



THE UNIVERSITY OF  
MELBOURNE

# SWEN90016

## Software Processes & Project Management

*Christoph Treude, Andrew Valentine*  
*School of Computing and Information Systems*  
*The University of Melbourne*  
*[christoph.treude@unimelb.edu.au](mailto:christoph.treude@unimelb.edu.au), [andrew.valentine@unimelb.edu.au](mailto:andrew.valentine@unimelb.edu.au)*

2023 – Semester 1

Week 2

Copyright University of Melbourne 2020-2023

## Week 2 – Intended Learning Objectives

~~Module 5 – Software Development  
Lifecycles - Formal.~~

**Module 6 – Software Development  
Lifecycles - Agile.**

# Intended Learning Objectives

## Software Development Lifecycles

### Agile:

1. Understand what Agile is and its origins.
2. Understand the Agile framework.
3. Understand Scrum – Roles, Ceremonies and Artefacts.
4. Understand advantages / disadvantages of Agile.

## SDLCs

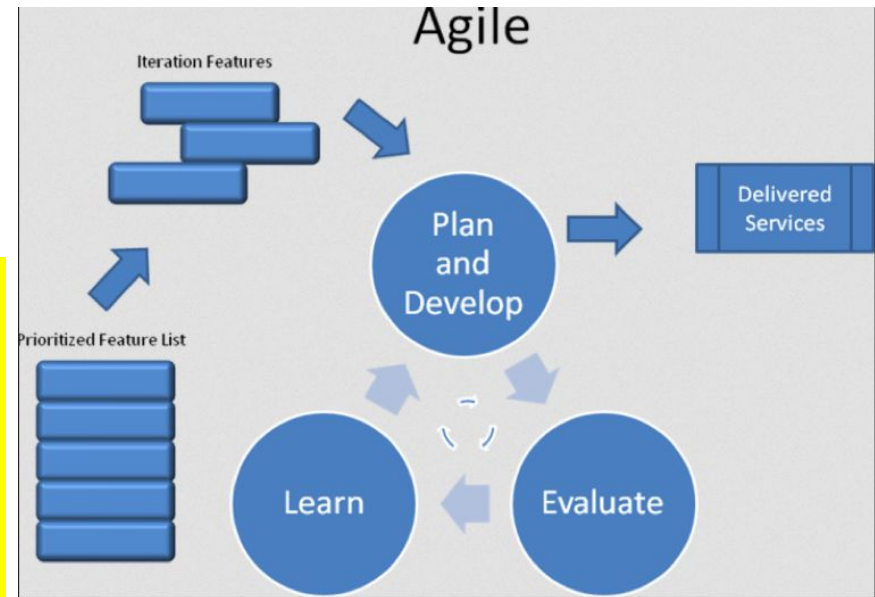
There are many SDLCs around with organisations typically favouring a blend of Formal and Agile approaches.

### 1. Formal

- Waterfall
- Incremental
- V-Model

### 2. Agile

- Kanban
- Scrum
- Extreme Programming



## Why is Agile attractive

- We are in an ever changing global world with the pace of change increasing
- Customer needs and demands are exponentially increasing – products must continually be delivered
- Low Technology cost, ease of use and the global market place has increased competition and reduced entry barriers
- The war for talent is over – and we have lost! Cross functional teams help minimise the potential loss
- Long development cycles are like long lunches – a thing of the past
- Quality is no longer something we do / check later – it must be part of everything we do
- Cross functional groups are more fun!

## What is Agile

- A framework based on iterative development where requirements and solutions evolve through collaboration between self-organising cross-functional teams
- A disciplined process that encourages frequent inspection and adaptation
- A leadership philosophy that encourages teamwork, self-organisation and accountability
- A set of engineering best practices intended to allow for rapid delivery of high-quality software
- A business approach that aligns development with customer needs and company goals

## What is Agile

In software development, we think about methodologies, activities, interactions, results, work products, artefacts and processes to organise the work.

*The main tasks of software development remain the same regardless of methodology used, however with Agile, the flow of activities, how they are undertaken and who is involved is extremely different.*

# What is Agile - Origins

- Changes initiated in large US corporates in 1990's
- Software engineers frustrated with
  - long lead times before products were delivered
  - decisions made early in the project couldn't be changed later
- 17 software engineer thought leaders met first in 2000 to discuss software engineering and different approaches
- Famous meeting in 2001 at the Snowbird ski lodge in Utah where they met to change the way the industry designed, engineered and deployed software
- Brought together by the Agile Manifesto

<https://techbeacon.com/agility-beyond-history%E2%80%94legacy%E2%80%94agile-development>



# Intended Learning Objectives

## Module 8 - Software Development Lifecycles

### Agile:

- ~~1. Understand what Agile is and its origins.~~
2. Understand the Agile framework.
3. Understand Scrum – Roles, Ceremonies and Artefacts.
4. Understand advantages / disadvantages of Agile.

# Agile Framework

Primary elements of the Agile framework include:

- Manifesto
- 12 Key Principles
- Kanban
- Scrum

# Which four does Agile value the most?

Comprehensive documentation

Working software

Following a plan

Responding to change

Processes and tools

Contract negotiation

Individuals and interactions

Customer collaboration

# Agile Framework - *Manifesto*

## Agile Manifesto

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

Individuals & interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

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

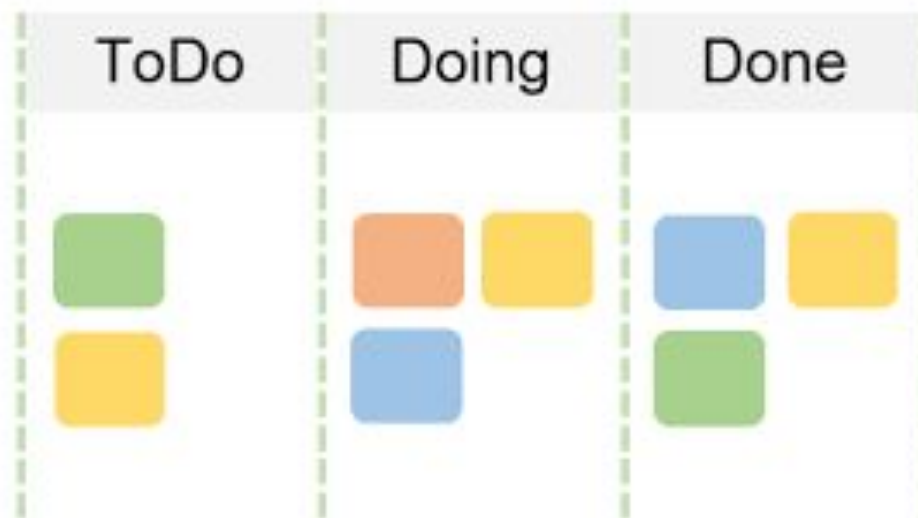
<http://www.agilealliance.org>

## Agile Framework – *12 Key Principles*

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, shorter timeframes is the preference.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need and trust them.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity - the art of maximizing the amount of work not done - is essential.
11. The best architectures, requirements, and designs emerge from self-organising teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

## Agile Framework - *Kanban*

- ***Signboard / Billboard:*** Work items are visualised to provide participants a view of progress and process, from start to finish usually via a Kanban board



# Agile Framework - *Kanban*

Visual progress gives transparency/accountability for self-organizing teams often referred to as **SWIMLANE** boards



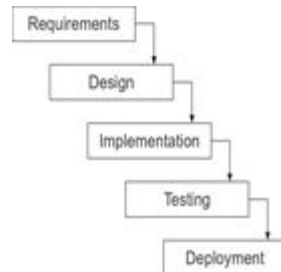
<https://agilefixer.files.wordpress.com/2017/02/swimlane-example-ios.png>

# Agile Framework - *Scrum*

Scrum is an agile way to manage a project

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Hiroataka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”, *Harvard Business Review*, January 1986.





# Intended Learning Objectives

## Module 8 - Software Development Lifecycles

### Agile:

- ~~1. Understand what Agile is and its origins.~~
- ~~2. Understand the Agile framework.~~
3. Understand Scrum – Roles, Ceremonies and Artefacts.
4. Understand advantages / disadvantages of Agile.

# Scrum

## Scrum in 100 words

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two to four weeks).
- The business sets the priorities. Teams self-organise to determine the best way to deliver the highest priority features.
- Every two to four weeks, you can see real working software and decide to release it as is or continue to enhance it for another sprint.

# Scrum is used by many organisations

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

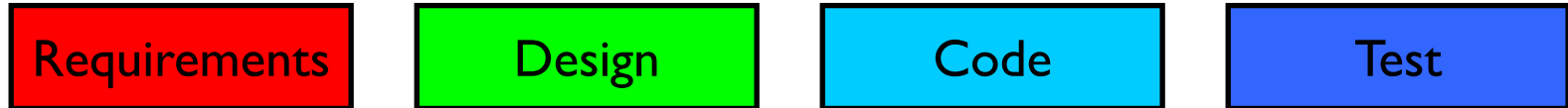
# Scrum is used for all types of projects

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

# Scrum Key Characteristics

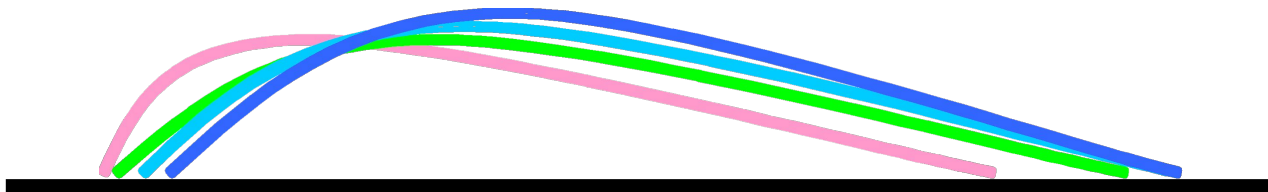
- Self-organising teams
- Product progresses in a series of focused Sprints
- Requirements are captured as items in a Product Backlog list
- Scrum is one of the Agile processes – the one most widely used, discussed and debated
- Time frame is contained to a manageable size (weeks or months)

# Scrum Framework - *Sprints*



Rather than doing one thing at a time...

...Scrum teams do a little of everything all the time



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

# Select all terms you're familiar with

Product Owner

Scrum Master

Sprint Planning Meeting

Sprint Review Meeting

Sprint Retrospective Meeting

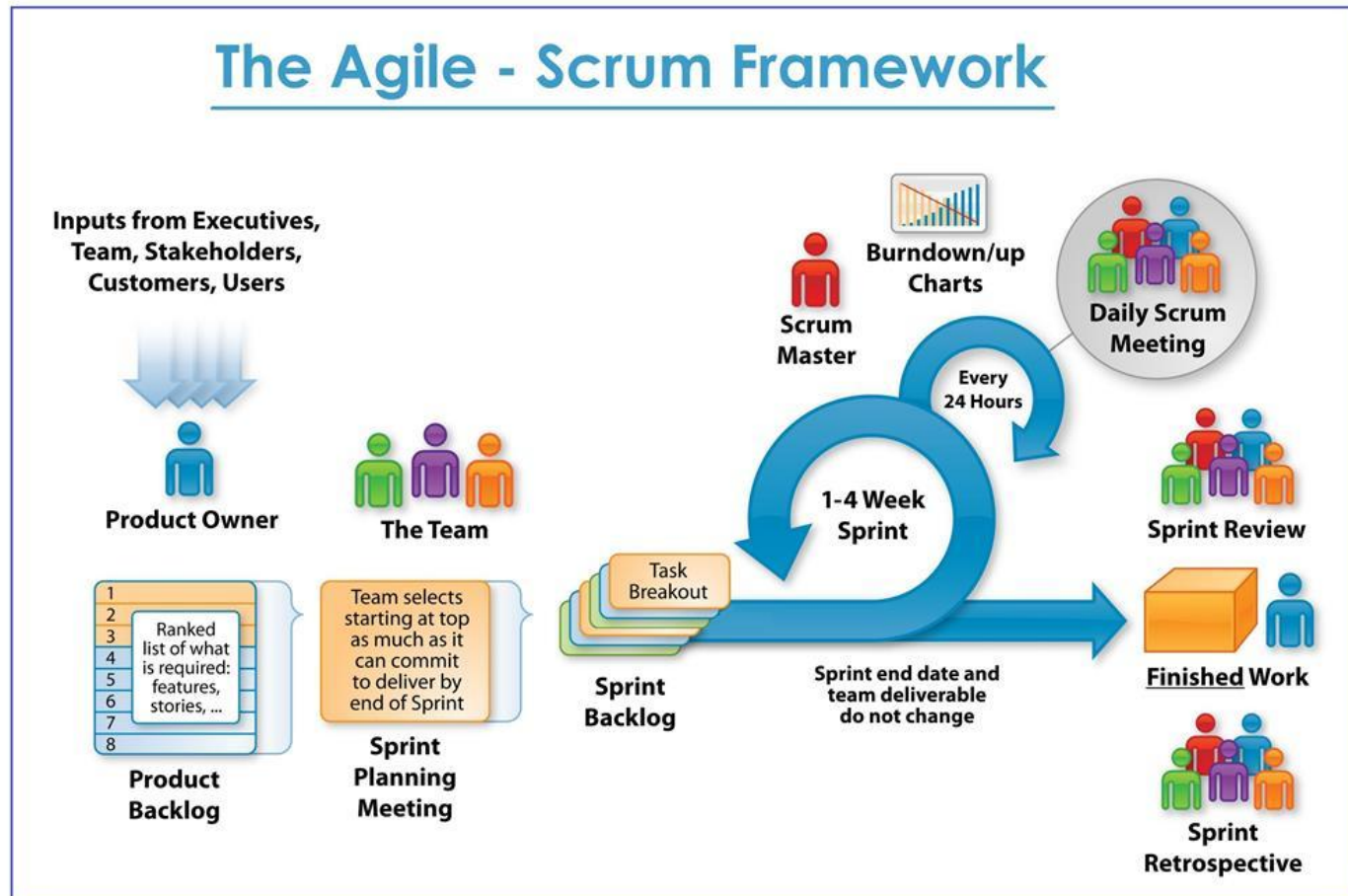
Daily Stand-up Meeting

Product Backlog

Sprint Backlog

Burndown Chart

# Scrum Framework - *Framework*



<https://www.c-sharpcorner.com/UploadFile/d9c992/the-agile-scrum-framework/>



# Scrum Framework

## Roles

- Product owner
- ScrumMaster
- Team

[www.mountaingoatsoftware.com](http://www.mountaingoatsoftware.com)

## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily stand-ups

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

# Scrum Roles

## Roles

- Product owner
- ScrumMaster
- Team

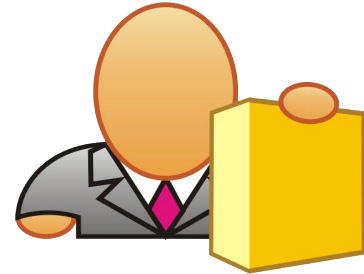
## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily stand-ups

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

## Scrum Roles – *Product Owner*



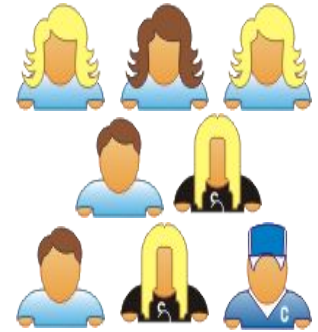
- Defines the features of the product
- Is responsible for the benefits / profitability of the product (ROI)
- Prioritises features according to market value
- Adjusts features and priority every iteration, as needed
- Accepts or reject work results

## Scrum Roles – *Scrum Master*



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments / road blocks
- Ensures that the team is fully functional and productive
- Enables close cooperation across all roles
- Shields the team from external interferences
- Is a member & active participant of the Scrum Team

## Scrum Roles – *The Dev Team*



- Typically 6 - 9 people
- Estimates possible release date of features
- Cross-functional:
  - programmers, testers, user experience designers, business representatives etc.
- Members should be full-time – some exceptions
- Co-located (physically or virtually)

# Scrum Ceremonies / Meetings

## Roles

- Product owner
- ScrumMaster
- Team

## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily stand-ups

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

## Scrum Ceremonies / Meetings

### *Sprint Planning*

- Defines how to achieve Sprint Goal (design)
- Create Sprint Backlog (User Stories) from Product Backlog
- Estimate Sprint Backlog using team velocity and Story Point estimates
- Product Owner priority guides the work
- Release Plan is created
- High-level design is considered

# Scrum Ceremonies / Meetings

## *Sprint Planning Meeting*

### 1<sup>st</sup> Half of the meeting

- Team defines **what** can be done in this sprint
- Starts by writing down the Sprint Goal
- Identify the feature User Story items in the Product Backlog that can achieve this Goal

### 2<sup>nd</sup> Half of the meeting

- Team figures out **how** the work will get done
- Break down each feature user story into detailed Sprint Backlog User Stories, to capture all the work required
- Estimate the detailed User Stories using Story Points to plan and monitor the Sprint progress

Source: *Head First Agile – A Brain-Friendly Guide to Agile Principles, Ideas, and Real-World Practices* By [Andrew Stellman](#), [Jennifer Greene](#)



# Scrum Ceremonies / Meetings

## *Daily Stand-up*

- Parameters
  - Daily
  - 15-minutes and no more than 30 mins
  - Stand-up
- Not for problem solving / Not a status meeting
  - Whole world is invited
  - Only team members, ScrumMaster, Product Owner can **clarify** any questions about the User Stories
- Helps avoid other unnecessary meetings
- 3 key questions asked:
  1. What did I do yesterday.
  2. What will I do today.
  3. What is in my way to get my work completed.



# Scrum Ceremonies / Meetings

## *Sprint Reviews - Showcase*

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
- 2-hour prep time rule
- No slides
- Whole team participates
- Invite the world



## Scrum Ceremonies / Meetings

### *Sprint Retrospective*

- Periodically look at what is and isn't working
- Typically 30 minutes
- Done after every sprint
- Whole team participates:
  - ScrumMaster and Dev Team
- Possibly Product Owner, customers and others (But generally NOT)
- Discuss what to:
  - Start Doing, Stop Doing and Continue Doing

# Scrum Artefacts

## Roles

- Product owner
- ScrumMaster
- Team

## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily stand-ups

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

# Scrum Artefacts – Product Backlog

## User Stories

- A User Story is a requirement expressed from the perspective of an end-user / customer of the system
- User Stories shift the focus from writing about requirements to talking about them
- User Stories are short, simple descriptions of a feature told from the perspective of the customer who wants the new capability of the system. They follow a simple template:

*As a < type of user >, I want < some goal > so that < some reason >*

**As a < type of user >, I want < some goal > so  
that < some reason >**

# Scrum Artefacts – Product Backlog

## User Stories - (Feature or Epic)

- User Stories are written at varying levels of detail.
- They can cover a large amounts of functionality, for example from a Professional Membership Website:

*As a site visitor,*

*I want to get all information associated with my professional membership*

*so that I have access to all information centrally.*

- This level of detail is too large for an agile team to complete in one iteration, so split it into smaller user stories before it is worked on

# Scrum Artefacts – Product Backlog

## Story Points

- Story Points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work
- Story Points help estimate how much work can be done in a Sprint
- A story that is assigned a 2 should be twice as much effort as a story that is assigned a 1, and two-thirds that is estimated as a 3 story point.
- When estimating with Story Points, a value is assigned to each User Story. The raw values are unimportant.
- Instead of assigning 1, 2 and 3, that dev team could assign 100, 200 and 300. Or 1 million, 2 million and 3 million. It is the relative ratios that matter.



# Scrum Artefacts – Product Backlog

## Product Backlog



User Story 1

User Story 2

User Story 3

User Story 4

User Story 5

User Story nn

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Product Backlog User Stories are selected for a Sprint by Product Owner
- Reprioritised at the start of each sprint

# Scrum Artefacts – Product Backlog

## Example - Professional Body Website

### Product Backlog

User Story 1

User Story 2

User Story 3

User Story 4

User Story 5

User Story 6

### *News Section – Sprint 1*

- User Story 1 - As a site visitor, I can read current news on the home page so that I stay current on key professional items
- User Story 2 - As a site visitor, I can email news items to the editor so that they can be considered for publication
- User Story 3 - As a site member, I can subscribe to an RSS feed of news so that I can stay current on the news that is of interest to me

### *Courses and Events – Sprint 2*

- User Story 4 - As a site visitor, I can see a sorted list by date of all upcoming “Certification Courses” so that I can choose the best course for me
- User Story 5 - As a site visitor, I can see a list of all upcoming “Other Courses” (non-certification courses) so that I can choose the best course for me
- User Story 6 - As a site visitor, I can see a list of all upcoming “Social/Networking Events.” so that I can select ones I am able to attend

[www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog/example](http://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog/example)

# Scrum Artefacts – Product Backlog

## Example - Professional Body Website

### Product Backlog

User Story 1

User Story 2

User Story 3

User Story 4

User Story 5

User Story 6

### **News Section** - Total of 6 Story Points to complete Sprint 1

- User Story 1 - As a site visitor, I can read current news on the home page so that I stay current on key professional items – 1 Story Points
- User Story 2 - As a site visitor, I can email news items to the editor so that they can be considered for publication – 2 Story Points
- User Story 3 - As a site member, I can subscribe to an RSS feed of news so that I can stay current on the news that is of interest to me – 3 Story Points

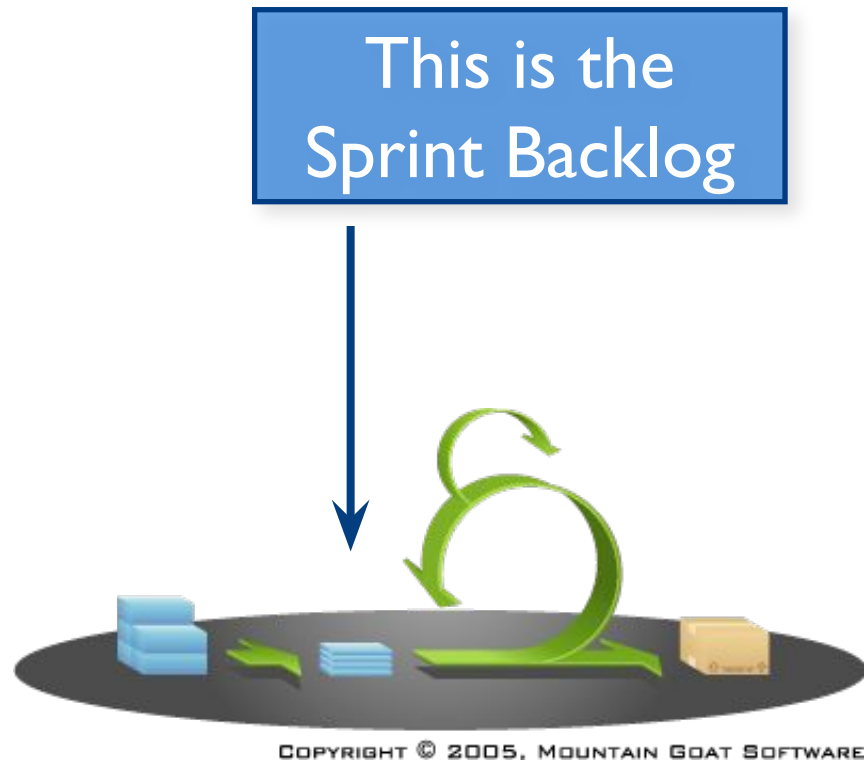
### **Courses and Events** - Total of 8 Story Points to complete Sprint 2

- User Story 4 - As a site visitor, I can see a sorted list by date of all upcoming “Certification Courses” so that I can choose the best course for me – 2 Story Points
- User Story 5 - As a site visitor, I can see a list of all upcoming “Other Courses” (non-certification courses) so that I can choose the best course for me – 5 Story Points
- User Story 6 - As a site visitor, I can see a list of all upcoming “Social/Networking Events.” so that I can select ones I am able to attend – 1 Story Points

**To complete this total Product Backlog would take 14 Story Points**

[www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog/example](http://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog/example)

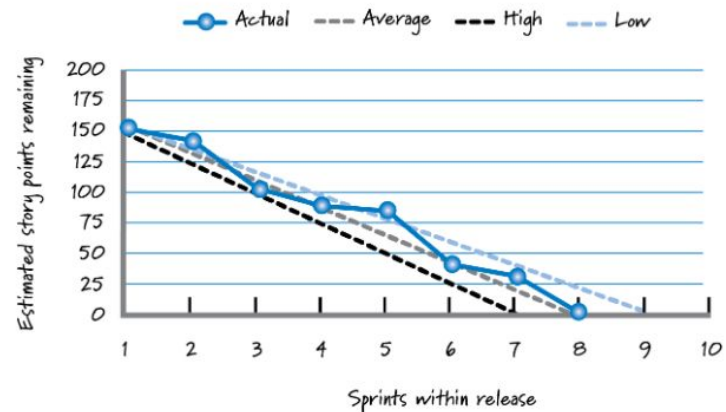
## Scrum Artefacts – Sprint Backlog / User Story



- Scrum team decompose User Stories to Low level User Stories during Sprint Planning
- The User Stories are used for a conversation between the SME and developer. Developer updates the User Stories with the tasks and hours estimates, "Just-In-Time"
- Remaining estimated items are updated daily
- Sprint Backlog is seldom altered
- User stories in the sprint are either completed 100% or not done

## Scrum Artefacts – Burn Down Chart

- A burn down chart is a graphical representation of work left to do versus time.
- The outstanding work (or backlog of user stories) is often on the vertical axis, with time along the horizontal.
- It is used to predict when all of the work will be completed.



# Scrum Summary

- Another look at **SCRUM**

<https://www.youtube.com/watch?v=9TycLR0TqFA>

# Intended Learning Objectives

## Software Development Lifecycles

### Agile:

- ~~1. Understand what Agile is and its origins.~~
- ~~2. Understand the Agile framework.~~
- ~~3. Understand Scrum — Roles, Ceremonies and Artefacts.~~
4. Understand advantages / disadvantages of Agile.

# Disadvantages of Agile

Powered by  **Poll Everywhere**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at [pollev.com/app](https://pollev.com/app)



# Agile – Advantages & Disadvantages

## Advantages

- Customer satisfaction by rapid, continuous delivery of usable software
- People and interactions are emphasised rather than process and tools
- Continuous attention to technical excellence, good design and quality
- Regular adaptation to changing circumstances

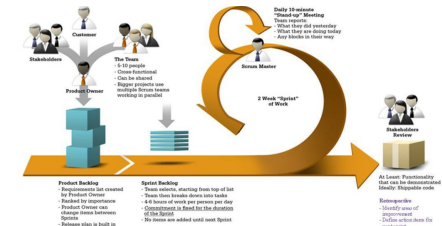
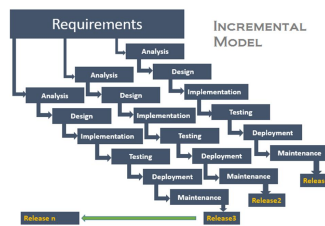
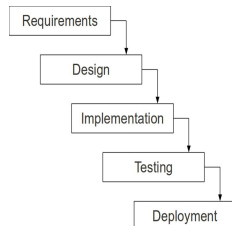
## Disadvantages

- Difficult to assess the effort required at the beginning
- Can be very demanding (from traditional approaches) on users time
- Harder for new starters to integrate into the team
- Agile is a very different approach – It can be intense for the team
- Requires experienced resources (which are limited in today's market)

# Formal or Agile which one Should I use???

There is no one right answer. The following questions can assist deciding:

- How stable are the requirements?
- Do the end users need to collaborate?
- Is the timeline aggressive or conservative?
- What is the size of the project?
- Where are the project teams located?
- What are the critical resources?



# Agile Where to find out more information

- [www.agilealliance.org](http://www.agilealliance.org)
- [www.mountangoatsoftware.com/scrum](http://www.mountangoatsoftware.com/scrum)
- [www.scrumalliance.org](http://www.scrumalliance.org)
- [www.controlchaos.com](http://www.controlchaos.com)