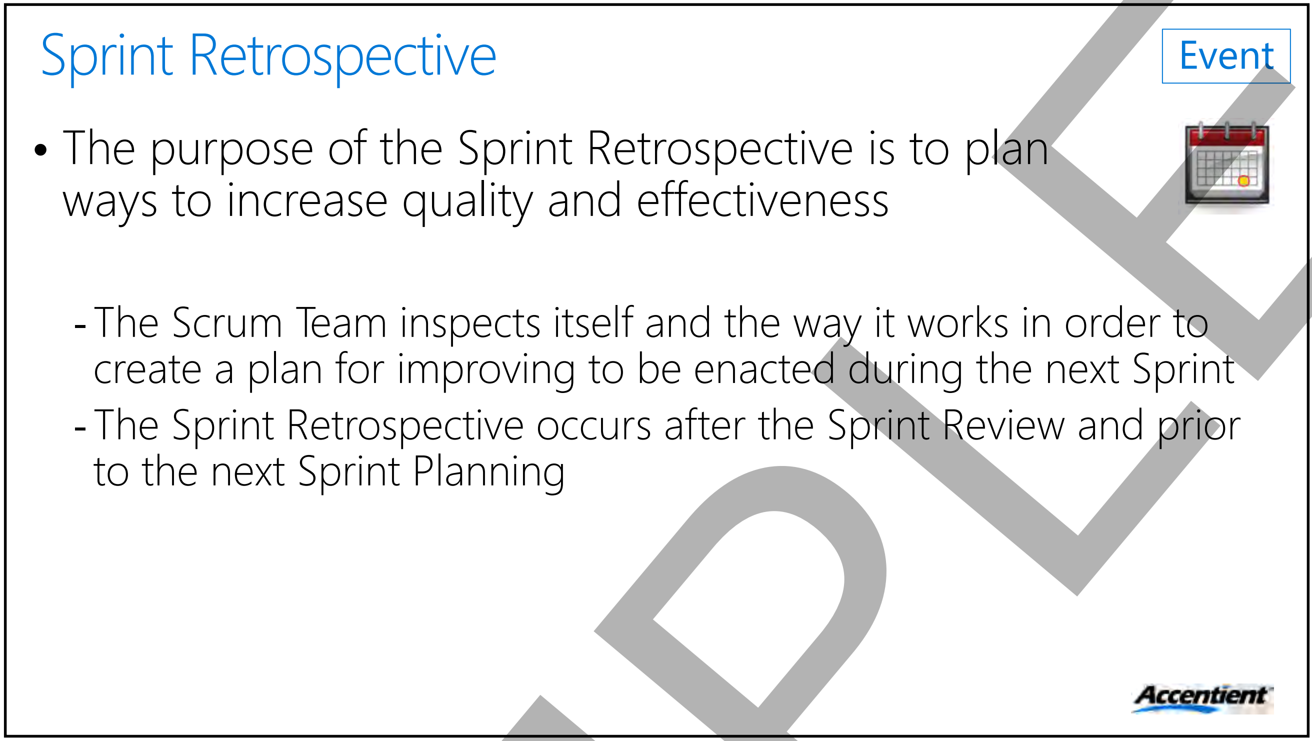
Accentient

## Sprint Review

Event

- A Sprint Review is held at the end of the Sprint for stakeholders to inspect the Increment and adapt the Product Backlog as needed
  - During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint as well as progress toward the Product Goal
  - Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on what could be done in upcoming Sprints to further optimize value

Accentient

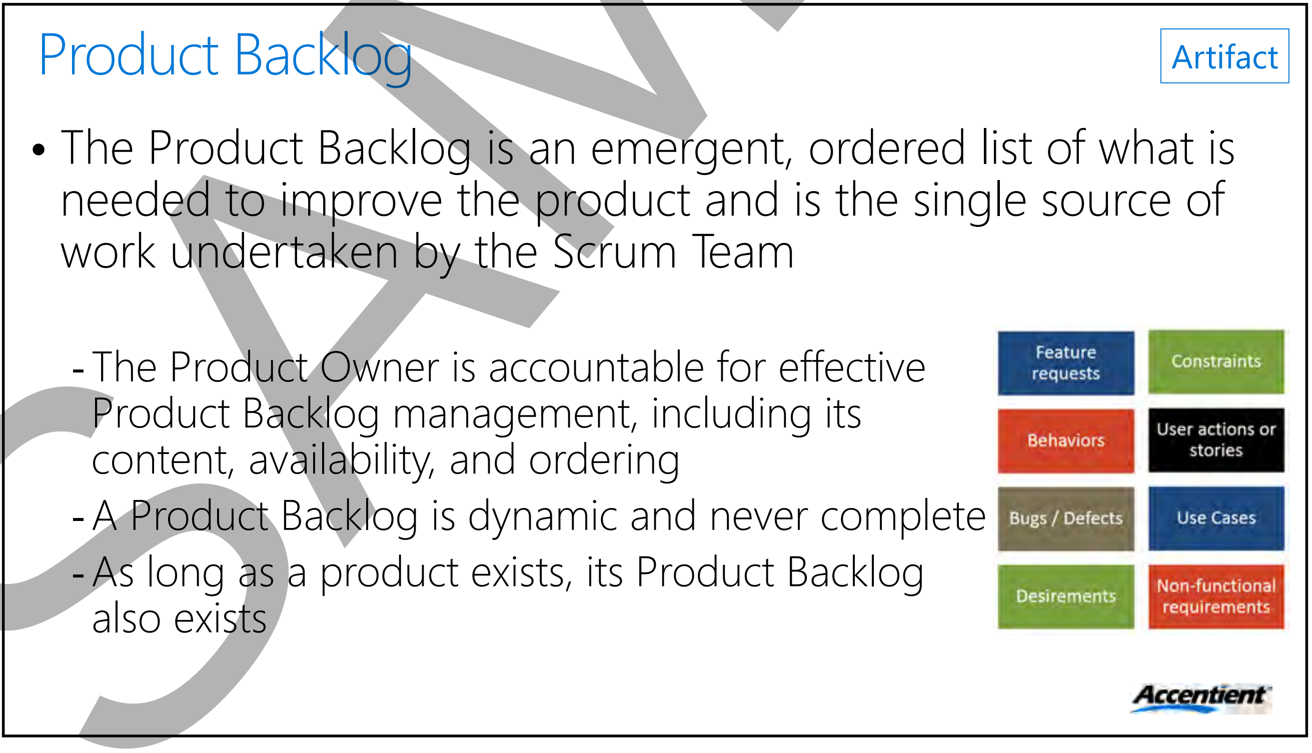


## Sprint Retrospective

Event

- The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness
  - The Scrum Team inspects itself and the way it works in order to create a plan for improving to be enacted during the next Sprint
  - The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning

Accentient



## Product Backlog

Artifact

- The Product Backlog is an emergent, ordered list of what is needed to improve the product and is the single source of work undertaken by the Scrum Team
  - The Product Owner is accountable for effective Product Backlog management, including its content, availability, and ordering
  - A Product Backlog is dynamic and never complete
  - As long as a product exists, its Product Backlog also exists

| | |
|---|---|
| Feature requests | Constraints |
| Behaviors | User actions or stories |
| Bugs / Defects | Use Cases |
| Desirements | Non-functional requirements |

Accentient

## Sprint Backlog

Artifact

- The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), and an actionable plan for delivering the Increment (how)
  - The Sprint Backlog is a forecast by the Developers about what functionality will be in the next Increment and the work needed to deliver that functionality as a Done Increment
  - The Sprint Backlog makes visible all of the work that the Developers identify as necessary to meet the Sprint Goal

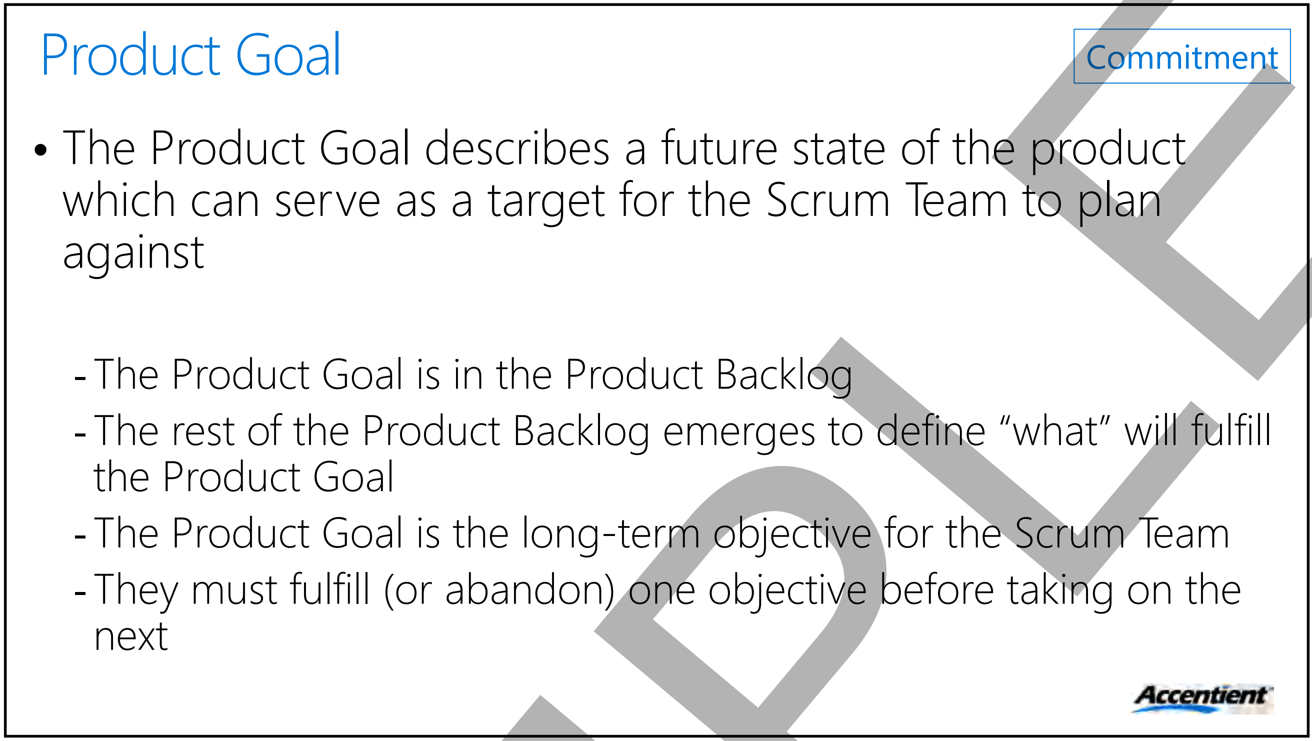
Accentient

## The Increment

Artifact

- An Increment is a concrete stepping stone toward the Product Goal
  - Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together
  - In order to provide value, the Increment must be usable

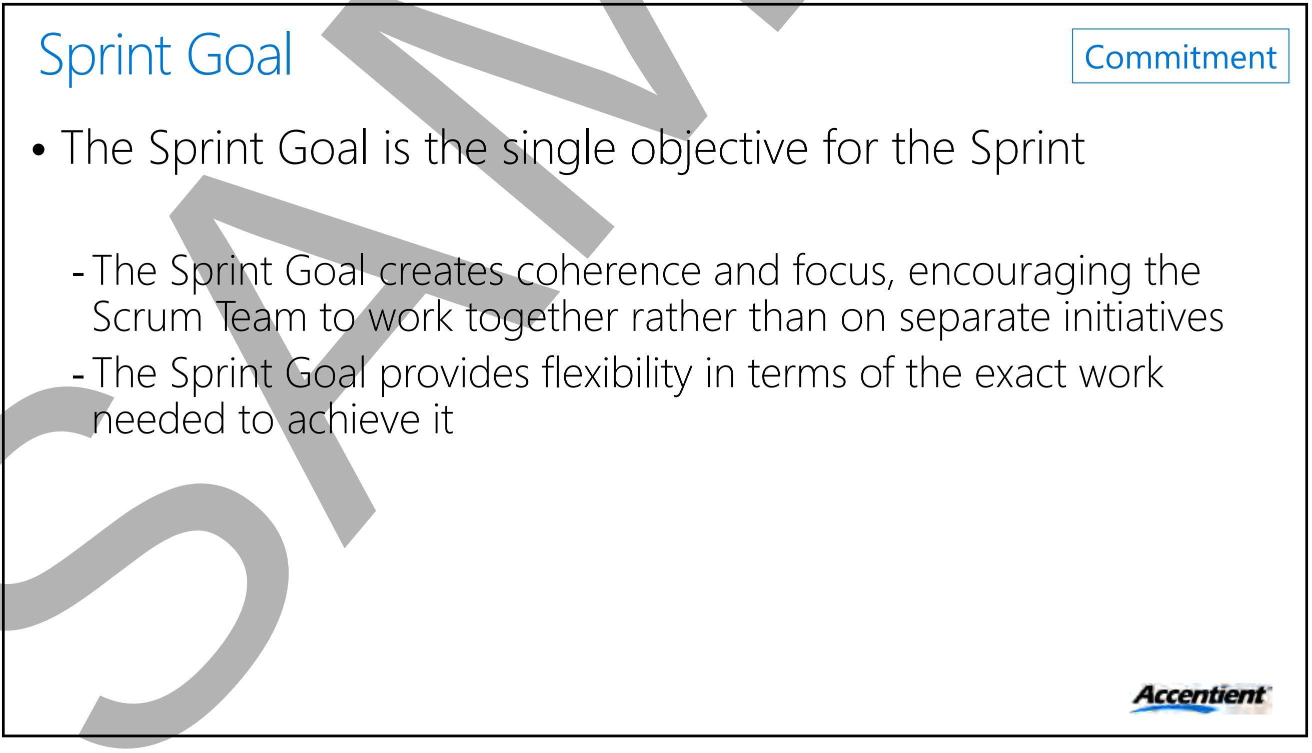
Accentient



## Product Goal

Commitment

- The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against
  - The Product Goal is in the Product Backlog
  - The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal
  - The Product Goal is the long-term objective for the Scrum Team
  - They must fulfill (or abandon) one objective before taking on the next



## Sprint Goal

Commitment

- The Sprint Goal is the single objective for the Sprint
  - The Sprint Goal creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives
  - The Sprint Goal provides flexibility in terms of the exact work needed to achieve it

## Definition of Done

Commitment

- The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product
  - Creates transparency by providing a shared understanding of what work was completed as part of the Increment
  - If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review
  - The moment a Product Backlog item meets the Definition of Done, an Increment is born

Accentient

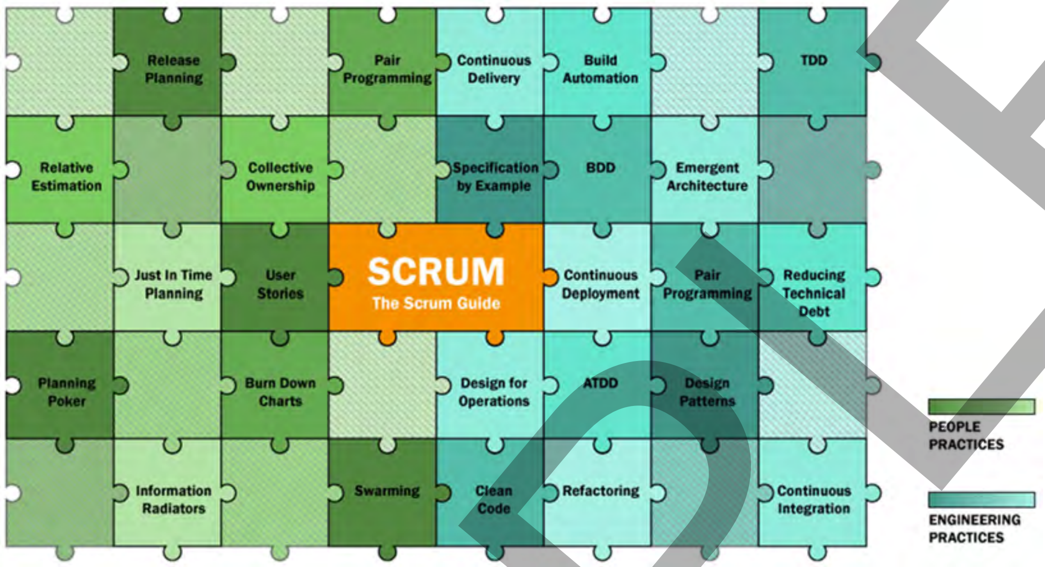
## So, Are You Currently Practicing Scrum?

5 MIN

✓ Are there a Product Owner, Scrum Master, and Developers?
✓ Does the Scrum Team have 10 or fewer members?
✓ Is the Scrum Team self-managing?
✓ Is there an ordered Product Backlog?
✓ Is there a Sprint Backlog that shows remaining work?
✓ Are Sprints fixed in length of 1 month or less?
✓ Is a daily plan created during the Daily Scrum?
✓ Is usable product created each Sprint?
✓ Can stakeholders inspect the product each Sprint?
✓ Is the Scrum Team actively improving its process each Sprint?

Accentient

## Complementary Practices

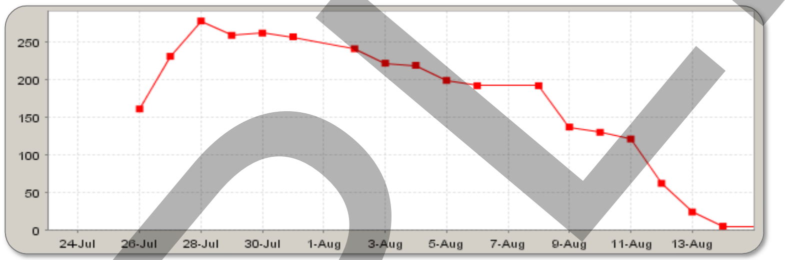


## Complementary Practice: Task Board

| PBI | To-Do | In-Progress | Done |
|---|---|---|---|
| 🟩 | | | 🟨 🟨 🟨 🟨 |
| 🟩 | 🟨 🟨 | 🟨 🟨 | 🟨 🟨 |
| 🟥 | 🟨 🟨 🟨 | | |
| 🟩 | 🟨 🟨 🟨 🟨 🟨 | | |

## Complementary Practice: Burndown Chart

- Burndown charts show a measure of work remaining over time
- Release burndowns show work remaining in a release
  - Based on # of PBIs or sum of their story points
- Sprint burndowns show work remaining in a Sprint
  - Typically based on hours remaining for the tasks
  - Could also be based on number of failing tests



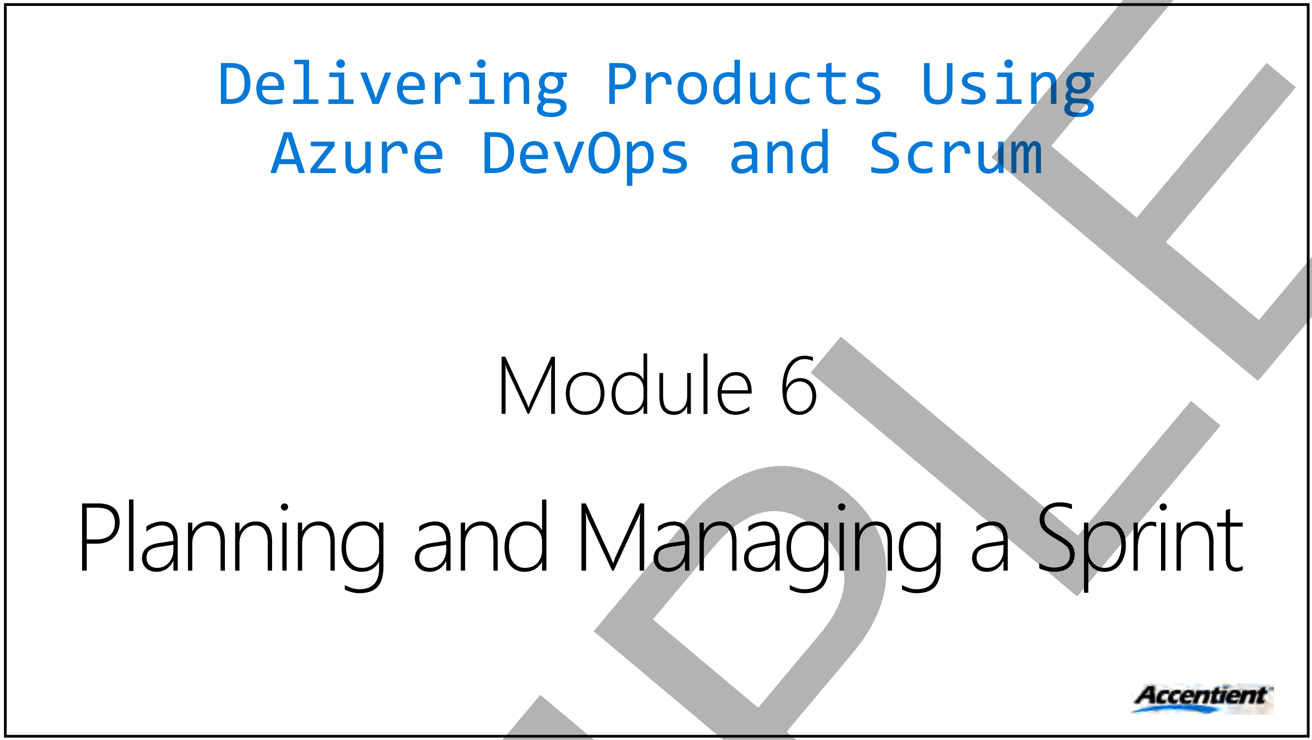
Accentient

## Module Retrospective

What have we learned in this module?

- The process wars are over – Agile won
  - And by Agile, we mean Scrum
- You can't *do* Agile, you can only *be* Agile
  - But you can do Scrum
- Scrum is over 25 years old
- The Scrum framework defines the accountabilities, events, artifacts, commitments, and the rules that bind them
  - Scrum is defined by the Scrum Guide

Accentient

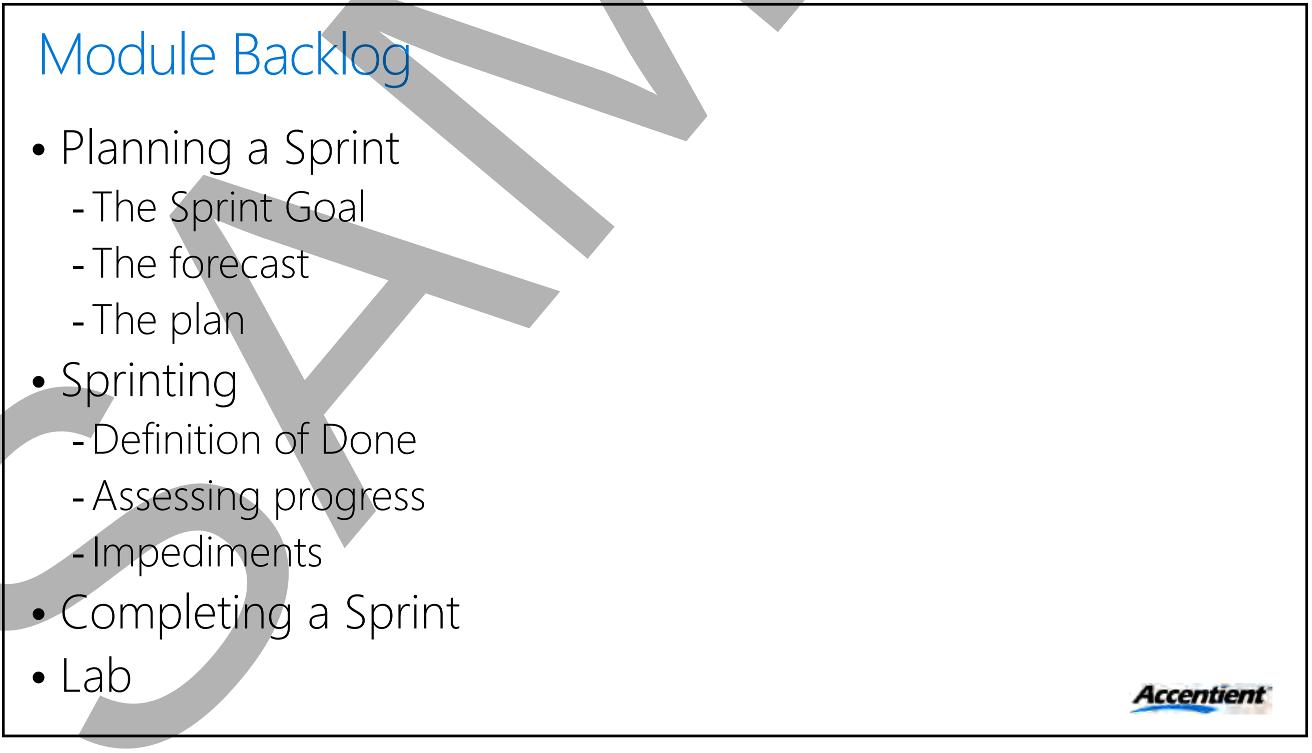


Delivering Products Using Azure DevOps and Scrum

# Module 6

# Planning and Managing a Sprint

Accentient



## Module Backlog

- Planning a Sprint
  - The Sprint Goal
  - The forecast
  - The plan
- Sprinting
  - Definition of Done
  - Assessing progress
  - Impediments
- Completing a Sprint
- Lab

Accentient

## Planning a Sprint

Accentient

## Sprint Planning

Event

The work to be performed in the Sprint is planned at Sprint Planning. The plan is created through the collaborative work of the entire Scrum Team

The Sprint Backlog contains the Sprint Goal, the set of Product Backlog items selected for the Sprint, plus a plan for delivering those items

Accentient

## Sprint Planning Inputs and Outputs

| Inputs | Outputs |
|---|---|
| The Increment | Sprint Goal |
| Definition of Done | The forecast |
| Past performance of the Developers (e.g. velocity) | The plan |
| Availability of the Developers (capacity) | |
| The refined Product Backlog | |
| Retrospective commitments | |

Accentient

## Setting Up The Sprint

✓ Ensure that the Sprint (iteration) has been created
- Make sure it has the correct start and end date
- Make sure it's visible to your team

✓ Collaborate on a Sprint Goal

✓ Forecast the PBIs that the team will deliver
1. Agree on a velocity to guide the forecast
2. Drag to the current Sprint in the *Planning* window
3. Set the State to *Committed* for each PBI

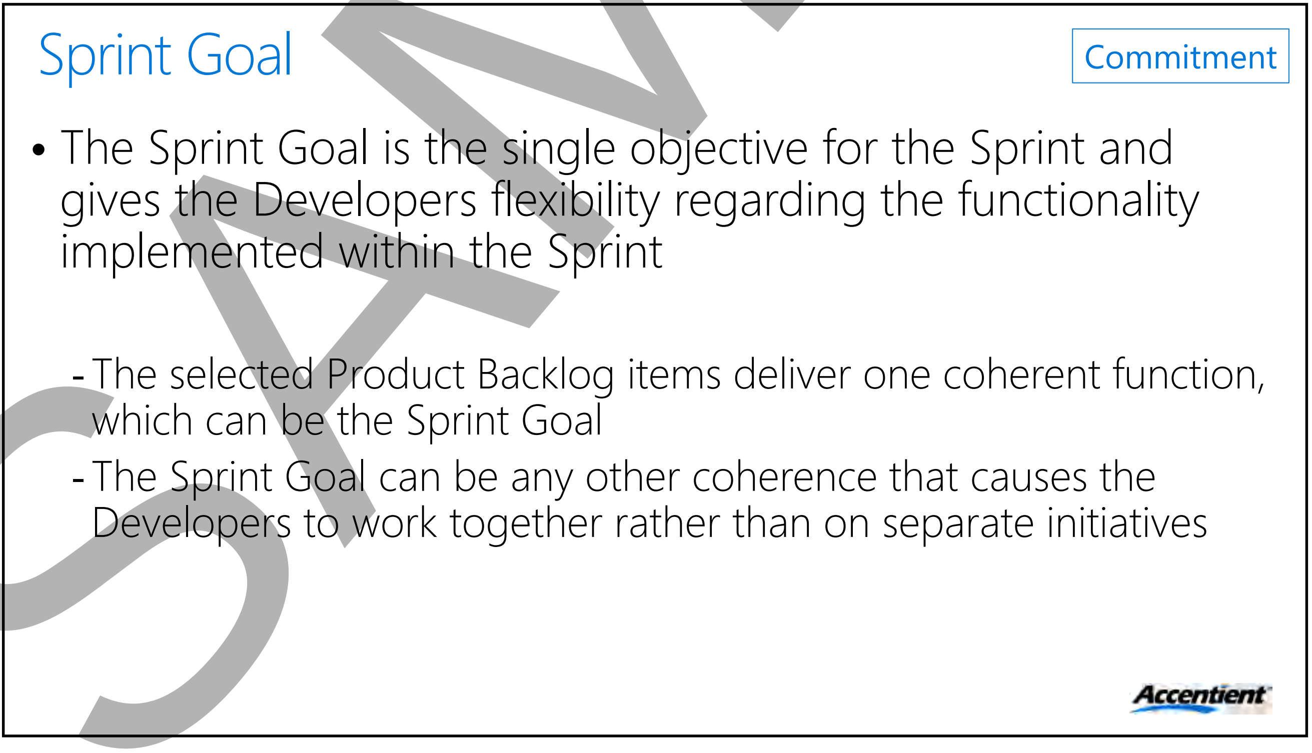
✓ Create the plan (using tasks, tests, etc.)

Accentient

## The Sprint Goal

*The "Why"*

Accentient



## Sprint Goal

Commitment

- The Sprint Goal is the single objective for the Sprint and gives the Developers flexibility regarding the functionality implemented within the Sprint
  - The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal
  - The Sprint Goal can be any other coherence that causes the Developers to work together rather than on separate initiatives

Accentient

## Sprint Goal Examples

"To enable the application to run while offline"

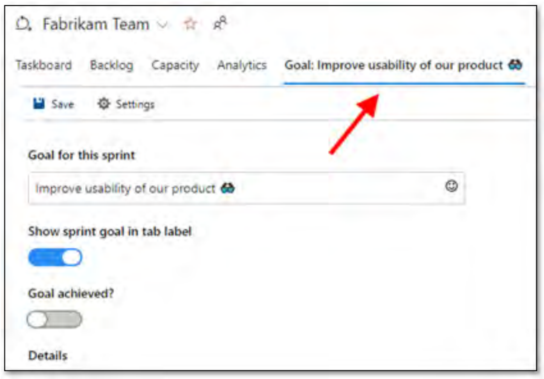
"To deliver a minimal set of administration features"

"To decrease latency by 50%"

"To increase search accuracy by 20%"

## Sprint Goal Extension

- Enables a team to configure a Sprint goal to be seen on the Sprint pages in Azure Boards
  - Set a goal per Sprint (iteration)
  - Set a goal per team (optional)
  - Show goal in the tab label
  - Specify if the goal as achieved
  - View previous Sprint Goals



Goal: Improve usability of our product

Goal for this sprint: Improve usability of our product

Show sprint goal in tab label: on

Goal achieved?: off

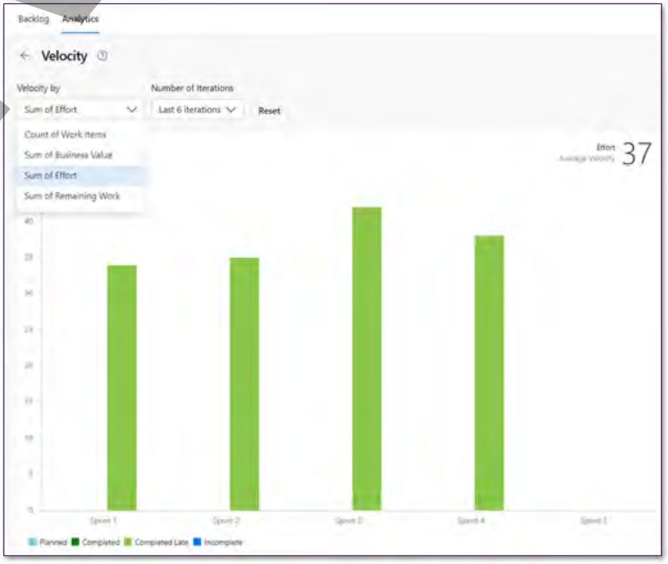
Visit https://bit.ly/3xSE92F for more information

## The Forecast

*The "What"*

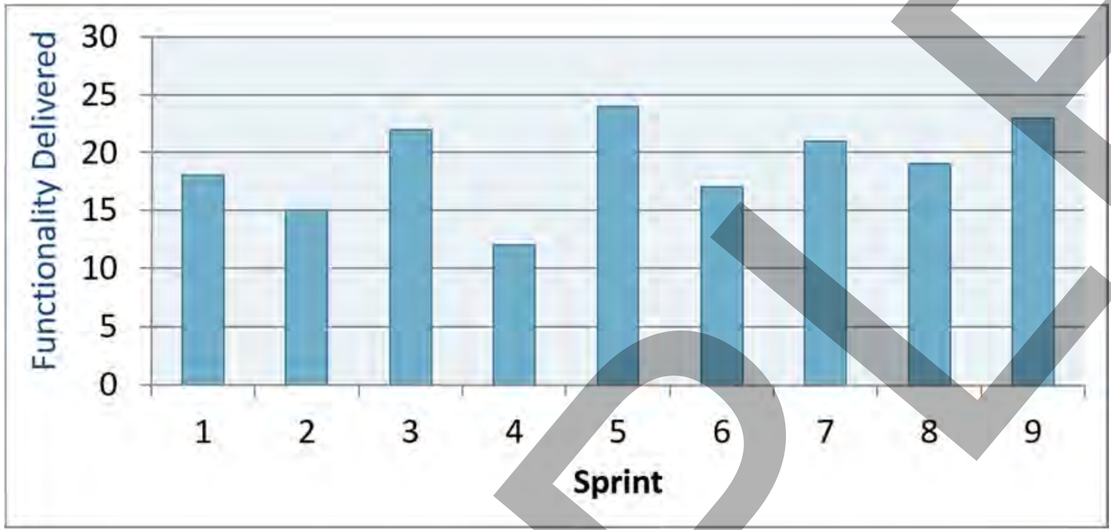
## Know Your Team's Velocity

- The Analytics page presents velocity in various ways
  - Count of work items
  - Sum of business value
  - Sum of effort
  - Sum of remaining work
- <u>Remember</u>: Velocity may vary over time for many reasons



## What is This Team's Velocity?

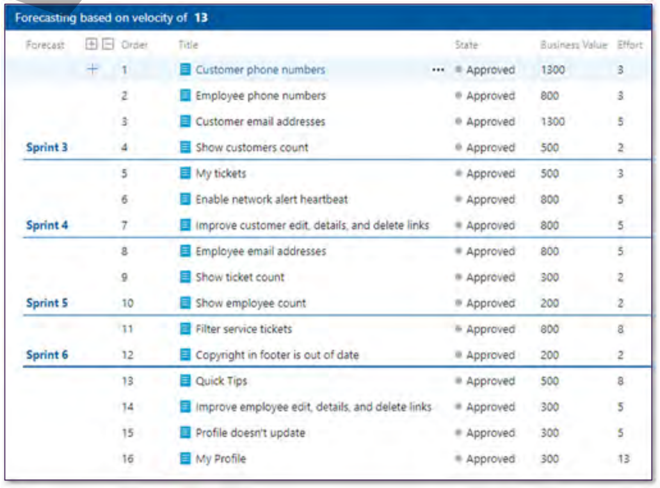
1 MIN



Accentient

## Use the Forecast Tool

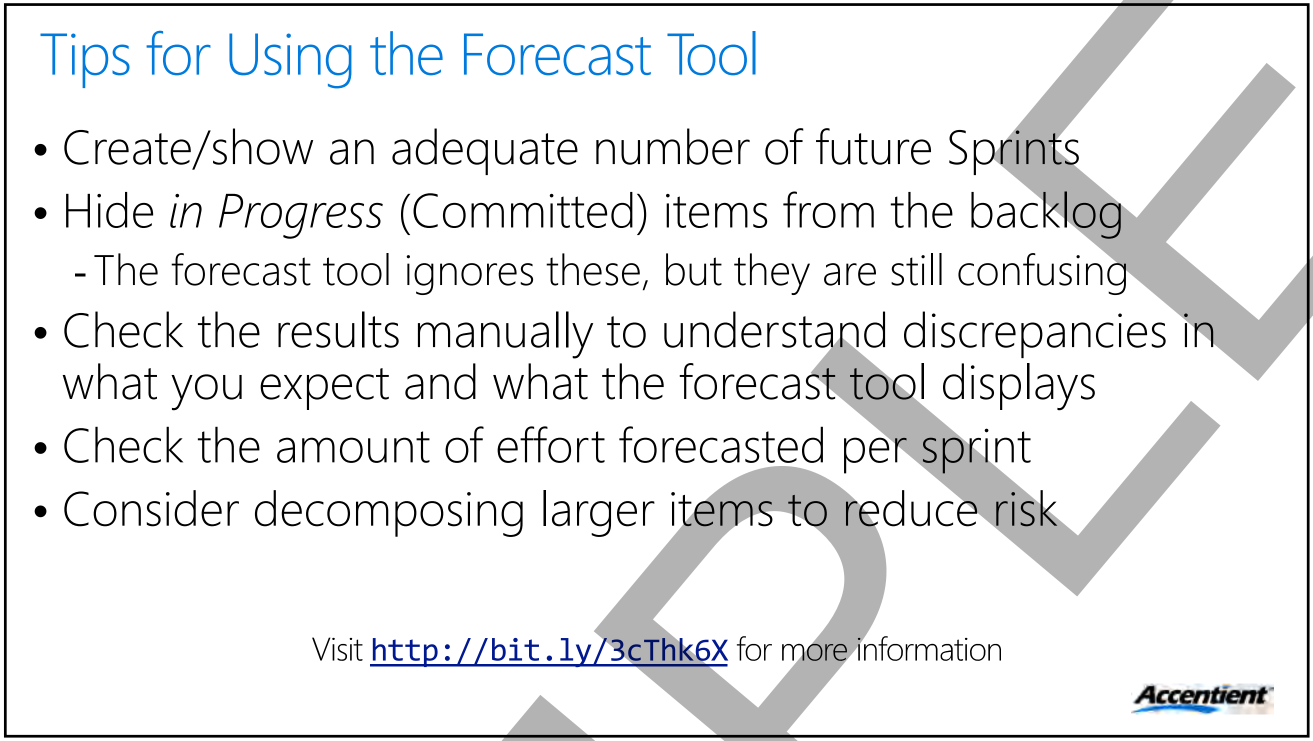
- The forecast tool helps a team anticipate the work that can be delivered in upcoming Sprints
  - Helpful for release planning
  - Requires a known velocity
- <u>Caution</u>: This tool should not be a replacement for the conversations that take place during Sprint Planning



Forecasting based on velocity of 13

| Forecast | Order | Title | State | Business Value | Effort |
|---|---|---|---|---|---|
| | 1 | Customer phone numbers | Approved | 1300 | 3 |
| | 2 | Employee phone numbers | Approved | 800 | 3 |
| | 3 | Customer email addresses | Approved | 1300 | 5 |
| Sprint 3 | 4 | Show customers count | Approved | 500 | 2 |
| | 5 | My tickets | Approved | 500 | 3 |
| | 6 | Enable network alert heartbeat | Approved | 800 | 5 |
| Sprint 4 | 7 | Improve customer edit, details, and delete links | Approved | 800 | 5 |
| | 8 | Employee email addresses | Approved | 800 | 5 |
| | 9 | Show ticket count | Approved | 300 | 2 |
| Sprint 5 | 10 | Show employee count | Approved | 200 | 2 |
| | 11 | Filter service tickets | Approved | 800 | 8 |
| Sprint 6 | 12 | Copyright in footer is out of date | Approved | 200 | 2 |
| | 13 | Quick Tips | Approved | 500 | 8 |
| | 14 | Improve employee edit, details, and delete links | Approved | 300 | 5 |
| | 15 | Profile doesn't update | Approved | 300 | 5 |
| | 16 | My Profile | Approved | 300 | 13 |

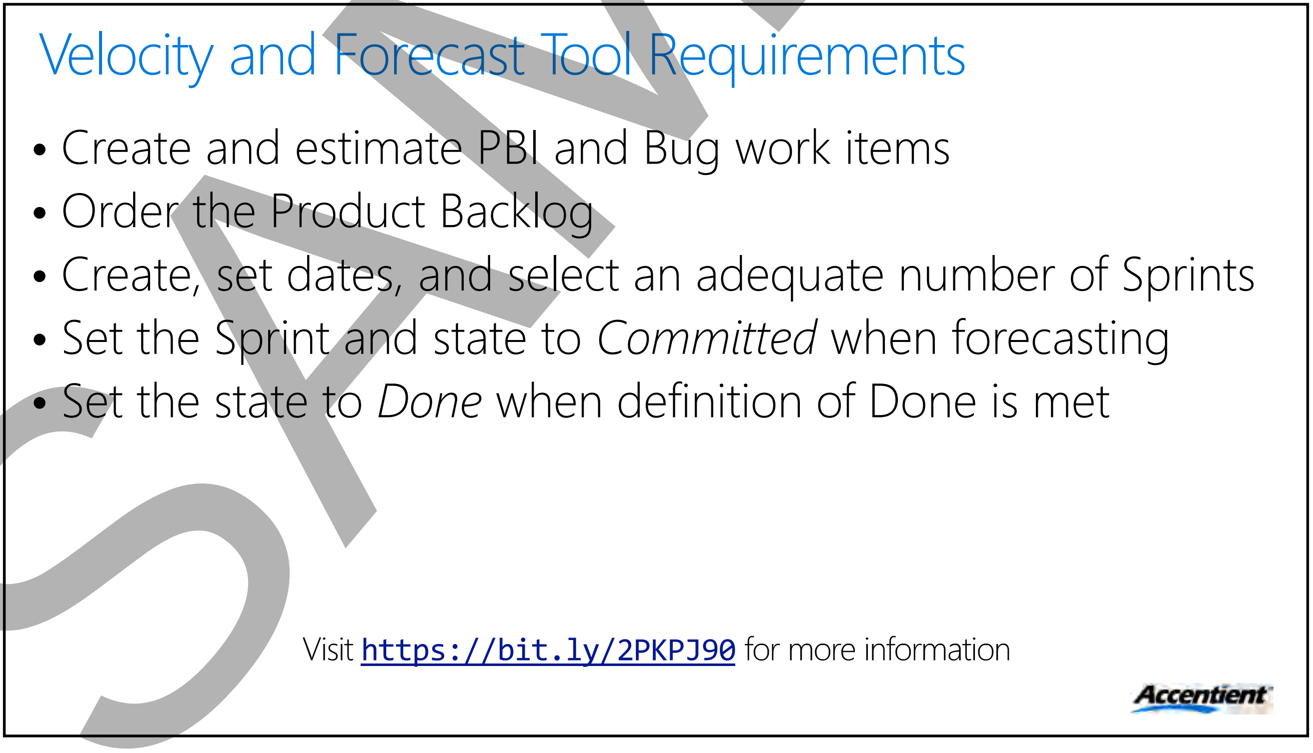
Accentient



## Tips for Using the Forecast Tool

- Create/show an adequate number of future Sprints
- Hide *in Progress* (Committed) items from the backlog
  - The forecast tool ignores these, but they are still confusing
- Check the results manually to understand discrepancies in what you expect and what the forecast tool displays
- Check the amount of effort forecasted per sprint
- Consider decomposing larger items to reduce risk

Visit `http://bit.ly/3cThk6X` for more information



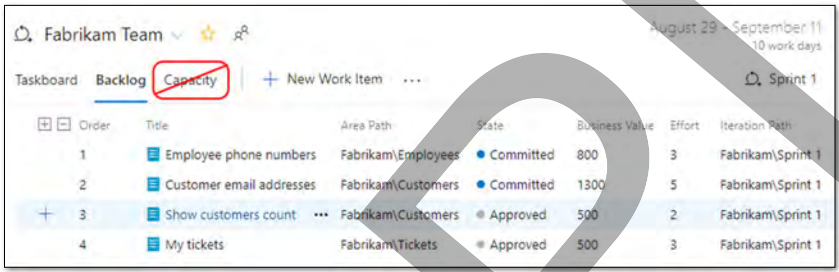
## Velocity and Forecast Tool Requirements

- Create and estimate PBI and Bug work items
- Order the Product Backlog
- Create, set dates, and select an adequate number of Sprints
- Set the Sprint and state to *Committed* when forecasting
- Set the state to *Done* when definition of Done is met

Visit `https://bit.ly/2PKPJ90` for more information

## Consider Capacity During Sprint Planning

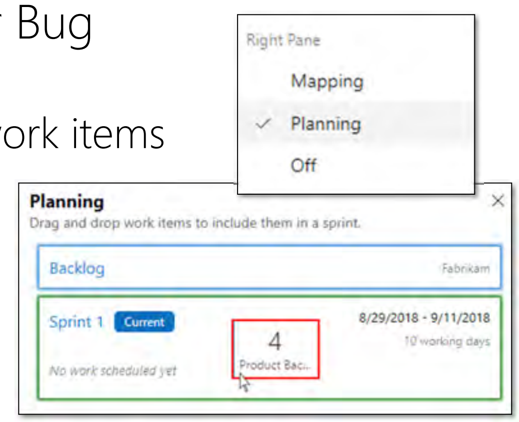
- Capacity, as with Velocity, is another input to Sprint Planning
  - The team should have a feeling for its own capacity (holidays, vacations, shared team members, variation in productivity, etc.)
- Tip: The Scrum Team should have a conversation rather than use the capacity planning tool



| Order | Title | Area Path | State | Business Value | Effort | Iteration Path |
|---|---|---|---|---|---|---|
| 1 | Employee phone numbers | Fabrikam\Employees | Committed | 800 | 3 | Fabrikam\Sprint 1 |
| 2 | Customer email addresses | Fabrikam\Customers | Committed | 1300 | 5 | Fabrikam\Sprint 1 |
| 3 | Show customers count | Fabrikam\Customers | Approved | 500 | 2 | Fabrikam\Sprint 1 |
| 4 | My tickets | Fabrikam\Tickets | Approved | 500 | 3 | Fabrikam\Sprint 1 |

## Create the Forecast

- From the Backlogs page, show the *Planning* window
  - If necessary, create a missing Sprint from the Planning window
- Drag and drop one or more PBI or Bug work items to the current Sprint
  - This sets the *Iteration Path* of those work items
- Next, go to the Sprint backlog and manually set the *State* of those items to *Committed*



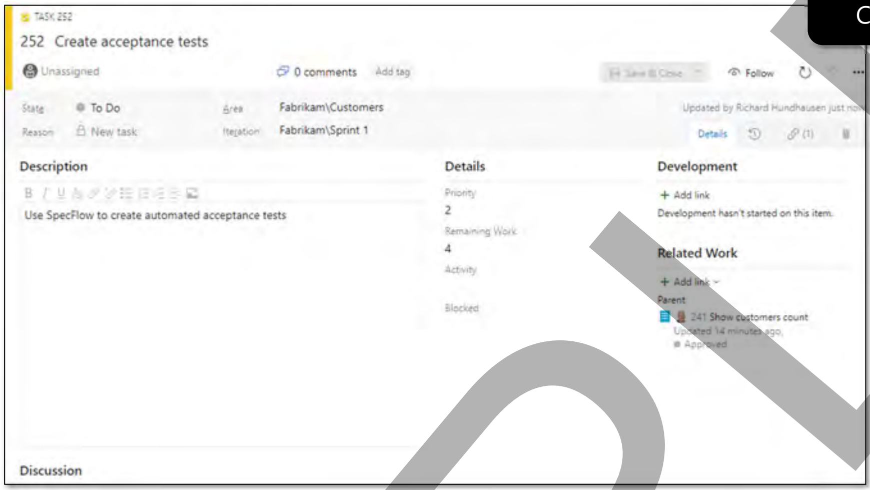
## The Plan

### The "How"

## Practice: Use Tasks to Represent Your Plan

- Once the forecast is made, decompose the *PBI* work items into *Task* work items
- These tasks represent the actual work to be performed by the Developers to implement the item
  - May include test, design, code, database, content, release, etc.
  - Expressed in hours and estimated as a team
  - Should be 8 hours or less
  - Should be unassigned until someone starts work
- Progress can be assessed by the *number* of done vs. undone tasks or the *sum* of the hours of those undone tasks

## The Task Work Item Type



252 Create acceptance tests

| | | | |
|---|---|---|---|
| State | To Do | Area | Fabrikam\Customers |
| Reason | New task | Iteration | Fabrikam\Sprint 1 |

Updated by Richard Hundhausen just now

**Description**

Use SpecFlow to create automated acceptance tests

**Details**

| | |
|---|---|
| Priority | 2 |
| Remaining Work | 4 |
| Activity | |
| Blocked | |

**Development**

+ Add link

Development hasn't started on this item.

**Related Work**

+ Add link

Parent

241 Show customers count

Updated 14 minutes ago,

Approved

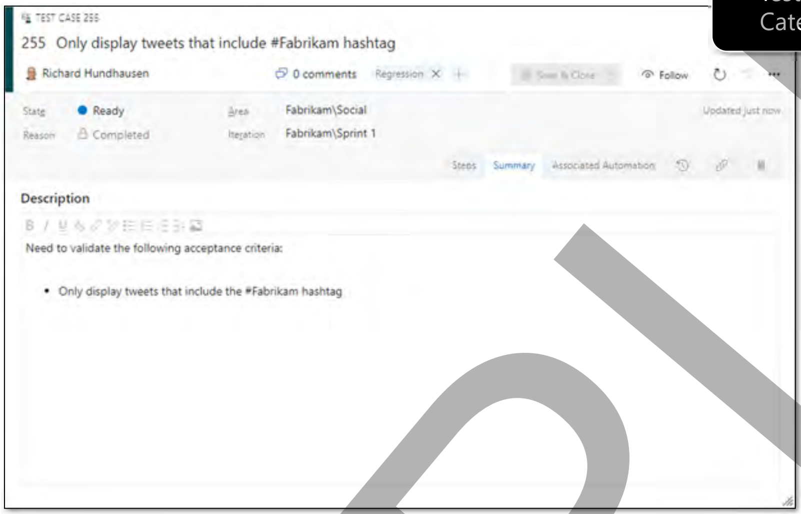
**Discussion**

Task Category

## Practice: Use Failing Tests to Represent Your Plan

- Once the forecast is made, convert the acceptance criteria of the PBI work items into failing acceptance tests in the form of *Test Case* work items
  - These are eventually associated with automated or manual tests
  - The team focuses every day on getting more tests to pass
  - This is the basis for Acceptance Test-Driven Development
- Progress can be assessed by the number of tests that are not yet passing

## The Test Case Work Item Type


TEST CASE 255

255 Only display tweets that include #Fabrikam hashtag

Richard Hundhausen | 0 comments | Regression | Save & Close | Follow

State: Ready | Area: Fabrikam\Social | Updated just now

Reason: Completed | Iteration: Fabrikam\Sprint 1

Steps | Summary | Associated Automation

Description

Need to validate the following acceptance criteria:

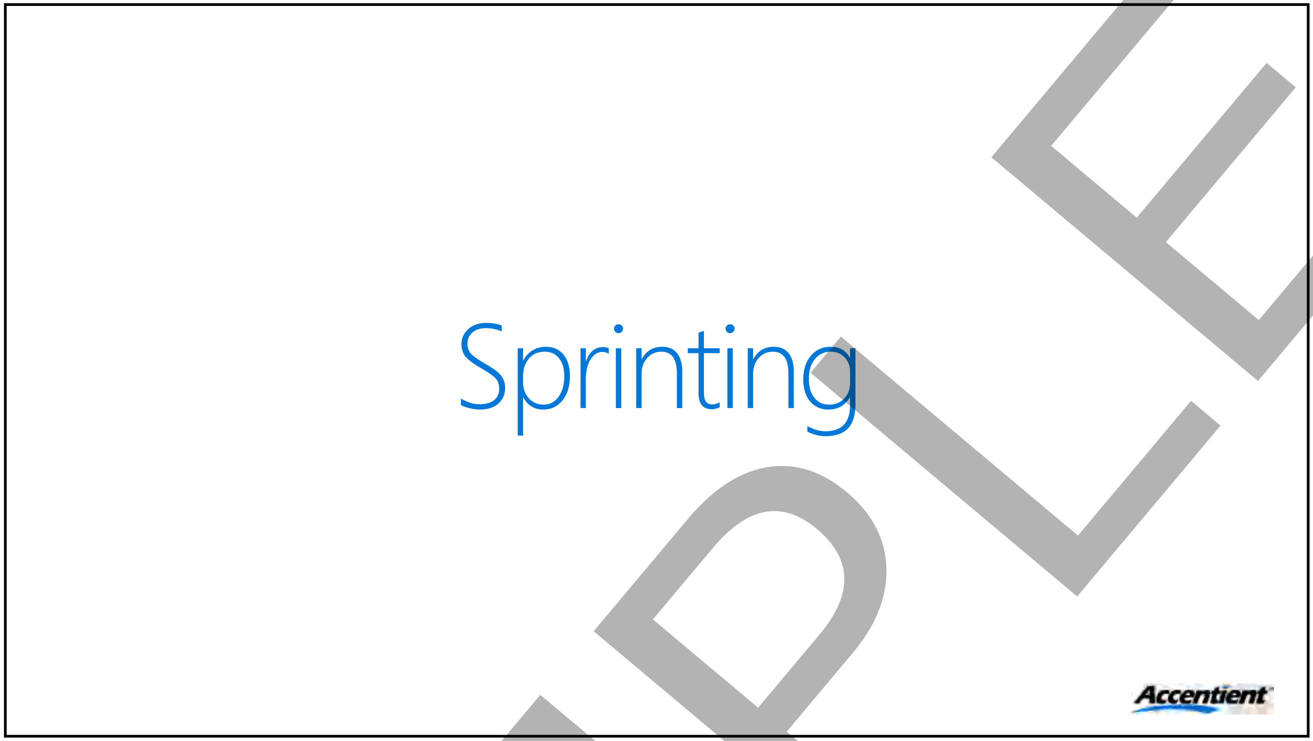
- Only display tweets that include the #Fabrikam hashtag

Test Case Category

## Practice: Use a Conversation to Represent Your Plan

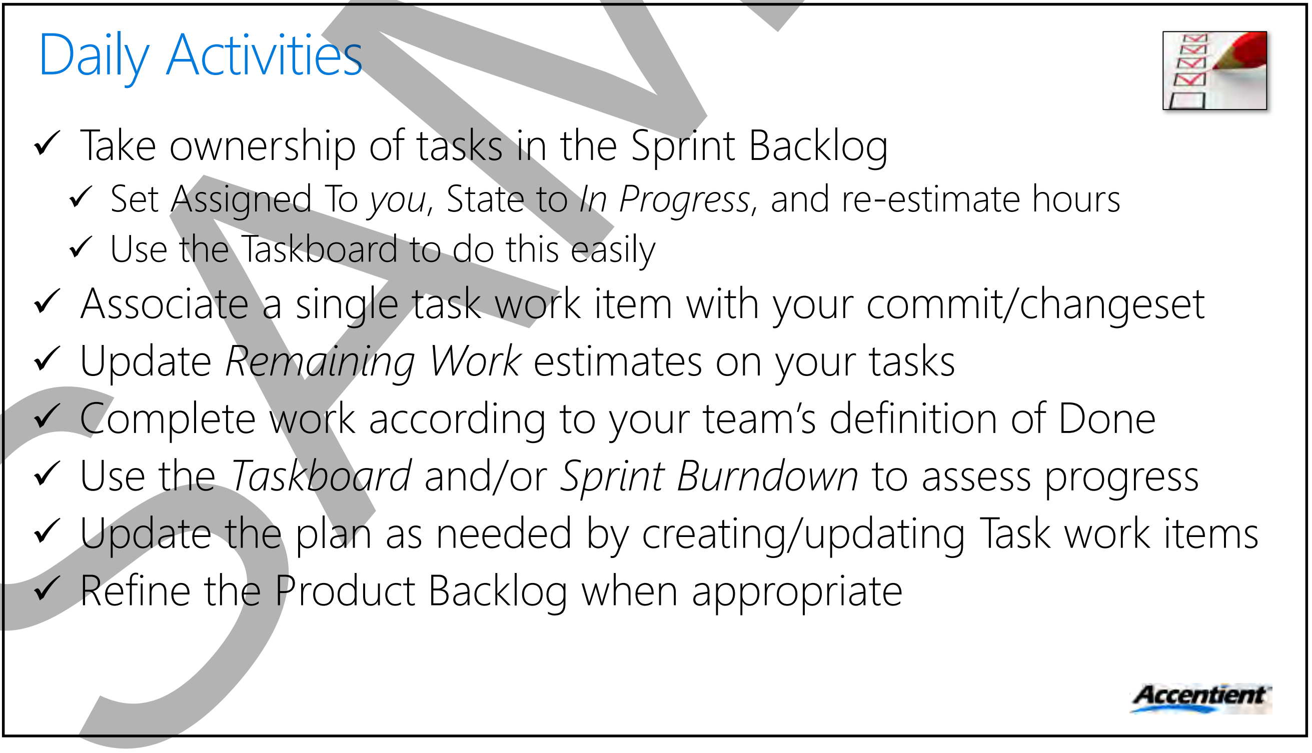
- Once the forecast is made, the PBI work items themselves represent the plan, knowing that the team will swarm on the work, designing it, and executing it
  - The Board view of the Product Backlog can be used
  - This practice is for high-performance teams

- Progress can be assessed by the *number* of PBI work items not yet in the *Done* state, or the percentage of Done effort (story points) vs. undone effort (story points)



## Sprinting

Accentient



## Daily Activities

- ✓ Take ownership of tasks in the Sprint Backlog
  - ✓ Set Assigned To *you*, State to *In Progress*, and re-estimate hours
  - ✓ Use the Taskboard to do this easily
- ✓ Associate a single task work item with your commit/changeset
- ✓ Update *Remaining Work* estimates on your tasks
- ✓ Complete work according to your team's definition of Done
- ✓ Use the *Taskboard* and/or *Sprint Burndown* to assess progress
- ✓ Update the plan as needed by creating/updating Task work items
- ✓ Refine the Product Backlog when appropriate

Accentient

## Definition of Done

Commitment

- A formal description of when the Increment meets the quality measures required for the product
  - Work cannot be considered part of an Increment unless it meets the Definition of Done
- The definition establishes a shared understanding of what it means when the team says *Done* vs. not *Done*

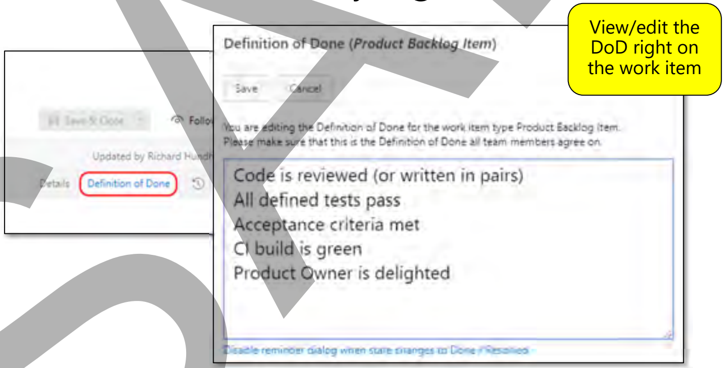
**Example Definition of Done**

- Code compiles without errors or warnings
- An automated CI build exists
- All new code covered by unit tests
- Test coverage does not go down
- All acceptance criteria has been met
- Release note describing new functionality
- Paula (Product Owner) likes our work

Accentient

## Definition of Done Extension

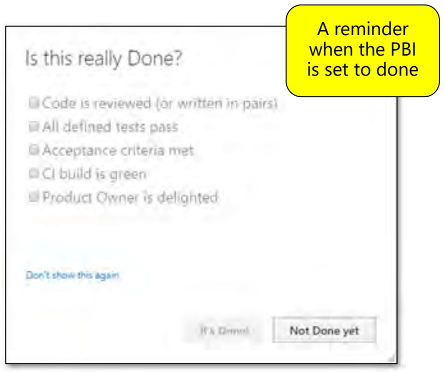
- Definition of Done support inside work item form
  - A free extension by Agile Extensions


Save & Close
Follow
Updated by Richard Hundh
Details
Definition of Done
Save
Cancel
View/edit the DoD right on the work item
Disable reminder dialog when state changes to Done / Resolved

**Definition of Done (*Product Backlog Item*)**

You are editing the Definition of Done for the work item type Product Backlog Item.
Please make sure that this is the Definition of Done all team members agree on.

Code is reviewed (or written in pairs)
All defined tests pass
Acceptance criteria met
CI build is green
Product Owner is delighted



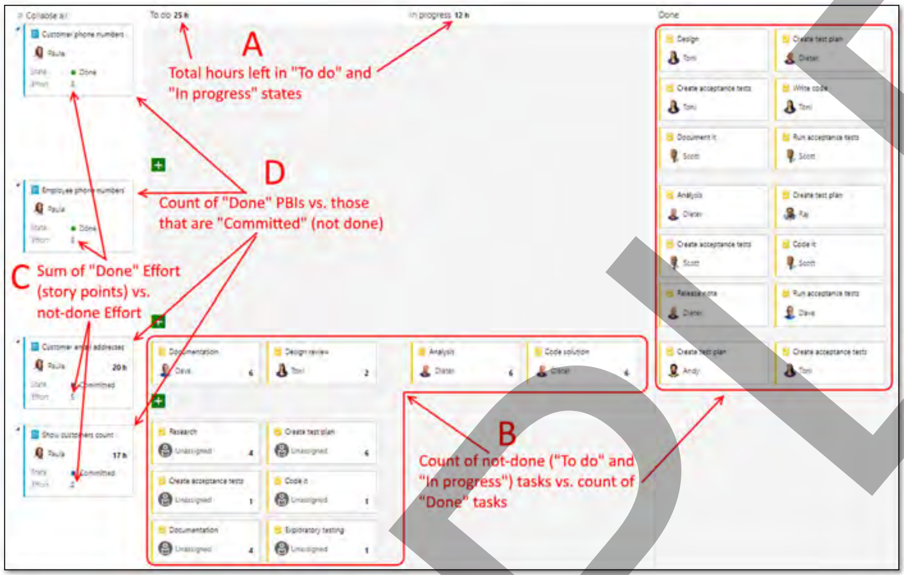
Is this really Done?

- ☐ Code is reviewed (or written in pairs)
- ☐ All defined tests pass
- ☐ Acceptance criteria met
- ☐ CI build is green
- ☐ Product Owner is delighted

Visit https://bit.ly/2xvM5bX for more information

Accentient

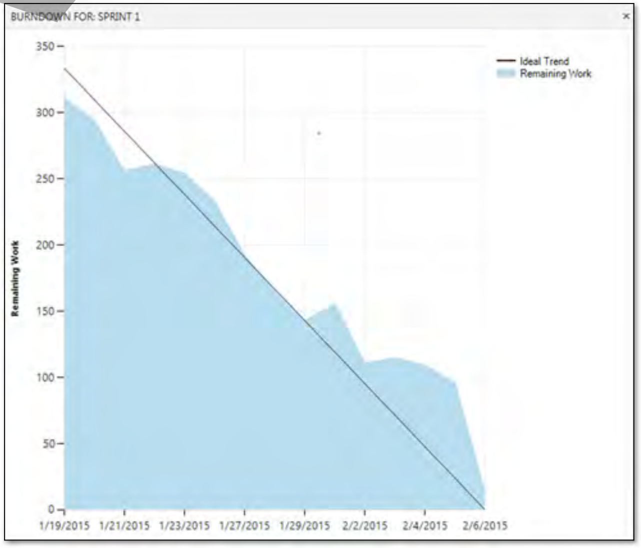
## Assessing Progress Using the Taskboard


A
Total hours left in "To do" and "In progress" states
D
Count of "Done" PBIs vs. those that are "Committed" (not done)
C
Sum of "Done" Effort (story points) vs. not-done Effort
B
Count of not-done ("To do" and "In progress") tasks vs. count of "Done" tasks

Visit https://bit.ly/2OxxQuf for more information

## Assessing Progress Using the Burndown Chart

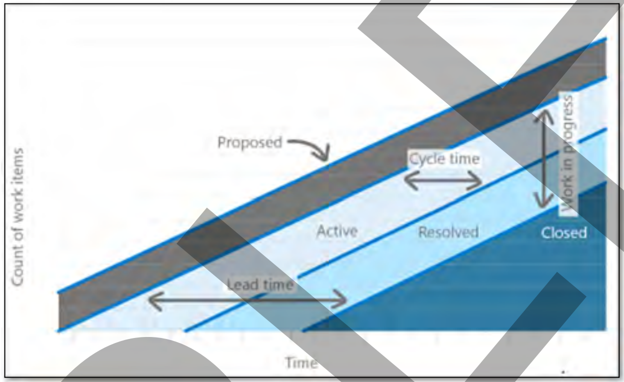
- Reflects the progress made by completing tasks
  - Examines *Remaining Work*
  - Available on the *Sprints* page
- The ideal trend line shows a smooth and steady burndown
  - The blue area shows actual changes in remaining work



Visit https://bit.ly/36sCLbu for more information

## Assessing Progress Using the Cumulative Flow Diagram

- Used to monitor the flow of work through a system
  - Displays the count of work items grouped by state/ column over time
- Two primary metrics
  - Cycle time
  - Lead time



- CFDs are used more by Kanban teams than by Scrum teams

Visit http://bit.ly/34WIKl0 for more information

Accentient

## Impediment

- Anything that impedes or slows a Scrum Team's progress *and* cannot be resolved by the Scrum Team internally
- Examples
  - Unforeseen/undesired changes in team composition
  - Product Owner unavailability or indecisiveness
  - Undesired pressure from management
  - Lots of technical debt, issues with tooling
  - Conflict between team members, scarcity of skills
  - Too many unimportant meetings
- Impediments should be removed, not tracked!

Accentient

## What Are Some Impediments?

5 MIN

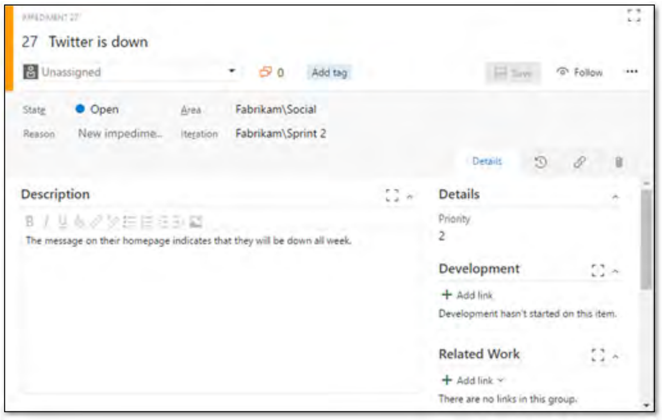
What impediments have you experienced?

How did you remove them?

Accentient

## The Impediment Work Item Type

- An impediment is anything that prevents the team or a team member from performing work as efficiently as possible
  - Impediments are not bugs
- Impediments can be captured as an Impediment work item



IMPEDIMENT 27

27 Twitter is down

Unassigned · 0 · Add tag · Save · Follow

State: Open · Area: Fabrikam\Social

Reason: New impedime... · Iteration: Fabrikam\Sprint 2

Details

Description

The message on their homepage indicates that they will be down all week.

Details

Priority
2

Development

+ Add link

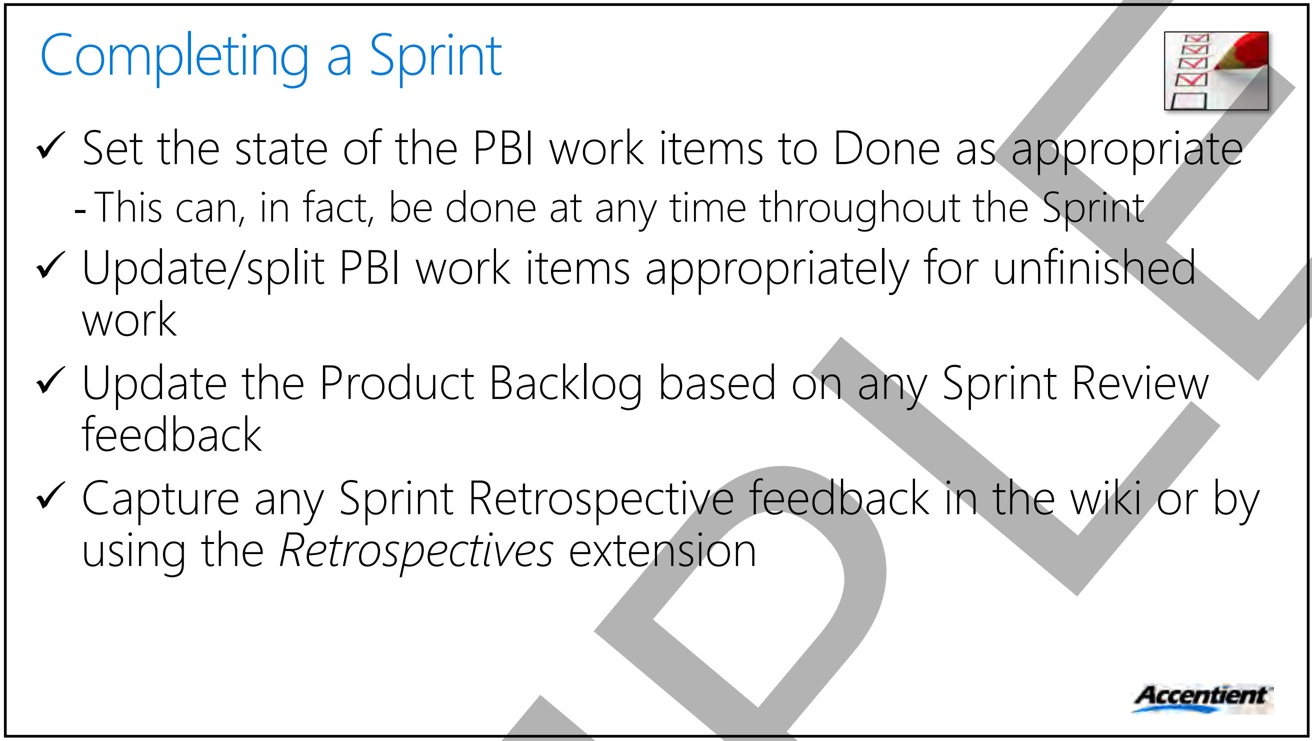
Development hasn't started on this item.

Related Work

+ Add link

There are no links in this group.

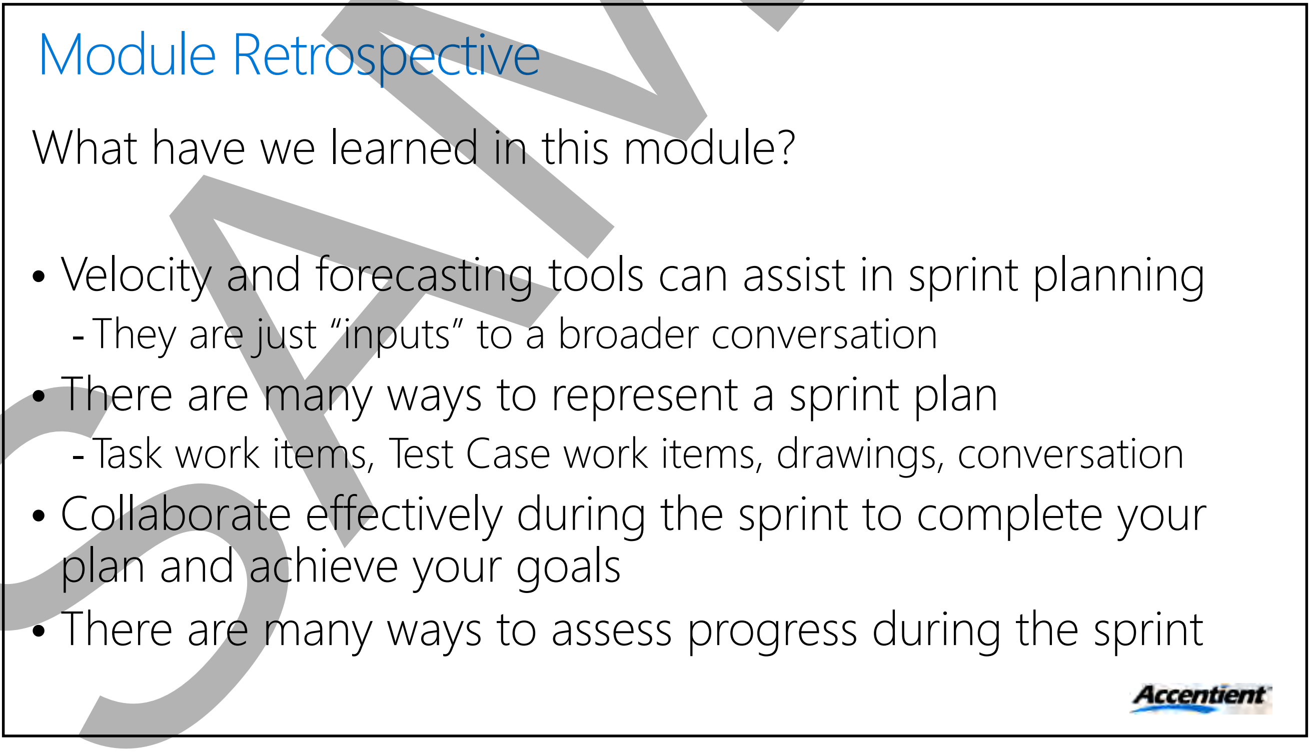
Accentient



## Completing a Sprint

- ✓ Set the state of the PBI work items to Done as appropriate
  - This can, in fact, be done at any time throughout the Sprint
- ✓ Update/split PBI work items appropriately for unfinished work
- ✓ Update the Product Backlog based on any Sprint Review feedback
- ✓ Capture any Sprint Retrospective feedback in the wiki or by using the *Retrospectives* extension

Accentient



## Module Retrospective

What have we learned in this module?

- Velocity and forecasting tools can assist in sprint planning
  - They are just “inputs” to a broader conversation
- There are many ways to represent a sprint plan
  - Task work items, Test Case work items, drawings, conversation
- Collaborate effectively during the sprint to complete your plan and achieve your goals
- There are many ways to assess progress during the sprint

Accentient

## Lab

60 Minutes

In this lab you will plan a Sprint.

- Configure Sprints
- Forecast work
- Plan a Sprint
- Create Task work items
- Create a Definition of Done
- Create an Impediment work item (optional)
- Install and use the Retrospectives extension (optional)

Accentient

# Lab 4: Create and Configure an Azure DevOps Project

**Delivering Products Using Azure DevOps and Scrum**

## LAB OVERVIEW

This lab walks you and your colleagues through the process of forming into teams and provisioning the Azure DevOps Services organization that will be used during this course, including creating and configuring an Azure DevOps project.

Estimated time to complete this lab: **45 minutes**

### Task Execution

As this is a team-based training course, there are a number of opportunities for team members to learn to collaborate more effectively. Unfortunately, there is a possibility for team members to accidentally impede, block, or otherwise cause unintentional conflicts. To minimize the possibility of conflicts, critical tasks in this course have been marked with an icon indicating who on the team should execute the task:

- ●●● The team can self-organize and execute the task however they decide
- ●○○ Only the "leader" should execute this task
- ○●● Only the "followers" (not the leader) should execute this task
- ● Everyone on the team should execute this task
- ● ● Everyone on the team should execute this task (working in pairs)

<u>Tip</u>: Look for the "leader" ●○○ tasks and ensure that they are only performed once per team. Also, ensure that the "follower" ○●● tasks are only performed by everyone else (not the leader).

### Teams of One

If you are working by yourself and not on a team, make sure to perform all of the "leader" ●○○ tasks, and none of the "follower" ○●● tasks. This scenario is common for students learning remotely.

## EXERCISE 1 – SET UP THE ENVIRONMENT

### Task: Install Courseware Files

In this task you will install the files required by this class.

Dependencies

- Signed in as *Administrator*

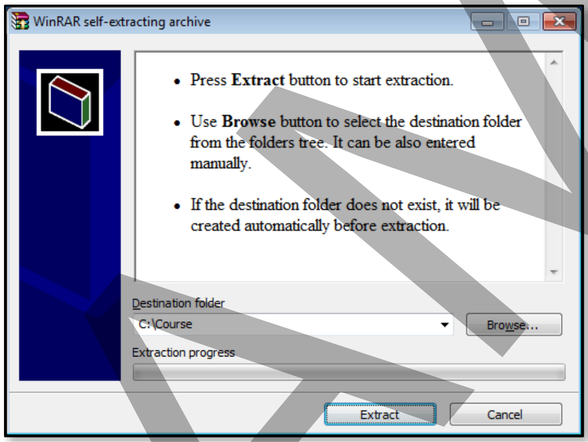
1. Verify that the **C:\Course** folder does not exist.

   Note: If this folder already exists, then your computer may have been used for a prior training class. If that is the case, then the effectiveness of the hands-on labs that follow may be diminished.

2. If necessary, copy the courseware file to your desktop.

   This file may already be on the desktop. If not, you may have to ask your instructor for help locating and/or copying this file. If you cannot locate this file, please email support@accentient.com to obtain a copy.

3. Extract the courseware files, specifying **`C:\Course`** as the **Destination folder**.



WinRAR self-extracting archive

- Press **Extract** button to start extraction.
- Use **Browse** button to select the destination folder from the folders tree. It can be also entered manually.
- If the destination folder does not exist, it will be created automatically before extraction.

Destination folder: C:\Course

Extraction progress

It can take a few moments to extract the files. The folder C:\Course will be created during the process. After extracting the files, you should have one or more of the following sub-folders:

- C:\Course\Guidance
- C:\Course\Labs
- C:\Course\Software

## EXERCISE 2 – SET UP YOUR TEAM

### Task: Form Into Teams (optional)

If necessary, your instructor will facilitate the creation of equally sized (5 team members or less), cross-functional, collocated teams.

- Identify yourself by role …
  - QA/test
  - Architect/designer
  - Programmer/coder
  - Database developer
  - Business/requirements
  - Management
- Form into cross-functional teams (of 5 members or less)
- Collocate (physically or virtually)
- Introduce yourself (if necessary)
- Decided on a team name (i.e. “Team Blue”, “The Honey Badgers”, “Backlog Boyz”, etc.)
- Write your name and team name where it is visible to others in the class

What is the name of your team? ______________________________________________________

Who are your team members? ______________________________________________________

________________________________________________________________________________

________________________________________________________________________________

________________________________________________________________________________

________________________________________________________________________________

## Task: Identify a Microsoft Account

In this task you will identify, or create if necessary, a Microsoft Account that you will use for the various online services leveraged during this class.

What is your Microsoft Account? ______________________________________________

1. If you already have a Microsoft Account, write it on the line above and skip the rest of these steps.
2. Open **Chrome** and navigate to `https://signup.live.com`.

   If you don't have Google Chrome installed, you can install it from here: `www.google.com/chrome`.
3. Provide the required information.

   This will include your name, username, password, country/region, zip code, birthdate, and gender. You can create a new *outlook.com* or *hotmail.com* address. You will also need to provide a phone number or other method to be able to reset your password.
4. **Create** the account, accepting the Microsoft Services Agreement and privacy statement.

## Task: Create an Azure DevOps Services Organization

In this task your team will select someone to create a new Azure DevOps Services Organization, configure a Professional Scrum process, and create a new Azure DevOps project.

Who will be performing this task? ______________________________________________

1. Launch **Chrome**, navigate to https://dev.azure.com, and get started for free.

   You may be asked to review and accept the licensing terms. Don't create a new project yet.
2. With the new organization selected, select **Organization settings**.
3. Go to the **Overview** page.

   What is the organization *Name*? *URL*? ______________________________________

   If you are happy with this organization name and URL, then skip the next step.
4. Change the organization **Name** to a more meaningful name – maybe a variation of your team name.

   What is the new organization *URL*? ______________________________________
5. Share the **URL** with your team members.
6. Change the **Time zone** accordingly.

## Task: Create a Professional Scrum process

In this task the same user who performed the previous task will create an inherited Professional Scrum process to be used later. For more information, see `http://bit.ly/ProScrumProcess`.

Who will be performing this task? ___________________________________________________

1. From **Organization Settings**, go to the **Process** page.

   What processes are available? ___________________________________________________

2. Right-click on the **Scrum** process and select **Create inherited process**.

3. Name the new process **Professional Scrum**, specify the description **Scrum according to the Scrum Guide**, and click **Create process**.

4. Click the new **Professional Scrum** process.

5. Right-click on the **Bug** work item type and select **Disable**.

   No, this isn't a clever way to say that there are "no bugs" in our product. We will just use the *Product Backlog Item* work item type to represent all work, including bugs.

6. Click the **Product Backlog Item** work item type to edit it.

7. In the **Details** group, right-click on **Priority** and select **Hide from layout**.

   We won't need this field on our Scrum Team.

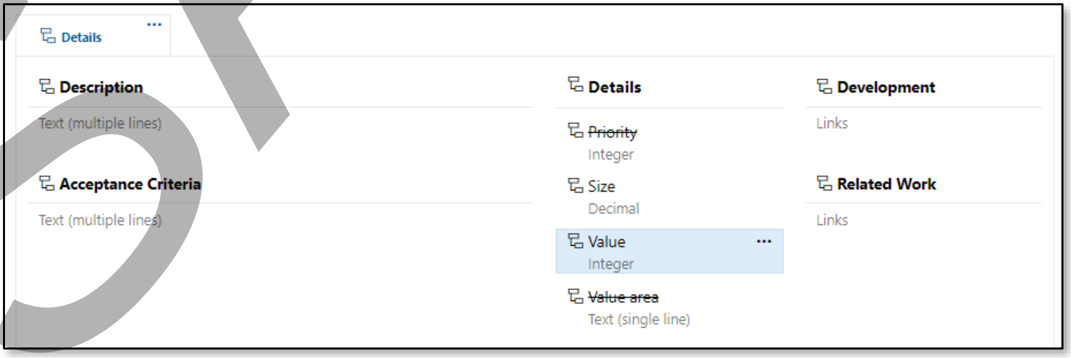
8. Do the same for the **Value area** field.

9. In the **Details** group, right-click on **Effort** select **Edit**, change the label to **Size** and click **Save**.

   *Size* is a more generic term than "effort", "complexity", "points", "story points", etc.

10. Edit the **Business Value** field and change its label to just **Value**.

    *Value* is a more generic, and simpler term. Your new process should look something like this so far …



11. Click **States** at the top.

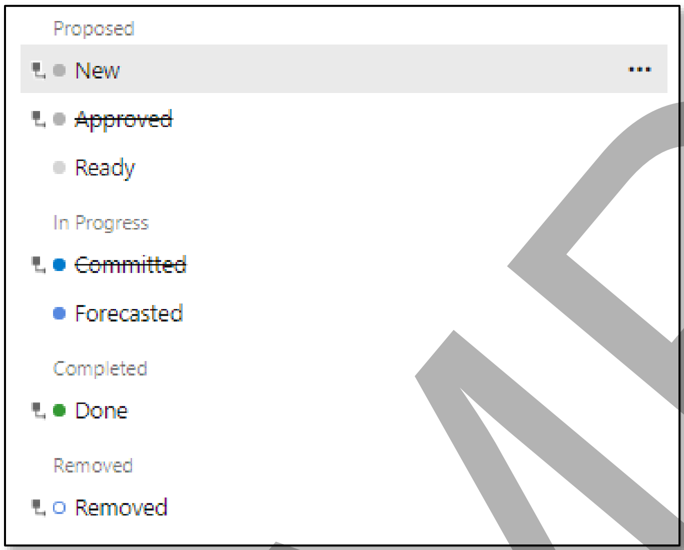
    We are going to make some corrections to the State values.

12. Add a **New state**, named **Ready** that maps to the **Proposed** category, and click **Create**.

13. Add a **New state**, named **Forecasted** that maps to the **In Progress** category, and click **Create**.

14. **Hide** the redundant **Approved** and **Committed** states.

    Our new states match the Scrum Guide a bit closer now …


Proposed
New
~~Approved~~
Ready
In Progress
~~Committed~~
Forecasted
Completed
Done
Removed
Removed

15. Return to the Professional Scrum page by clicking **Professional Scrum** in the breadcrumb navigation.

16. Go to **Backlog levels**.

17. In the Requirement backlog section, right-click on **Backlog items** and select **Edit/Rename**.

18. Change the name to **Stories** and click **Save**.

    Even though we will continue to use the Product Backlog Item work item type, we'll call this backlog level *Stories*, to go along with Epics and Features. This is a pretty common hierarchy among contemporary Scrum Team. Also, remember that in Scrum, items in any of these levels are simply called Product Backlog Items (PBIs).

19. Return to the Process page by clicking **All Processes** in the breadcrumb navigation.

20. Right-click on the new **Professional Scrum** process and select **Set as default process**.

## Task: Create the Fabrikam Project

In this task the same user who performed the previous task will create a new Fabrikam project.

1. From **Organization Settings**, go to the **Projects** page.

2. Create a **new project**:

   - Project name: **Fabrikam**
   - Description: **Fabrikam Fiber**
   - Visibility: **Private**
   - Version Control: **Git**
   - Work item Process: **Professional Scrum**

3. Select the newly created **Fabrikam** project.

   This will take you to the Fabrikam *Project Settings* page. If you need to get here manually, click the *Project settings* link at the bottom left.

4. Go to the **Overview** page.

   Which Azure DevOps *services* are enabled? ______________________________________________

## Task: Configure the Team

In this task the same user who performed the previous task will configure the default Fabrikam team.

1. In Fabrikam **Project Settings**, go to the **Permissions** page.

2. Select the **Fabrikam Team** group.

3. Click the **Member of** link at the top.

4. **Add** the **[Fabrikam]\Project Administrators** group.

   Note: In practice, you should check with your administrator before adding all members of a team to the *Project Administrators* group.

5. **Delete Membership** in the **Contributors** group.

   There should now be two groups listed: *Project Valid Users* and *Project Administrators*.

6. Click the **Members** link at the top.

   How many members are currently listed? ______________________________________________

7. **Add** the Microsoft Accounts of your teammates.

   How many team members are listed? ______________________________________________

   Note: Microsoft may send invitational emails to your colleagues. These emails can be ignored.

## Task: Join the Fabrikam Team

In this task the rest of the team members will access the newly created project.

Note: You may receive an invitational email from Microsoft. You can ignore this.

1. Open your browser and navigate to the URL of your project.

   This was the URL that was shared with you in a previous task. You should bookmark this URL.

2. If prompted, enter your **Microsoft Account** and **password**.

   If you have any difficulties signing in, double-check your account and password. If that doesn't solve the problem, ask your colleague to double-check your team membership. If you are still blocked, visit `https://aka.ms/vssignout` to sign out, go *Incognito* (Chrome) or check with your instructor.

## Task: Update Your Profile

In this task everyone will update their individual profile.

1. In the upper-right corner, click User settings, and go to your **Profile**.

2. Verify that your full name and contact email address are accurate.

3. Change your image to something that better "represents" you.

   This is an optional step, but will up your team's social experience.

4. Return to User settings, select **Notifications**, and unsubscribe from all notifications.

   This page shows subscriptions that you have created or have been created for you by an administrator or OOB (out-of-the-box). Active subscriptions are indicated with the State as *On*.

   Note: There may be a slight delay as you click each toggle, but in the end, your inbox will thank you.

5. If you are working in a pair, sign out, and have your buddy do these tasks to update their profile.

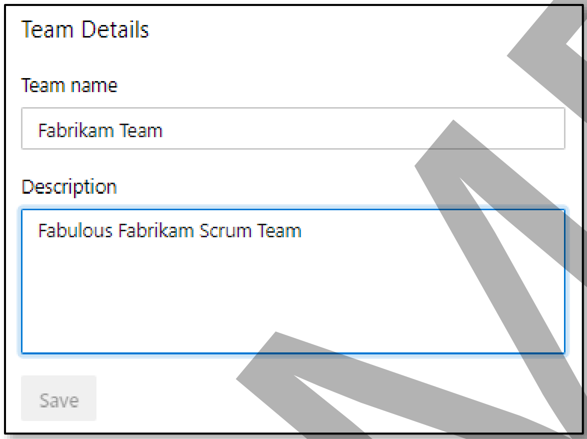
## Task: Update Your Team's Profile (optional)

In this task your team will self-organize and decide who will update the team's description and image.

Who will be performing this task? ____________________________________________

1. Return to your project's home page.
2. Select **Project settings**.
3. Go to the **Teams** page.
4. Select the **Fabrikam Team**.
5. On the **Settings** page, change the Description to **Fabulous Fabrikam Scrum Team** and click **Save**.

   Important: Change the *Description*, not the *Name*.



Team Details

Team name: Fabrikam Team

Description: Fabulous Fabrikam Scrum Team

6. Click the project image placeholder and change it to something more appropriate.

   Sample images can be found in *C:\Course\Labs\Setup\Photos\Team*.

## EXERCISE 3 – CONFIGURE ITERATIONS AND AREAS

### Task: Configure Iterations

In this task your team will self-organize and configure the iterations (sprints). Try to split the work up so that all team members can participate.

1. Return to your project's home page.

   If prompted, enter your Microsoft Account and password.

2. Select **Project settings**.

3. In the **Boards** section, select **Project configuration**.

   You should be on the *Iterations* page by default.

   How many *sprints* have been created for you? ________________________________________

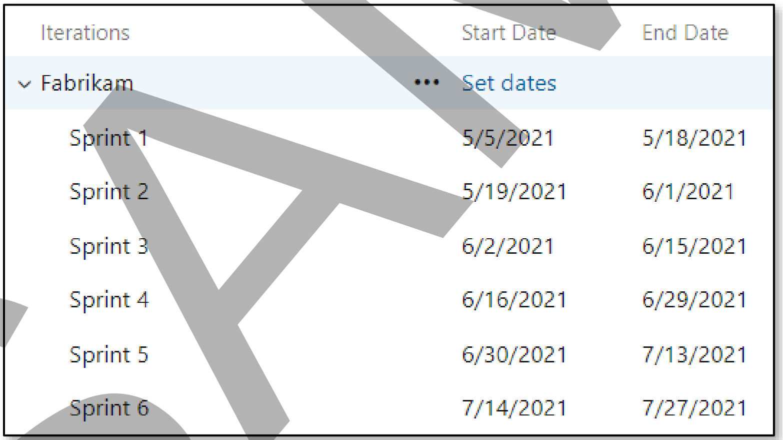
4. As a team, decide how long each sprint should be and what day of the week they should start.

   Note: Sprint lengths are typically 1-4 weeks, with 2 weeks being the most popular.

   How long will your sprints be? What day of the week will they start? __________________________

5. As a team, edit the sprints, setting the dates accordingly.

   Important: Make sure that *Sprint 1* has ended and *Sprint 2* contains "today" in its date range, such as this example (created on 20 May 2021):



| Iterations | Start Date | End Date |
|---|---|---|
| Fabrikam | Set dates | |
| Sprint 1 | 5/5/2021 | 5/18/2021 |
| Sprint 2 | 5/19/2021 | 6/1/2021 |
| Sprint 3 | 6/2/2021 | 6/15/2021 |
| Sprint 4 | 6/16/2021 | 6/29/2021 |
| Sprint 5 | 6/30/2021 | 7/13/2021 |
| Sprint 6 | 7/14/2021 | 7/27/2021 |

## Task: Select Iterations for Your Team

In this task someone on your team will select the first two sprints.

Who will be performing this task? ____________________________________________________________

1. In the **Boards** section of Project Settings, select **Team configuration**.

2. Click **Iterations**.

   What is the *Default iteration*? ____________________________________________________________

   How many iterations are listed at the bottom? _______________________________________________

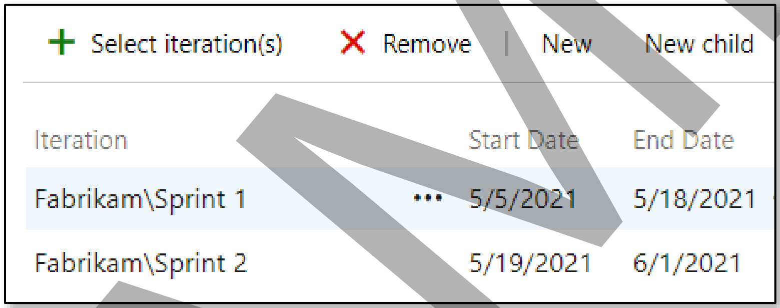
3. Change the Default iteration to **Fabrikam**.

   This way, new work items won't be added to the current sprint.

4. At the bottom, remove **Fabrikam\Sprint 3** and higher iterations.

   Note: This won't remove the iterations themselves. It will just remove them from view.

   You should now have two sprints selected for your team: Sprint 1 which has ended, and Sprint 2 which is the current sprint:

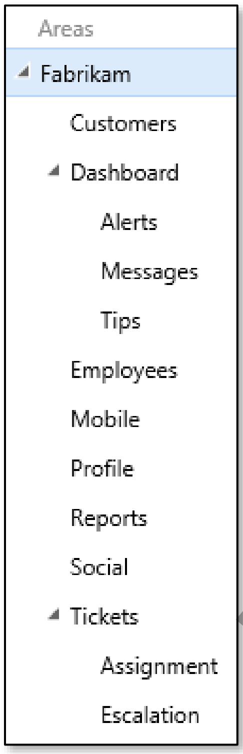


| Iteration | Start Date | End Date |
|---|---|---|
| Fabrikam\Sprint 1 | 5/5/2021 | 5/18/2021 |
| Fabrikam\Sprint 2 | 5/19/2021 | 6/1/2021 |

## Task: Configure Areas

In this task your team will self-organize and configure the hierarchy of areas. Try to split the work up so that all team members can participate.

1. In the **Boards** section of Project Settings, select **Project configuration**.
2. Click **Areas**.
3. As a team, create a hierarchy that looks like this:



## Task: Select Areas for Your Team

In this task someone on your team will select all areas for your team.

Who will be performing this task? ______________________________________________________________

1. In the **Boards** section of Project Settings, select **Team configuration**.
2. Click **Areas**.
3. At the bottom, right-click on the **Fabrikam** (default area) and select **Include sub areas**.
4. Click **OK** to confirm the setting.

# Lab 5: Create and Refine a Product Backlog

Delivering Products Using Azure DevOps and Scrum

## LAB OVERVIEW

In this lab your team will use Azure Boards to create and refine a Product Backlog.

Estimated time to complete this lab: **60 minutes**

### Task Execution

As this is a team-based training course, there are a number of opportunities for team members to learn to collaborate more effectively. Unfortunately, there is a possibility for team members to accidentally impede, block, or otherwise cause unintentional conflicts. To minimize the possibility of conflicts, critical tasks in this course have been marked with an icon indicating who on the team should execute the task:

- ●●● The team can self-organize and execute the task however they decide
- ●○○ Only the “leader” should execute this task
- ○●● Only the “followers” (not the leader) should execute this task
- ● Everyone on the team should execute this task
- ● ● Everyone on the team should execute this task (working in pairs)

Tip: Look for the “leader” ●○○ tasks and ensure that they are only performed once per team. Also, ensure that the “follower” ○●● tasks are only performed by everyone else (not the leader).

### Teams of One

If you are working by yourself and not on a team, make sure to perform all of the “leader” ●○○ tasks, and none of the “follower” ○●● tasks. This scenario is common for students learning remotely.

## EXERCISE 1 – ADD DESIREMENTS TO THE PRODUCT BACKLOG

### Task: Triage a Batch of Desirements

In this task your team will discuss the following disorganized list and decide what kind of "desirement" each item is (user story, epic, bug, or noise). A couple of examples have been provided.

| DESIREMENT | What is this? | MoSCoW |
|---|---|---|
| 1. Back up the database every night | NOISE | |
| 2. Customer can look up television lineup online | | |
| 3. Add ability to export reports to PDF | | |
| 4. Customer can sign in using Facebook credentials | | |
| 5. Integrate with other systems | | |
| 6. Service reps can view ticket details from dashboard | USER STORY | |
| 7. Mobile technicians can locate nearest hardware store | | |
| 8. Customers with Canadian addresses not displaying properly | | |
| 9. Beat Contoso Cable | | |
| 10. Ensure stakeholders' needs are met | | |
| 11. Customer can sign up for emails about service outages | | |
| 12. Support phone devices for our executives | | |
| 13. Technician can report busy/late on phone | | |
| 14. Customer can become a Fan of Fabrikam Fiber on Facebook | | |
| 15. Customer can pay invoices online | | |
| 16. Improve mobile support | EPIC | |
| 17. Technician can check on parts orders on phone | | |
| 18. Technician can see service tickets on phone | | |
| 19. Customer can find the nearest Fabrikam Fiber location | | |
| 20. Customer can reschedule appointments | | |
| 21. Synchronize customer changes with our Dynamics CRM | | |
| 22. Mobile technicians can update customer contact details | | |
| 23. Reports are not working | BUG | |
| 24. Go paperless | | |
| 25. Improve customer service | | |

1. Working as a team, note which items in the above list are user stories, bugs, epics (really large user stories), or just noise.

   How many bugs did you identify? User stories? Epics? ______________________________________

## Task: Use MoSCoW Method to Prioritize Items

In this task your team will use the *MoSCoW* method to classify the items in your list.

1. Elect a Product Owner.

   This person will be the "voice of the customer" during the rest of these exercises. He or she will also have the final say in the description and business value of the items in the Product Backlog, including the ordering of those items.

   Who is your team's Product Owner? ____________________________________________

2. As a team, review each user story/bug (excluding epics) and classify it as one of the following …

   - Must have
   - Should have
   - Could have
   - Won't have

   Note: MoSCoW is a technique used in management, business analysis, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each item - also known as MoSCoW prioritization or MoSCoW analysis. For more information visit http://en.wikipedia.org/wiki/MoSCoW_method.

   How many "Must have" items did you identify? "Won't have" items? ____________________

## Task: Disable Other Backlogs

In this task one member of your team will ensure that the Epics and Features backlogs are not visible.

Who will be performing this task? ____________________________________________

1. If necessary, open your browser and navigate to your project's home page.

2. Select **Project settings**.

3. In the **Boards** section, select **Team configuration**.

4. On the **General** page, view the **Backlog navigation levels** settings.

   Which Backlogs are currently enabled (checked)? ________________________________

5. Clear the checkboxes except the lowest-level **Stories** backlog.

6. Return to the Fabrikam home page.

## Task: Create Work Items

In this task your team will self-organize and create work items for the items you just triaged.

1. If necessary, open your browser and navigate to the Fabrikam project home page.

2. From the **Boards** hub, go to the **Backlogs** page.

3. Press **F5** to refresh the page.

   How many items does your Product Backlog currently contain? ______________________________

4. Divide up the items evenly amongst the team members.

   Tip: Working in pairs might be useful for this task.

5. Skipping the *Epic* and *Noise* items, create a **Product Backlog Item** work item for each *User Story* or *Bug* that your identified.

   - Enter the **Title** (or a shortened variation)
   - Assign the work item to the **Product Owner**
   - Select an appropriate **Area**
   - Add a short, meaningful, creative **Description**
   - Add a **Bug** tag for any PBIs that you've deemed are a bug

   Note: If it seems that items are not visible in the backlog, you may have forgotten to "include sub-areas" when your colleague configured team areas in the previous lab.

6. As a team, review the Product Backlog to ensure accuracy.

   - Rename any items to make them more clear
   - All items should be in the *New* state
   - Every item should have a non-root *Area* (e.g. *Fabrikam\Mobile* – create new areas as necessary)
   - Every item should be in the root *Iteration* (*Fabrikam*)

   Note: If there are any disputes, your Product Owner has the final say

## EXERCISE 2 – IMPLEMENT A DEFINITION OF "READY"

For more background on this technique, read the blog post *Using the Kanban Board to Implement a Definition of Ready* at https://bit.ly/3wwAHJH.

### Task: Customize the Board

In this task one team member will customize the Board by adding additional columns in order to implement the following definition of "ready" based loosely on INVEST (http://bit.ly/1TxK01e):

- ✓ Product Owner is interested
- ✓ Priority assigned
- ✓ Business value assigned
- ✓ Sized
- ✓ Ready

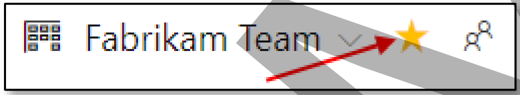
Who will be performing this task? ______________________________________________

1. If necessary, open your browser and navigate to the Fabrikam project home page.

2. From the **Boards** hub, go to the **Boards** page.

3. If prompted, select the **Fabrikam Team Boards**.

   If you want, you can make this board a favorite, to find it quicker the next time you are looking for a specific board. This may make sense to do if you are working in an environment with many teams.

4. Review the items in the board.

   What are the current columns? ______________________________________________

   By default, the columns match the states of the work items on the board.

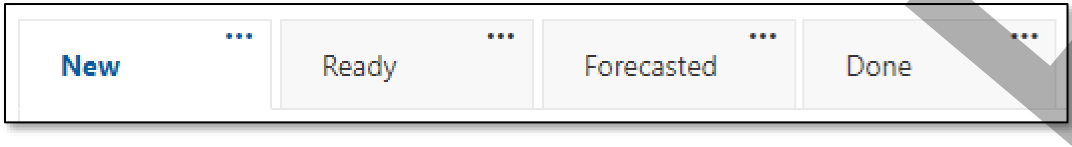
5. Click ⚙ in the upper right to configure team settings.

6. Go to the **Columns** page.

7. Remove any columns listed besides **New**, **Ready**, **Forecasted**, and **Done**.

   Sometimes columns from disabled work item types (e.g. bugs) will show in here.

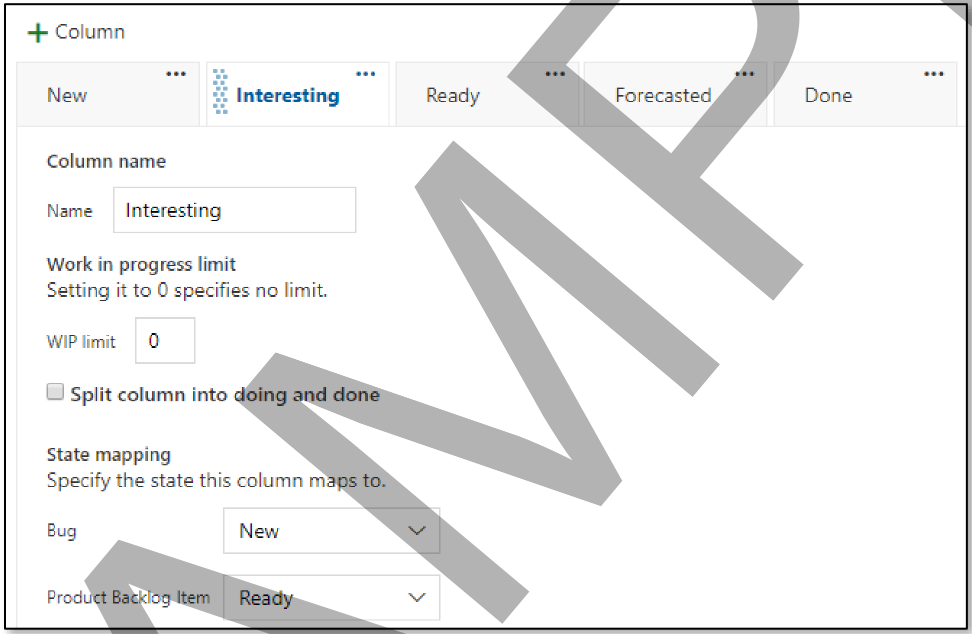
   Your columns should look like this:



8. Right-click on the **New** column and select **Insert right**.

9. Name the column **Interesting**, set the WIP limit to **0**, and select the **Ready** state for the Product Backlog Item type.

   You can ignore the Bug work item type mapping because we are not using the bug work item type.



Column name

Name: Interesting

Work in progress limit
Setting it to 0 specifies no limit.

WIP limit: 0

Split column into doing and done

State mapping
Specify the state this column maps to.

Bug: New

Product Backlog Item: Ready

10. Right-click on the **Interesting** column and select **Insert right**.

11. Name the column **Prioritized**, set the WIP limit to **0**, and ensure the **Ready** state is selected for the Product Backlog Item type.

12. Repeat the above steps to add a **Has Value** and **Sized** columns to the right of Prioritized.

    Make sure to set the WIP limit to 0, and ensure the Ready state for the Product Backlog Item type.

13. Select the **Ready** column and set its WIP limit to **0**.

14. Select the **Forecasted** column and set its WIP limit to **0**.

15. Click **Save and close**.

    Your customized board should now have these columns:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| New | Interesting | Prioritized | Has Value | Sized | Ready | Forecasted | Done |

## Task: Identify the Interesting Items

In this task your team will collaborate to identify the "interesting" work items.

1. If necessary, open your browser and navigate to the Fabrikam project home page.

2. From the **Boards** hub, go to the **Boards** page, and then to the **Fabrikam Team Boards**.

3. Press **F5** to refresh the page.

   How many items are in the *New* state? ______________________________________________

4. Click View options in the upper right and then turn **Live updates** on.

   As you work as a team during the next few tasks, rather than pressing F5 to refresh the board periodically, the *Live Updates* feature will update your board immediately when anyone in your team creates, updates, or deletes a work item on the board. Behind the scenes, server-side code pushes content to all of the connected clients as it happens, in real-time.

5. As a team, decide which items sound "interesting", and then drag them into the **Interesting** column.

   Tip: Use the MoSCoW values that you recorded earlier to decide which ones are interesting. Make sure to leave the "won't haves" in the New column. Also, don't worry about the order of the items in the columns – we'll order the backlog in a bit.

   How many items are in the *Interesting* column now? *New* column? __________________________

## Task: Prioritize the Interesting Items

In this task your team will prioritize the "interesting" items by using each item's MoSCoW value.

1. As a team, edit each **Interesting** item and add a **Must**, **Should**, or **Could** tag as appropriate.

   You can do this by clicking a card's title, adding the tag at the top, and clicking Save & Close.

2. After adding all the tags, drag the **Must** items to the **Prioritized** column, leaving 1 or 2 items.

3. Next, drag the **Should** items to the **Prioritized** column, leaving 1 or 2 items.

4. Next, drag the **Could** items to the **Prioritized** column, leaving 1 or 2 items.

5. Using drag and drop, order the items within the **Prioritized** column so that the **Must**-haves are at the top, **Should**-haves are in the middle, and **Could**-haves are at the bottom.

   Feel free to change your mind at any time, change the tags, and the order of the items. Just make sure that your Product Owner is satisfied with the Product Backlog.

## Task: Style the Cards

In this task one team member will customize the cards, displaying the Business Value field and adding coloring styles based on the tags.

Tip: Have a different team member do this task than the one who customized the Board view earlier.

1. Click ⚙ in the upper right to configure team settings.

2. On the **Fields** page, click + **Field** and select **Business Value**.

   This will cause the Business Value field to display on the cards on the board. We will be assigning Business Value in a few moments.

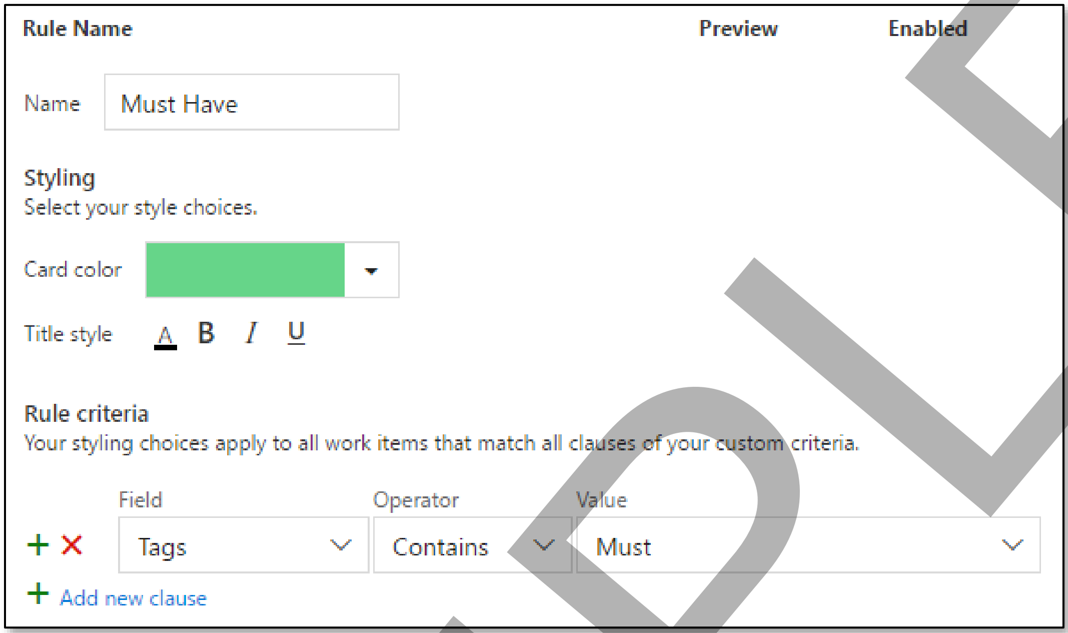
3. If listed, removed the **State** field.

   We can determine the state by the column the card is in.

4. Go to the **Styles** page.

5. Click + **Styling rule**.

6. Name the rule **Must Have**, select a bright green for the **Card color**, and add a rule criterion where **Tags** … **Contains** … **Must**.


**Rule Name** **Preview** **Enabled**

Name: Must Have

**Styling**
Select your style choices.

Card color: 🟩

Title style: A B I U

**Rule criteria**
Your styling choices apply to all work items that match all clauses of your custom criteria.

| Field | Operator | Value |
|---|---|---|
| Tags | Contains | Must |

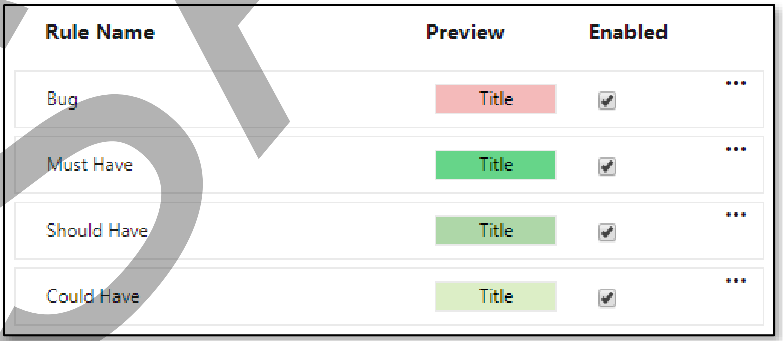
7. Click **+ Styling rule** and add a **Should Have** rule with a medium green card color and a criterion where **Tags** … **Contains** … **Should**.

8. Click **+ Styling rule** and add a **Could Have** rule with a light green card color and a criterion where **Tags** … **Contains** … **Could**.

9. Click **+ Styling rule** and add a **Bug** rule with a medium red card color and a criterion where **Tags** … **Contains** … **Bug**.

   Even though we are using the PBI work item type, we can still make the bugs stand out by styling them a different way. We could also have made additional styling rules for dark red and light red based on their Business Value.

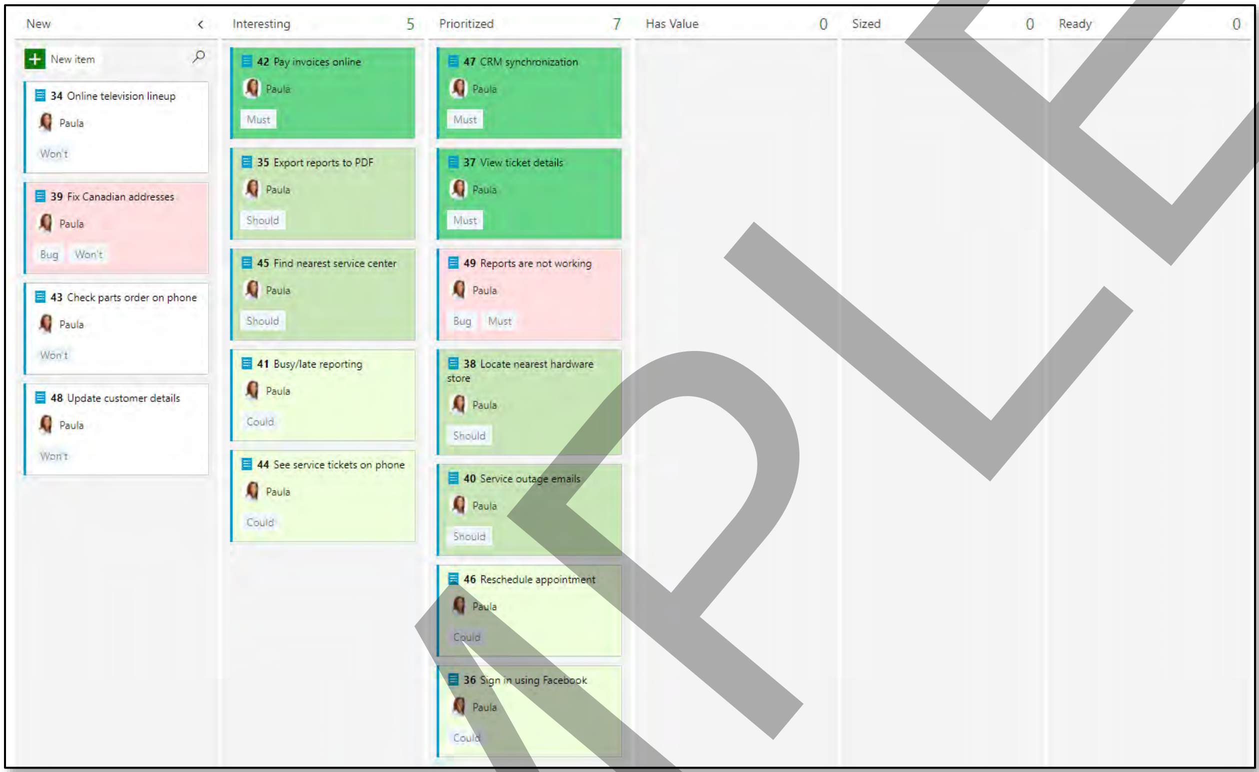
10. Drag and drop the styles so that they are in this order:



| Rule Name | Preview | Enabled |
|---|---|---|
| Bug | Title | ✓ |
| Must Have | Title | ✓ |
| Should Have | Title | ✓ |
| Could Have | Title | ✓ |

11. Click **Save and close**.

    Now anyone looking at the board can quickly see which items are higher priority/value:



New <
New item
34 Online television lineup
Paula
Won't
39 Fix Canadian addresses
Paula
Bug Won't
43 Check parts order on phone
Paula
Won't
48 Update customer details
Paula
Won't

Interesting 5
42 Pay invoices online
Paula
Must
35 Export reports to PDF
Paula
Should
45 Find nearest service center
Paula
Should
41 Busy/late reporting
Paula
Could
44 See service tickets on phone
Paula
Could

Prioritized 7
47 CRM synchronization
Paula
Must
37 View ticket details
Paula
Must
49 Reports are not working
Paula
Bug Must
38 Locate nearest hardware store
Paula
Should
40 Service outage emails
Paula
Should
46 Reschedule appointment
Paula
Could
36 Sign in using Facebook
Paula
Could

Has Value 0

Sized 0

Ready 0

## Task: Assign Business Value to the Prioritized Items

In this task your team will assign a business value number to the "prioritized" items.

1. Review the items on the **Board**.

    How many items are in the Prioritized column? How many are Bugs? ___________________________

2. As a team, edit each "Prioritized" **PBI** work item and set its **Business Value** accordingly:

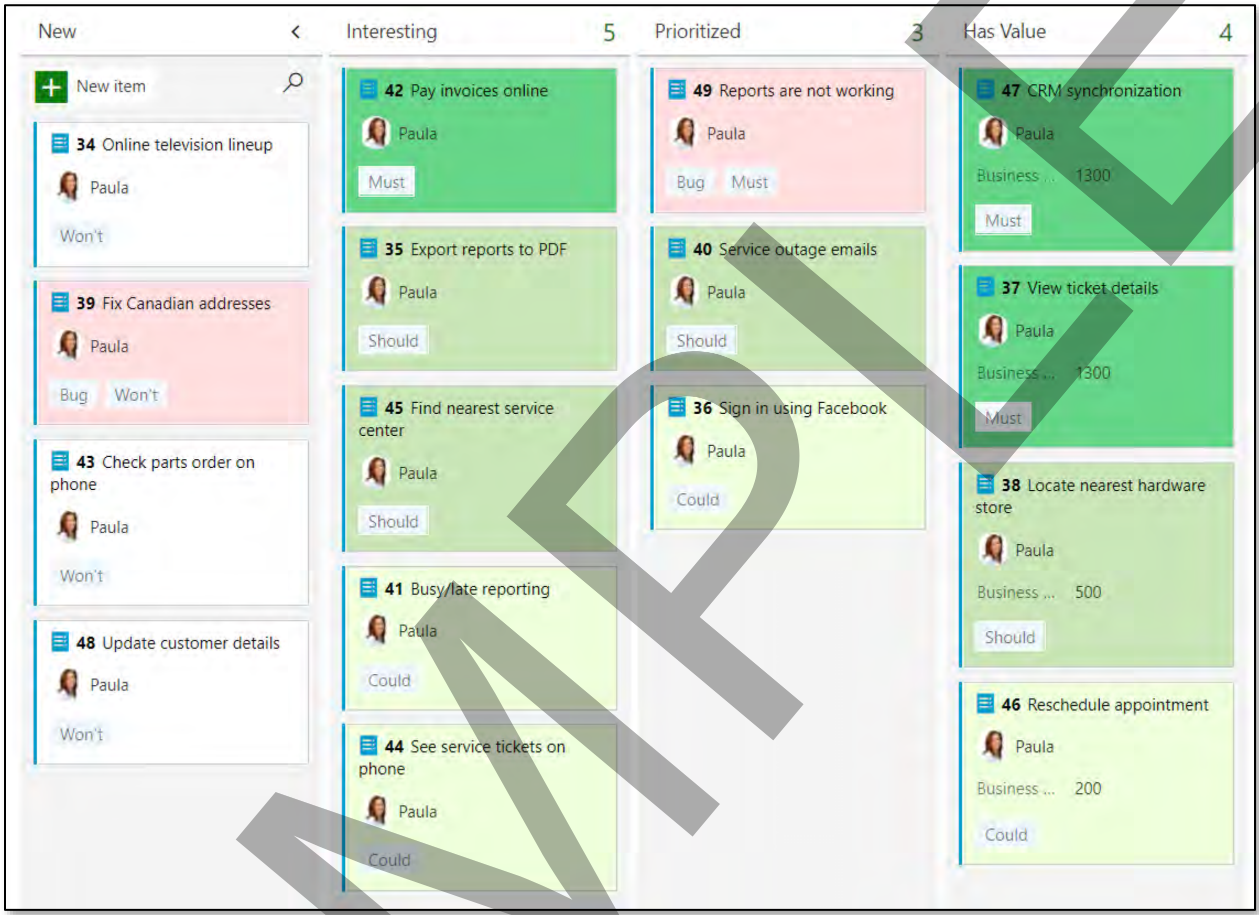
    - *1300* for *Must*-have PBI work items
    - *500* for *Should*-have PBI work items
    - *200* for *Could*-have PBI work items

3. After setting the **Business Value** of an item, drag the item to the **Has Value** column.

4. Leave one or two PBIs without a **Business Value** in the **Prioritized** column.

5. Reorder the cards (work items) in the **Has Value** column if necessary.

   Your board might look something like this now:



New

New item

**34** Online television lineup
Paula
Won't

**39** Fix Canadian addresses
Paula
Bug Won't

**43** Check parts order on phone
Paula
Won't

**48** Update customer details
Paula
Won't

Interesting 5

**42** Pay invoices online
Paula
Must

**35** Export reports to PDF
Paula
Should

**45** Find nearest service center
Paula
Should

**41** Busy/late reporting
Paula
Could

**44** See service tickets on phone
Paula
Could

Prioritized 3

**49** Reports are not working
Paula
Bug Must

**40** Service outage emails
Paula
Should

**36** Sign in using Facebook
Paula
Could

Has Value 4

**47** CRM synchronization
Paula
Business ... 1300
Must

**37** View ticket details
Paula
Business ... 1300
Must

**38** Locate nearest hardware store
Paula
Business ... 500
Should

**46** Reschedule appointment
Paula
Business ... 200
Could

## EXERCISE 3 – REFINE THE PRODUCT BACKLOG

### Task: Perform Relative Estimation

In this task your team will estimate the size of the work items that have "value".

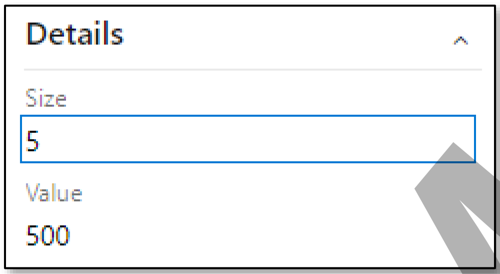
1. Review the items on the **Board**.

   How many items are in the *Has Value* column? How many are Bugs? ___________________________

2. As a team, identify one of the "Has Value" items which is of "medium" size (complexity/effort).

   Note: This will be a wild guess, especially for a team that's never met before today, working on a product they didn't develop and have never used, and using brand new tools. Do your best!

3. Have someone on the team edit that "baseline" work item and set its **Size** to **5**.



Details

Size

5

Value

500

4. Have someone on the team start a 5-minute timer and begin estimating.

   You have a 5-minute timebox to estimate as many items as you can in the "Has Value" column:

   - Starting at the top of the column, open an item and briefly discuss it with the rest of the team
   - Keeping in mind that the baseline work item size is a 5, vote if this item is "smaller" (e.g. a 3, 2, or 1) or "larger" (e.g. a 8, 13, 21, or huge); huge items would need to be broken down into smaller items in order to be delivered in a single sprint (which is beyond the scope of this class)
   - Keep discussing and voting until consensus is reached, or the group has tried for 3 times, or the timebox has expired
   - If consensus was reached (or an average can be agreed upon), enter it into the *Size* field
   - Repeat until all work items have been sized or the timebox has expired

   Tip: You can use, or quickly make, Planning Poker cards. There are also free apps for your phone – just search on "Planning Poker", "Scrum Poker", or "Agile Poker" in your app store.

5. After setting the **Size** value, drag the item to the **Sized** column.

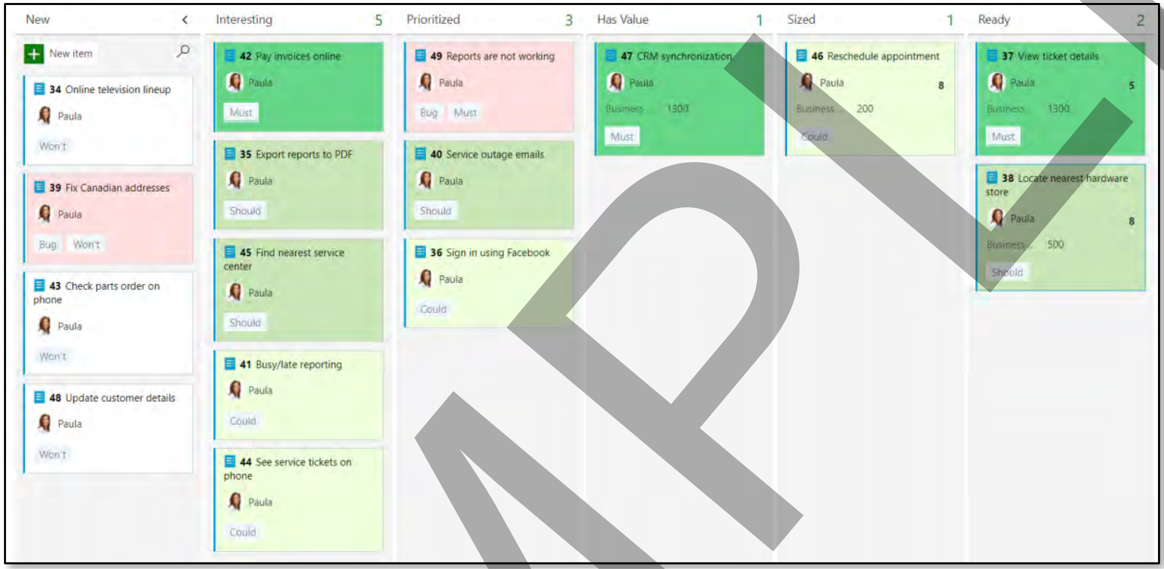
6. Leave 1 or 2 items without a **Size** in the **Has Value** column.

## Task: Order the Product Backlog Based on ROI (optional)

In this task your team, as directed by your Product Owner, will order the Product Backlog based on the Return On Investment (ROI) of each item in the Sized column.

1. Reorder the items in the **Sized** column based on ROI (**Business Value** / **Effort**).
2. Drag all 1 or 2 items from the **Sized** column to **Ready**.

   Your board might look something like this now:



New
New item
34 Online television lineup
Paula
Won't
39 Fix Canadian addresses
Paula
Bug Won't
43 Check parts order on phone
Paula
Won't
48 Update customer details
Paula
Won't

Interesting 5
42 Pay invoices online
Paula
Must
35 Export reports to PDF
Paula
Should
45 Find nearest service center
Paula
Should
41 Busy/late reporting
Paula
Could
44 See service tickets on phone
Paula
Could

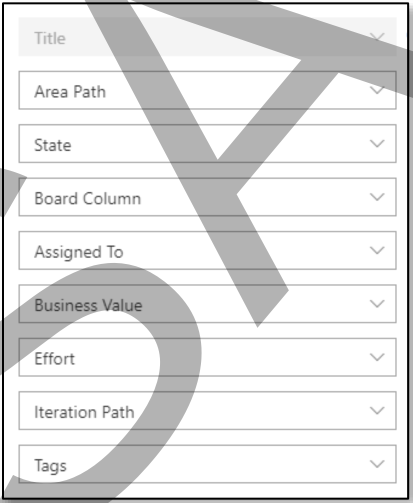
Prioritized 3
49 Reports are not working
Paula
Bug Must
40 Service outage emails
Paula
Should
36 Sign in using Facebook
Paula
Could

Has Value 1
47 CRM synchronization
Paula
Business ... 1300
Must

Sized 1
46 Reschedule appointment
Paula 8
Business ... 200
Could

Ready 2
37 View ticket details
Paula 5
Business ... 1300
Must
38 Locate nearest hardware store
Paula 8
Business ... 500
Should

3. View the **Backlog**.
4. Select **Column options** at the top of the Backlog.
5. Add, remove, and move columns until they look like this:

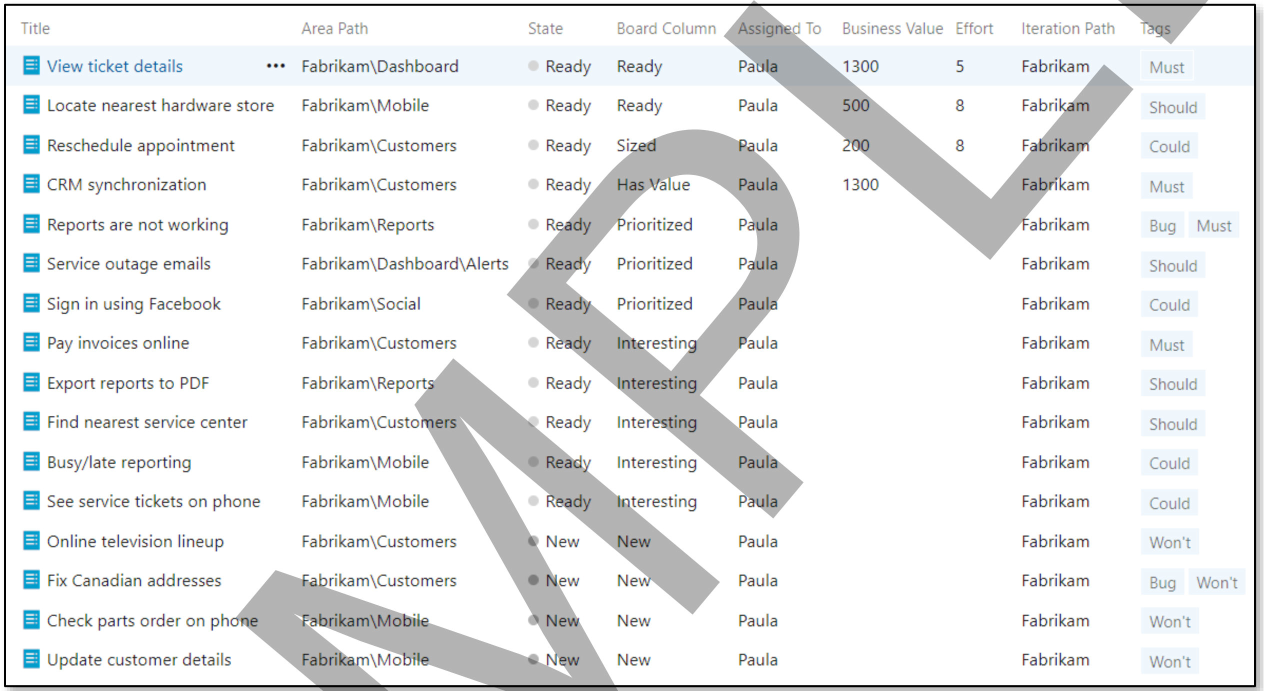


Title
Area Path
State
Board Column
Assigned To
Business Value
Effort
Iteration Path
Tags

A few notes and guidance about Product Backlogs …

- You can remove the *Work Item Type* column to gain some extra real estate
- The *Board Column* shows the virtual board column that this item sits in
- All *Assigned To* fields should be set to your Product Owner, or left blank
- All *Ready* state items should generally be ordered above *New* state items

6. Reorder the Product Backlog by Board Column so that the **Ready** items are at the top, followed by **Sized**, **Has Value**, **Prioritized**, **Interesting**, and finally **New** items, like in this example:



| Title | Area Path | State | Board Column | Assigned To | Business Value | Effort | Iteration Path | Tags |
|---|---|---|---|---|---|---|---|---|
| View ticket details | Fabrikam\Dashboard | Ready | Ready | Paula | 1300 | 5 | Fabrikam | Must |
| Locate nearest hardware store | Fabrikam\Mobile | Ready | Ready | Paula | 500 | 8 | Fabrikam | Should |
| Reschedule appointment | Fabrikam\Customers | Ready | Sized | Paula | 200 | 8 | Fabrikam | Could |
| CRM synchronization | Fabrikam\Customers | Ready | Has Value | Paula | 1300 | | Fabrikam | Must |
| Reports are not working | Fabrikam\Reports | Ready | Prioritized | Paula | | | Fabrikam | Bug Must |
| Service outage emails | Fabrikam\Dashboard\Alerts | Ready | Prioritized | Paula | | | Fabrikam | Should |
| Sign in using Facebook | Fabrikam\Social | Ready | Prioritized | Paula | | | Fabrikam | Could |
| Pay invoices online | Fabrikam\Customers | Ready | Interesting | Paula | | | Fabrikam | Must |
| Export reports to PDF | Fabrikam\Reports | Ready | Interesting | Paula | | | Fabrikam | Should |
| Find nearest service center | Fabrikam\Customers | Ready | Interesting | Paula | | | Fabrikam | Should |
| Busy/late reporting | Fabrikam\Mobile | Ready | Interesting | Paula | | | Fabrikam | Could |
| See service tickets on phone | Fabrikam\Mobile | Ready | Interesting | Paula | | | Fabrikam | Could |
| Online television lineup | Fabrikam\Customers | New | New | Paula | | | Fabrikam | Won't |
| Fix Canadian addresses | Fabrikam\Customers | New | New | Paula | | | Fabrikam | Bug Won't |
| Check parts order on phone | Fabrikam\Mobile | New | New | Paula | | | Fabrikam | Won't |
| Update customer details | Fabrikam\Mobile | New | New | Paula | | | Fabrikam | Won't |

## EXERCISE 4 – CREATE A STORY MAP (OPTIONAL)

### Task: Install the SpecMap Extension

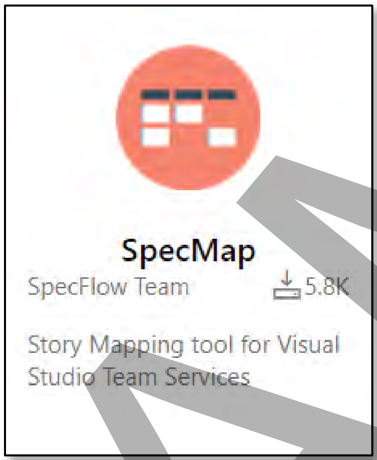
In this task the user who created the Azure DevOps Services organization will install the SpecMap extension.

SpecMap is a story mapping tool for Azure Boards that allows you to build visual story maps from work items and helps organize your Scrum Team. Story maps provide a great basis for discussing the needs of your users, and prioritizing development to deliver the biggest impact.

Who will be installing this extension? ______________________________________________

Tip: Another user can be working on the next task while this task is being performed.

1. If necessary, launch **Chrome** and navigate to the Fabrikam project.
2. Click the icon in the upper right and select **Browse Marketplace**.
3. Search for and select the **SpecMap** extension by **TechTalk Software**.



4. Install the **SpecMap** extension into your Azure DevOps Services organization.
5. Return to the Fabrikam project and go to the **SpecMap** page on the **Boards** hub.
6. Follow the instructions to sign-in and create an account.
7. Return to the **SpecMap** page on the **Boards** hub and refresh the page.

## Task: Create a Product Backlog query

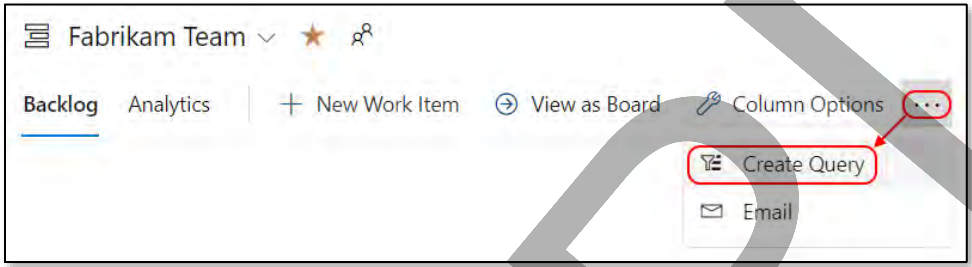
In this task someone on the team will create a Product Backlog query to be used by SpecMap.

Who will be creating the query? ______________________________________________

1. From the **Boards** hub, go to the **Backlogs** page.

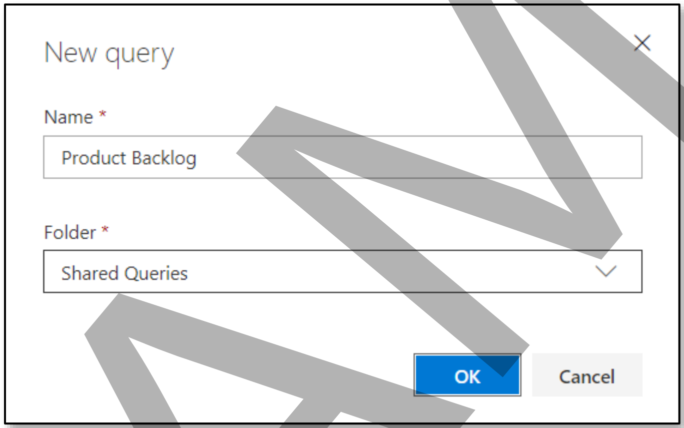
   How many PBIs are listed? ______________________________________________

2. Click the **...** more options dropdown, select **Create Query**.



   This will create a query that returns the Product Backlog Items formatted just the way you see it.

3. Rename the query **Product Backlog** and save it to the **Shared Queries** folder.



New query

Name *

Product Backlog

Folder *

Shared Queries

OK
Cancel

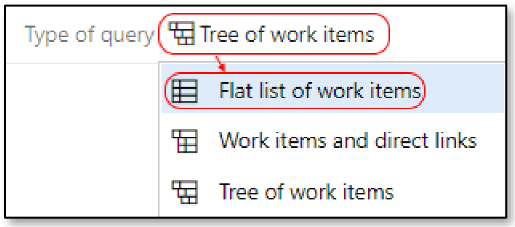
   This way everyone on the team has access to it.

4. From the **Boards** hub, go to the **Queries** page.
5. View **All** queries.
6. Click the new **Product Backlog** query to run it.

   How many PBIs are listed? ______________________________________________

7. Click **Editor** at the top to edit the query.

8. Click **Tree of work items** and select **Flat list of work items**.



This will make the query much simpler. You can click *Run query* to see if there's any difference.

9. Click **Save query**.

Have your teammates go to the Queries page and try out the new query.

## Task: Create a Story Map

In this task your team will self-organize and use SpecMap to create one or more story maps.

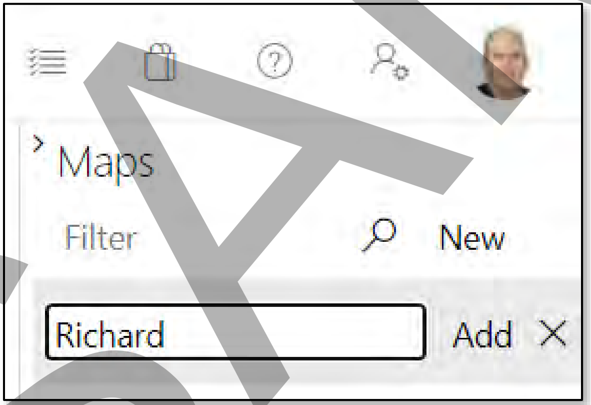
<u>Dependencies</u>

- The *SpecMap* extension has been installed into your Azure DevOps Services organization
- The *Product Backlog* shared query has been created

1. Press **F5** to refresh the page.

2. From the **Boards** hub, go to the **SpecMap** page.

3. Create a **New** map and name it ***<your name>***, like in this example:



<u>Note</u>: A map's name must be unique on the team.

4. Rename the *First user activity* column to **Social**.

Activities can be mapped into activities such as epics, features, themes, or journeys. In this map we will simply use a few product areas to represent activities.

5. Add a new user activity named **Customer**.

6. Add two more activities named **Reports** and **Mobile**.

7. Drag and drop the activities so they are arranged alphabetically.

8. Rename the *First slice* column to **Release1**.

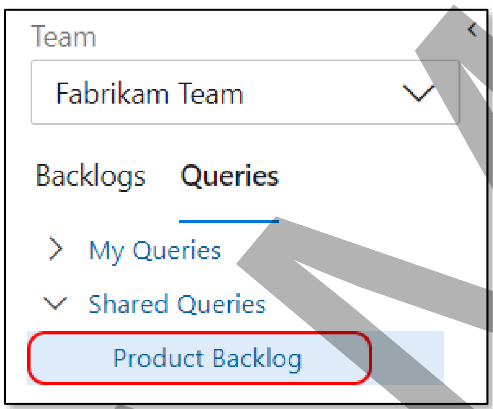
   These horizontal rows represent increasing sophistication in the product, such as versions, releases, or deployments. In this map we will simply use a couple of abstract "Release" slices.

9. Add a new slice named **Release2**.

10. If necessary, drag **Release2** so that it is below **Release1**.

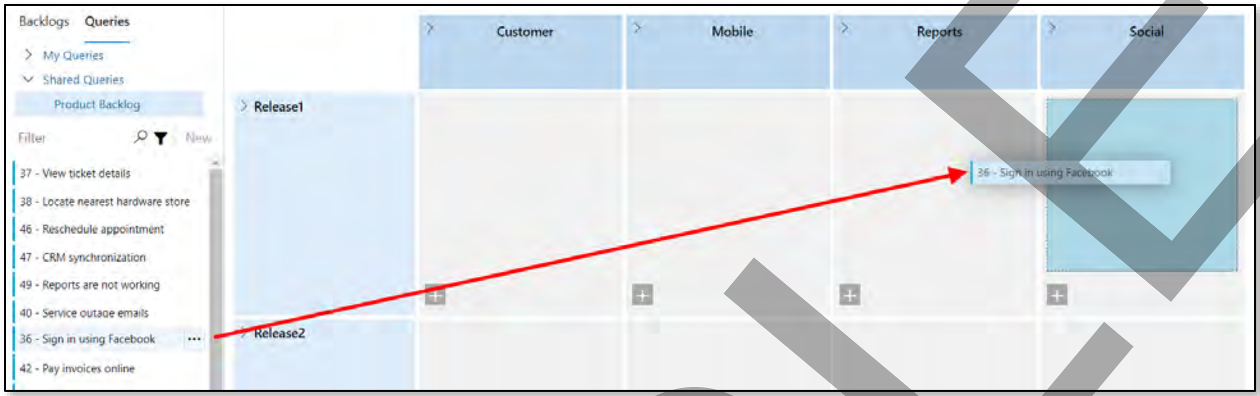
11. Select **Queries** at the top left.

12. Under **Shared Queries**, click the **Product Backlog** query.



13. Drag a Product Backlog Item from the left into the appropriate activity for **Release1**.

    Here is an example:



14. Repeat the last step, dragging more work items into appropriate activities and slices as you see fit.

    As you can see, *SpecMap* generates clean, two-dimensional visualizations of the product backlog organized anyway you want.

15. As your teammates complete their maps, open and review them too.