MONASH University

FIT 3161 – FIT 3163
CS – DS, Software Project 1

Software Development Project Management
Week 4: Agile approach to software development (compared to a Predictive approach)

Semester 2, 2023

Faculty of Information Technology

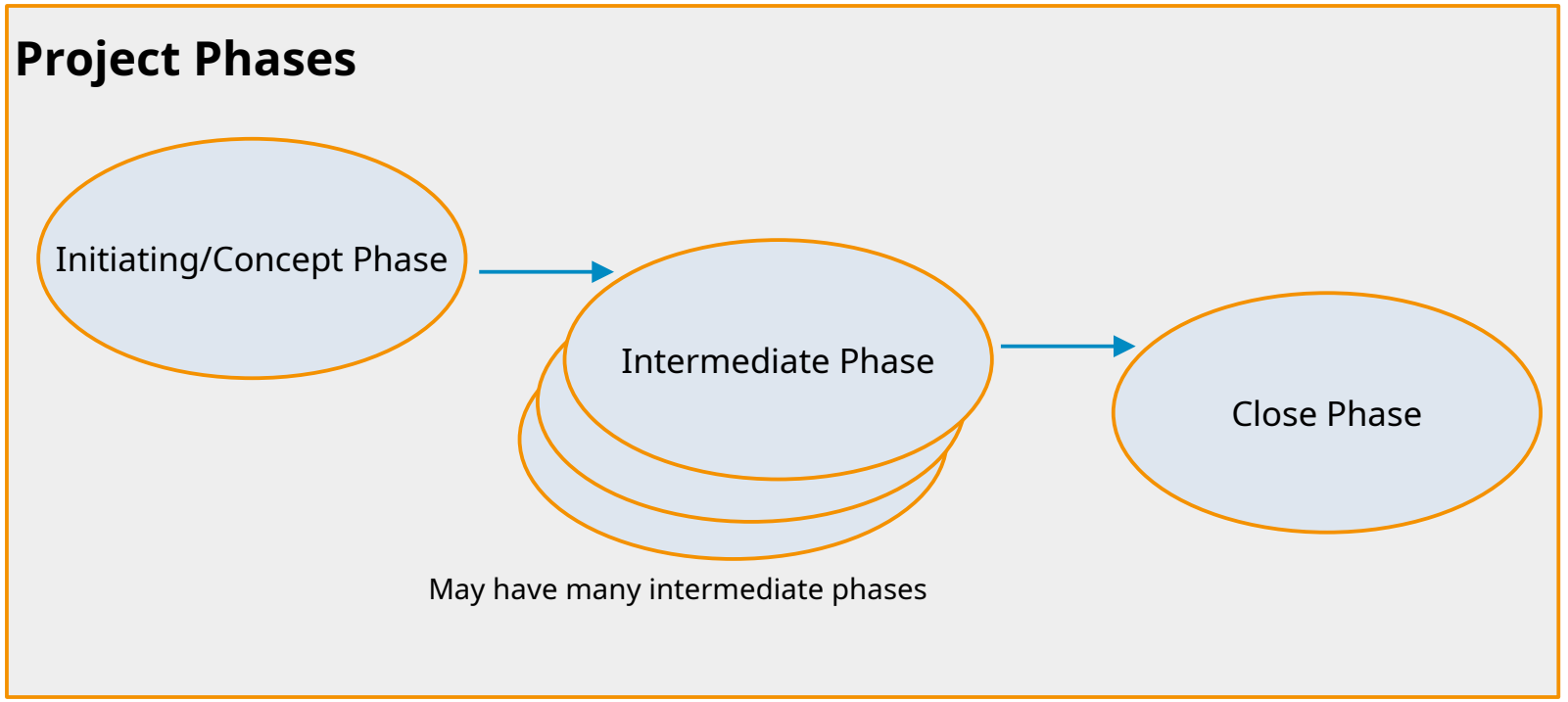
Lets review last week material... (week 3)

Factors affecting project execution:

- Scope
- WBS
- Schedule
- Milestone
- Gantt Chart
- Activity Sequence
- Triple Constraints

Lets check a few things first!

Projects progress in Phases ...



Lets check a few more things …

Project Phases have Process Groups …

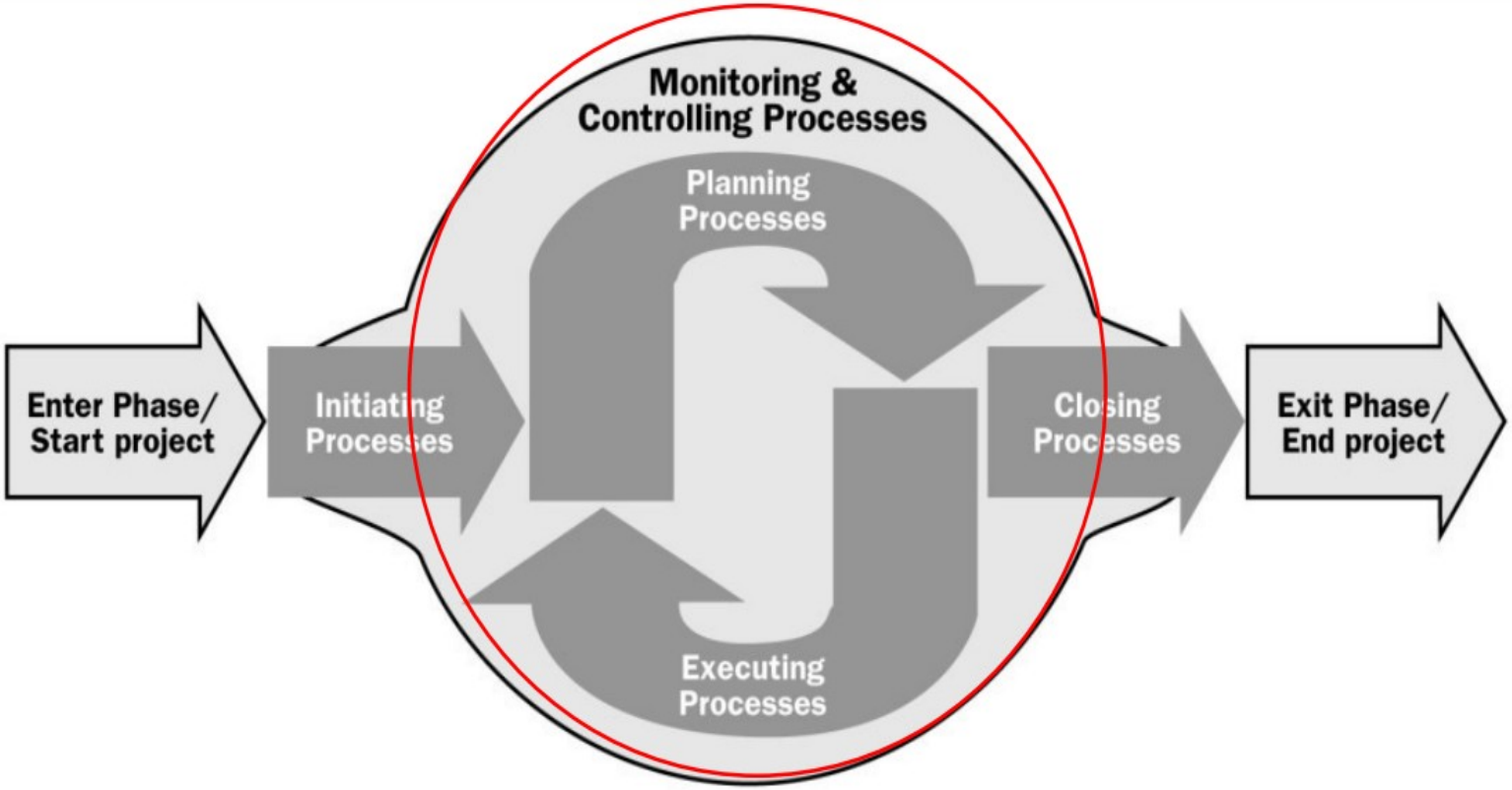


Image source: PMBOK Guide 5th Ed.

Lets check a few things first!

Phases are common to all project life cycles

Does not depend on which life cycle is followed

Differences:

- **For each life cycle the phases are arranged managed differently**
- **Start/Concept and Closing phases always exist**
- **Intermediate phases may differ: eg Predictive v/s Agile**

User Requirements

In following slides we will mention ***"User Requirements".***

For now the following explanation will be sufficient.

"User Requirements are the capabilities and characteristics of the software that the client wants, and that will be provided by the software developer."

(We will look at User Requirements in more details later in this unit.)

Software Projects have "Life Cycles"...

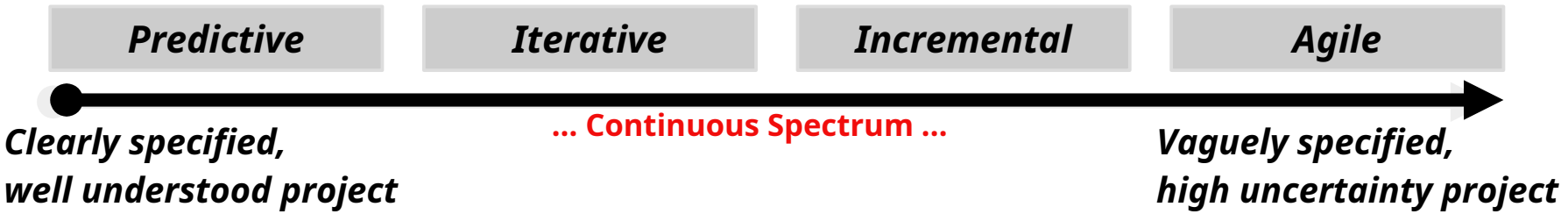
Life Cycles describe how activities during software development are related to each other and how changes in user requirements are handled.

Life Cycle Type	Activities	End Product Delivery	Aim	User Requirements
Predictive	*Each activity is performed once only*	*At end of project execution*	*Keep cost down, avoid project scope creep*	*User reqs are fixed early in project, and do not change*
Iterative	*Activities may be repeated to improve product*	*At end of project execution*	*Aim for correctness and quality*	*User reqs may change*

Continued on next slide ...

Software Projects have "Life Cycles" (contd)...

Life Cycle Type	Activities	Delivery	Aim	User Requirements
Incremental	*Each activity is performed once for each increment*	*Deliver in small increments several times during project execution*	*Gradual delivery to client in increments*	*User reqs can change*
Agile	*Repeated to improve product*	*Frequent small delivery*	*Aim for client satisfaction and incorporate client feedback*	*User reqs are likely to change and refined*



Software Development Methodologies

"A Software Development Methodology is a set of procedures and rules used to guide the software development process."

Methodology and Model are 2 words often used interchangeably in related literature specially on the web.
Eg: Waterfall Methodology and Waterfall Model.

Each Methodology fits somewhere on the life cycle spectrum. Eg:

Waterfall : Predictive end of spectrum

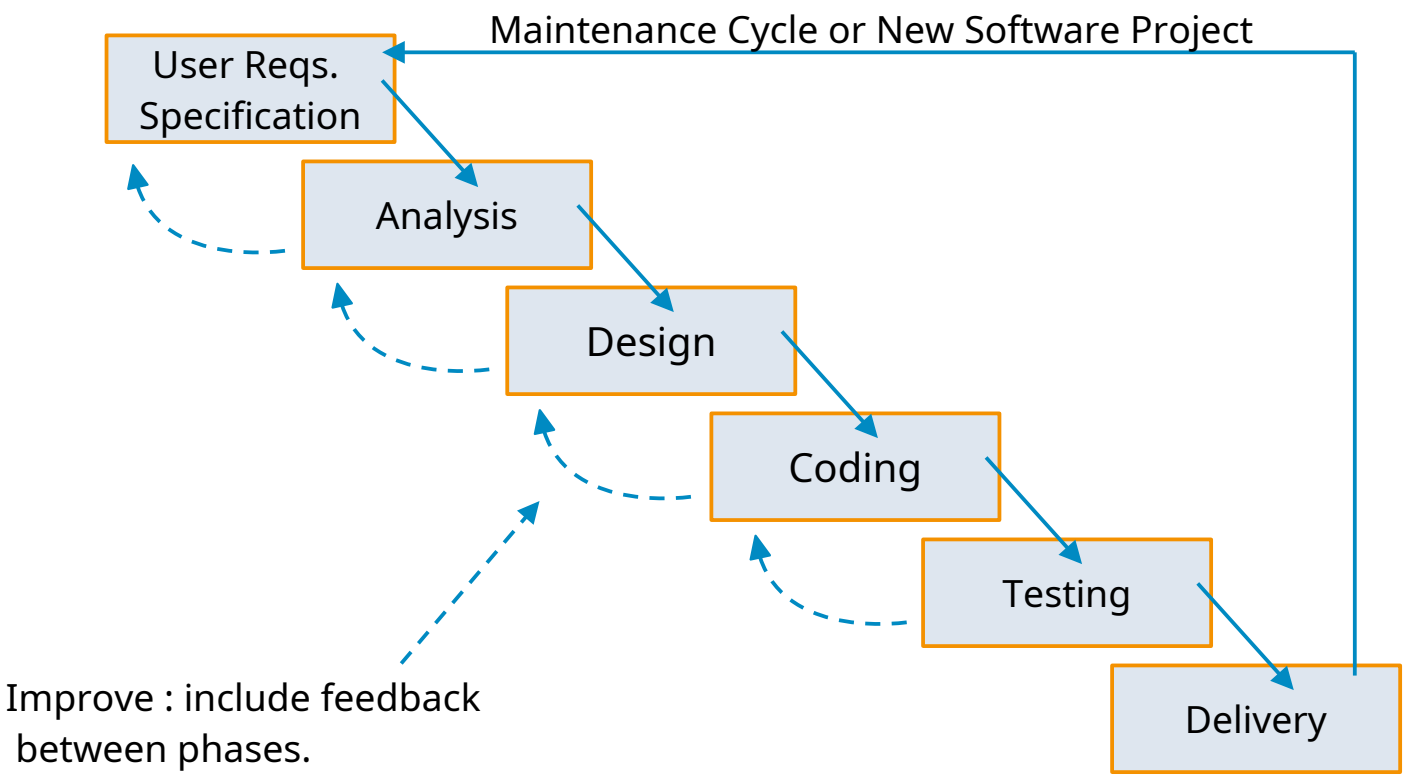
Spiral : Middle of spectrum

Kanban, Lean, Scrum : Agile end of spectrum

(Note: we will revisit Software Dev. Methodologies in more detail in S2)

Software Development Methodologies

Waterfall Methodology: Phases are executed strictly sequentially

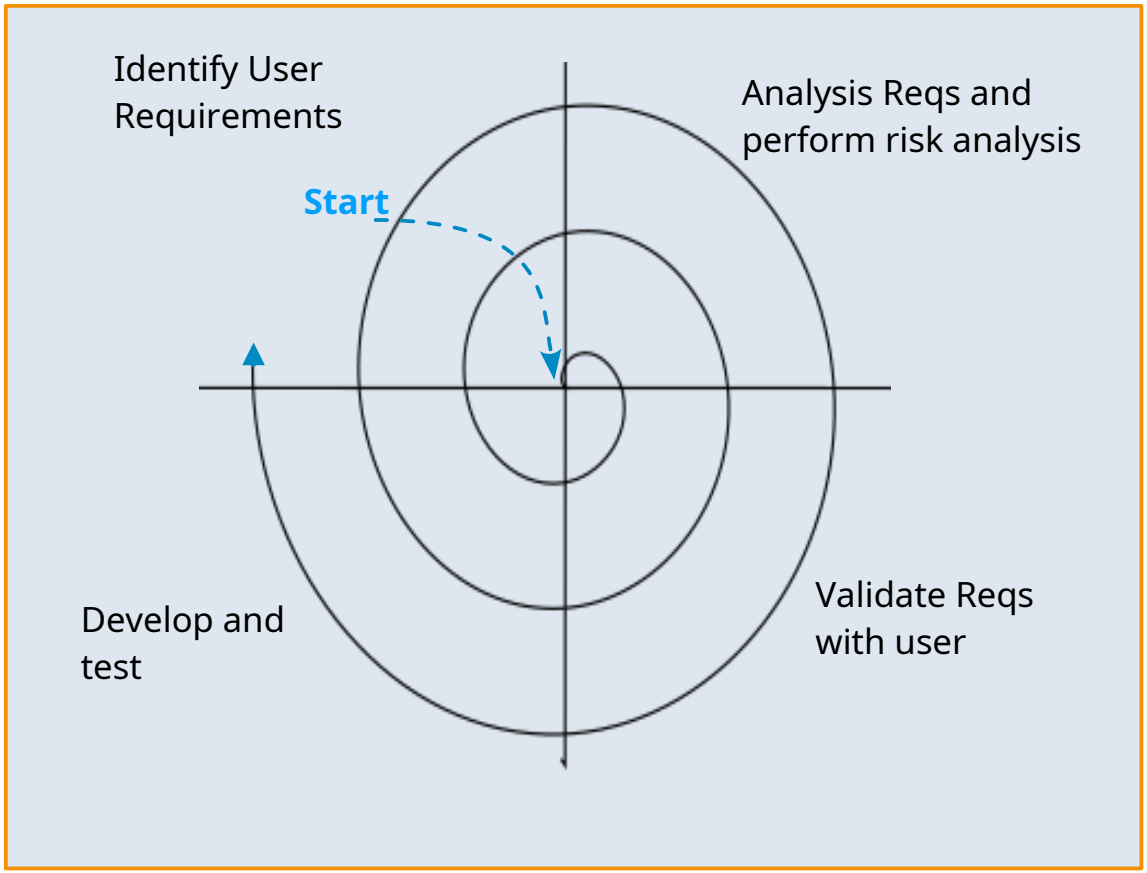


Software Development Methodologies

Spiral model

Start
→ Identify user Requirements
→ Analyse
→ Validate
→ Develop and Test
→ Repeat spiral ...
Incremental delivery

The Spiral model is complex and generally not suitable for small projects



Software Development Methodologies

Agile methodologie<u>S</u> : There are many types of Agile.

Note: Agile is NOT a Methodology. It is a set of values that all agile methodologies adhere to. These values are expressed in the **Agile Manifesto.**
(next slide)

Examples of Agile:

- eXtreme Programming (XP)
- Kanban
- Scrum [1]

[1] As Scrum is currently the most popular Agile methodology, people often say Agile when they actually mean Scrum. This is strictly NOT correct.

Software Development Methodologies

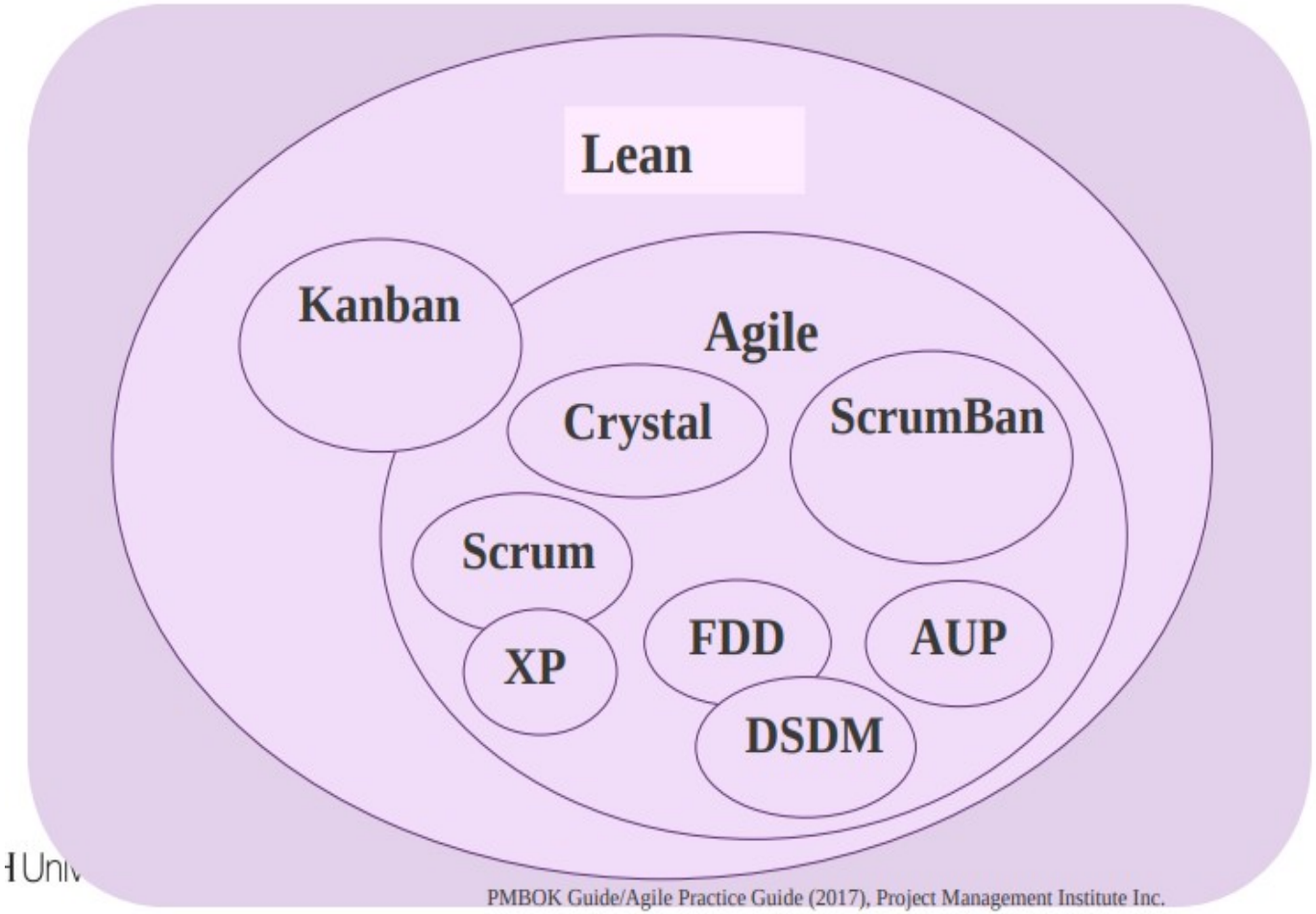
Agile Manifesto (agilemanifesto.org)

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.

Agile Methodologies



PMBOK Guide/Agile Practice Guide (2017), Project Management Institute Inc.

Scrum Methodology

- Scrum Team: Members have specific roles
- Scrum Ceremonies
- Scrum backlogs[1] and rundown chart
- Time boxed Sprints (ie: fixed time Sprints)

[1]: Scrum uses a Kanban Board to record backlogs, but this is not the same as Kanban Methodology which also uses a Kanban Board

Scrum Methodology: Scrum Ceremonies

Scrum requires regular “Ceremonies”

- **Sprint Planning**
  - Start Sprint and plan work for the coming Sprint
- **Daily StandUp Meeting/Daily Scrum** (This is not expected in a student project.)
  - Daily Progress review and discuss challenges (students can decide on alternative)
- **Sprint Review**
  - Demo to client and identify complete and incomplete user stories
- **Sprint Restrospective**
  - Discuss issues and difficulties and also successes encountered during completed sprint.

    Aim is to improve performance of Project Team

Scrum Methodology: Scrum User Stories

User Stories allows the developer to capture User Requirements in simple English format, while keeping the focus on the User.

Usually expressed as:

> As <user> I want <function/feature> so that I can <do something>

Example:

> *As a Medical Doctor,*
>
> *I want to be able to record the patient's blood pressure regularly,*
>
> *So that I can monitor the patient's progress and review their medication*

Scrum Methodology: Scrum User Stories → Epic

An Epic is a sequence of related User Stories

Example Epic:

As a Medical Doctor,

I want to be able to efficiently manage a patient health condition

Related User Stories:

1. As a Medical Doctor,

I want to be able to record the patient's blood pressure regularly,

So that I can monitor the patient's progress and review their medication

2. As a Medical Doctor,

I want to be able to review the patient's past medication

So that I can monitor manage and review their medication

Scrum Methodology: Scrum User Stories → Epic

An Epic is a sequence of related User Stories

In Agile:

Capture Epics from user

Breakdown into smaller User Stories

Similar process using Agile specific terminology

Break down task : EPIC into smaller sub task tasks : User Stories

Shown as Work Break Structure (WBS)

(See Week 3)

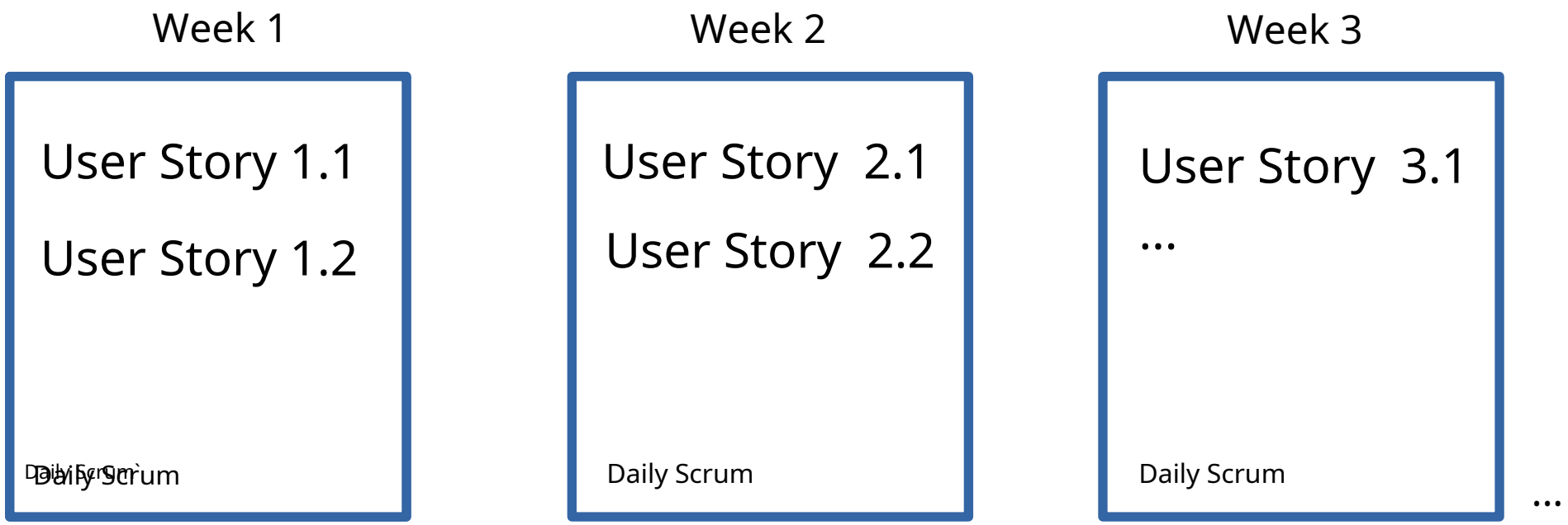
Scrum Methodology: Scrum Sprints

Each Sprint: Work is done on a set of User Stories (could be an Epic)

Sprints are “Time Boxed”, typically 1 - 4 weeks (may be longer)

Student Projects Sprints 1 or 2 weeks duration (1 week sprint example shown)

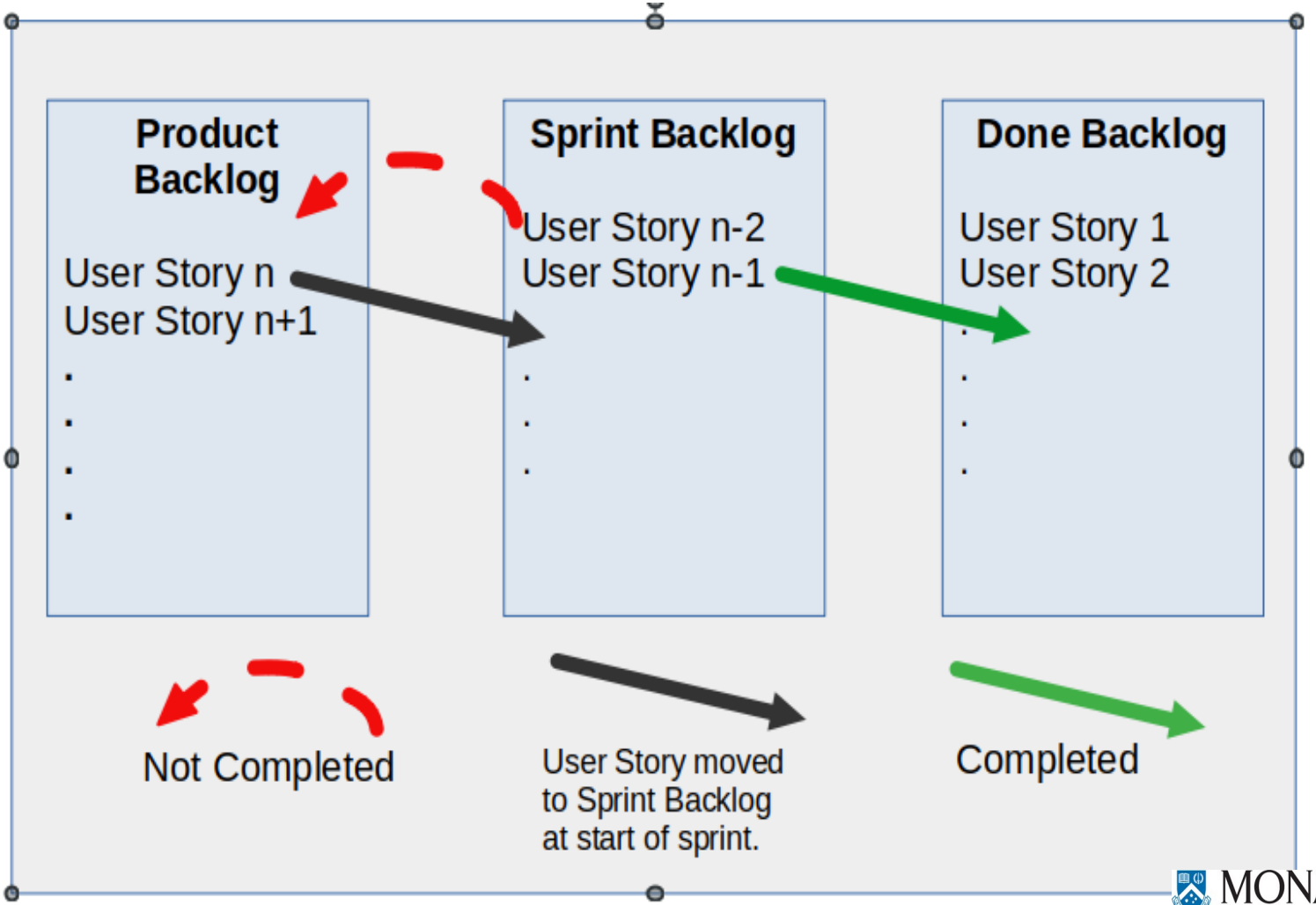
`



Scrum Methodology: Backlogs

Project Start: **Sprint** and **Done** are both empty

Any time: Add items to **Product** at Start, but also any time during Project



Project End: **Sprint** and **Product** are both empty

Scrum Methodology: How Backlogs works?

- Before Sprint and anytime during project
  - Add User Stories to Product Backlog
- Begin Sprint (refer to diagram in previous slide)
  - Sprint Planning
    - Select User Stories and **move to Sprint Backlog**
      - Work on User Stories for Sprint duration (typically 1 to 2 weeks)
  - End of Sprint
    - **Move completed User Stories to Done Backlog**
    - **Move incomplete User Stories back to Product Backlog**
    - Do Sprint Review and Sprint Retrospective
- Go to Begin Sprint for next Sprint.

- Incomplete User Stories may be moved back into next sprint OR delayed to a later sprint

Scrum Methodology: End Product Delivery

- Items in Done are usually “shippable”,ie: of quality that it can be tested/used by the end user.
  - → Encourages design to be modular and user focused

- This allows feedback from users and may lead to changes to initial User Requirements
  - → Allows new user reqs to be incorporated more easily
  - → Leads to better quality and client satisfaction

- Also allows for gradual delivery in increments, hence providing value to the client early

Scrum Methodology: Other Project Tasks.

While Scrum primarily deals with development to satisfy End User Stories, it can be adapted for other project activities.

Example: Sprint to write documentation. In this case, the “User” in the “User Story” is the Developer/yourself. Eg:

> As “the Developer”,
>
> I need to write documentation for the backend server
>
> So that documentation is available for software maintenance later.

Software Methodology: Recap

- Methodologies determines how software development activities are sequenced and organised
- Scrum is a type of Agile, among others, and is most popular (currently, 202x)
- Predictive and Agile Methodologies are at opposite ends of Life Cycle spectrum
- While Agile is more commonly used today, there may be cases where a Predictive or other methodology may be more appropriate.
- Often the actual methodology applied is an adapted version of a standard methodology or a blend of several, to suit specific circumstances: project characteristics, developer capabilities and experience, client expectations etc.

Software Methodology in Student Team Project

- Team must decide on Methodology to use
- Team can adapt Methodology to suit their specific circumstances
- Team must justify all adaptation and changes
- Documentation is key
- **Team must be able to demonstrate that they are applying a methodology through activities undertaken and organisation of activities**

Week 4 : Q and A?

Software Methodology in Student Team Project

Activity

- Write up 4 – 5 User Stories as a “Developer”
  - Relates to you as an individual developer or as a team
  - Relates to the project you have been allocated to
    - Eg:
      - As a developer, I need to setup my computer with the required applications so that I can contribute to the project effectively
      - As a developer team, we need to …