



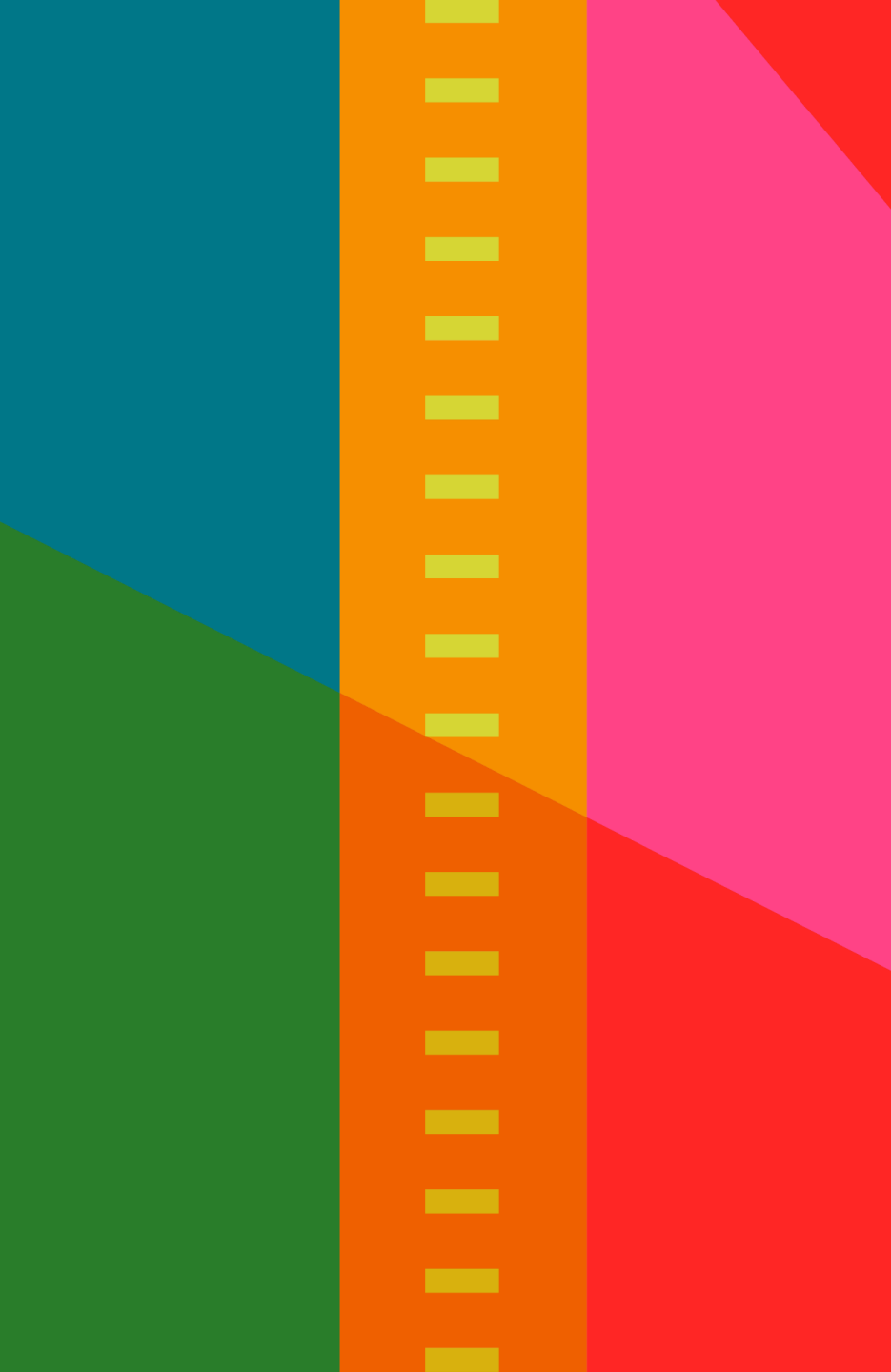
CO1605

Systems Analysis and Database Design

Julie Allen

JAllen17@uclan.ac.uk

C&T Building Room CM214



Learning Outcomes

Students will be able to:

- Identify various life cycles and methodologies
- Compare and evaluate the above
- Decide which of the above to choose for a given scenario

Lifecycles and Methodologies



Definition

Life Cycles

The phases of a development project

How a project is planned, controlled, and monitored

From inception to completion and use of product

Methodologies

Co-ordinating the activities of a team

An organised, documented set of procedures and guidelines for one or more phases of the software life cycle

Used Interchangeably



The Development of Life Cycles



Early attempts, in the 1960s and 1970s, to develop large and complex systems were discouraging

The focus was on programming

Very few software companies followed any formal methodology

The Software Development Crisis

New systems were often delivered years too late:

- Over budget
- Unreliable
- Difficult to maintain
- Not meeting requirements



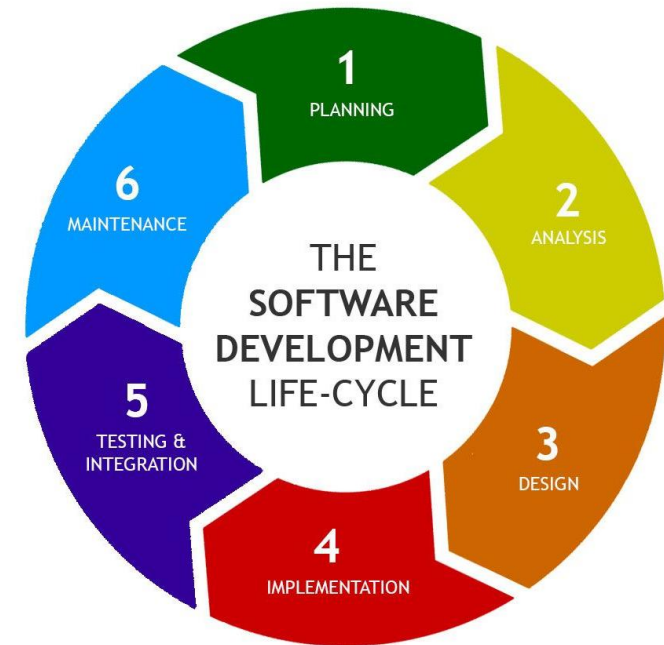
Software Development Life Cycle: SDLC

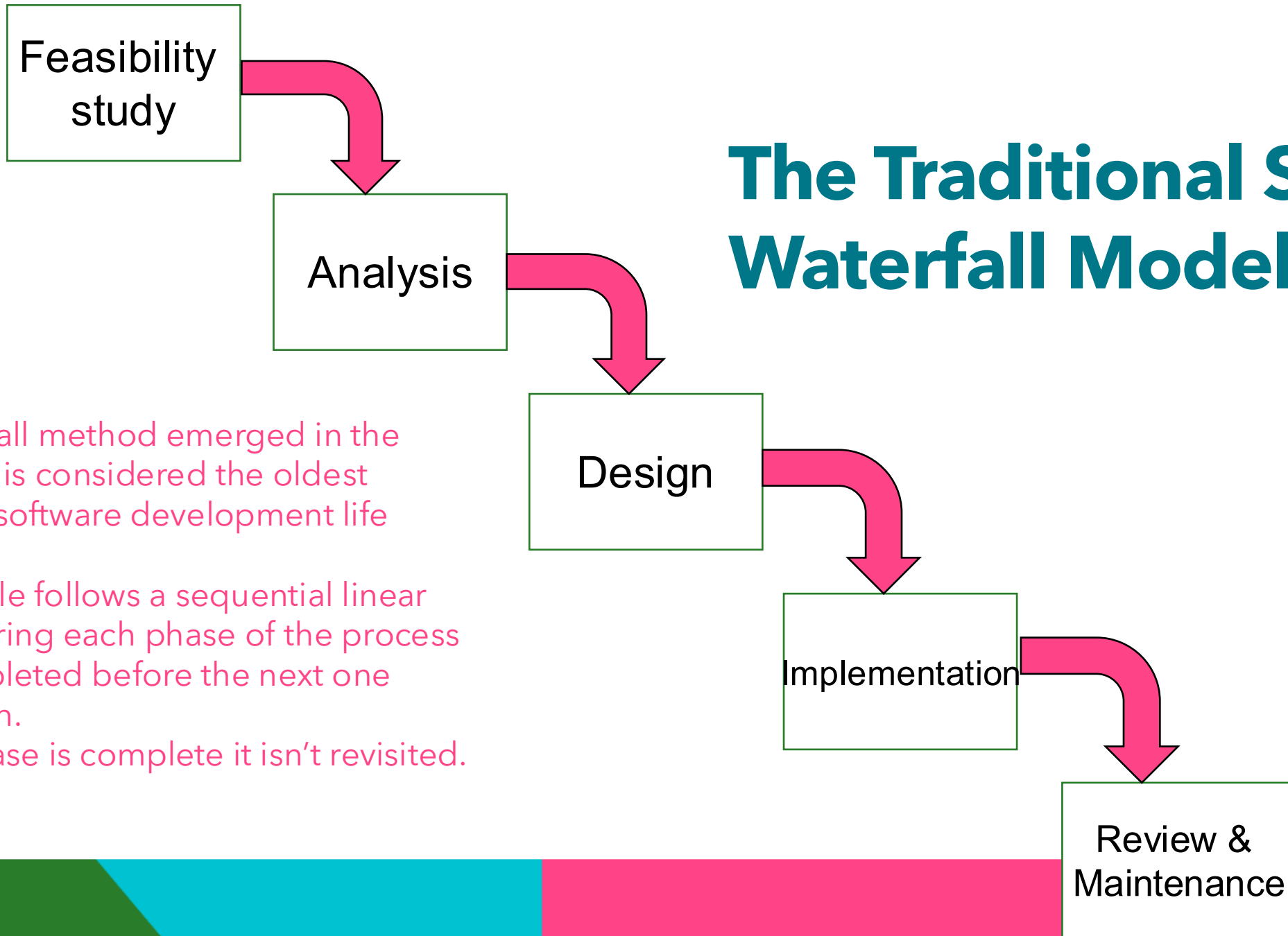
The systems development life cycle (SDLC) was an attempt to establish a systematic approach to Software Development

A process that outlines the stages that software goes through from conception to delivery

'Lifecycle' reinforces the idea that the project follows a process with a beginning, a middle and an end

The SDLC concept was adapted to different flavours.





The Waterfall method emerged in the 1970's and is considered the oldest structured software development life cycle.

The life cycle follows a sequential linear path, requiring each phase of the process to be completed before the next one could begin.

Once a phase is complete it isn't revisited.

SDLC (Waterfall)

Advantages

- The linearity makes it easy to understand and manage
- Clearly defined phases
- Clear progress is made as each phase is completed before the next phase starts
- Forces thorough analysis – all requirements are clearly defined before development starts

Disadvantages

- Inflexible: does not allow for changes to be made once a phase is complete
- No working model produced until late in the life cycle
- Client cannot visualise the product early on, and may only suggest improvements/additions during the testing phase
- Can lead to projects being over-budget, late, and not fit for business purpose
- Documentation can be labour intensive

The Agile Manifesto



On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen people met to talk, ski, relax, and try to find common ground—and of course, to eat. What emerged was the Agile 'Software Development' Manifesto.

Agile Philosophy



An iterative, incremental approach with each iteration building on the last

Focuses on collaboration, and continuous improvement.

Breaks down projects into smaller and more manageable chunks

The Agile Manifesto: 4 Values

1

**Individuals and
Interactions**

Over
Processes and
Tools

2

Working Software

Over
Comprehensive
Documentation

3

**Customer
Collaboration**

Over
Contract
Negotiation

4

**Responding to
Change**

Over
Following a
Plan

Agile

Advantages

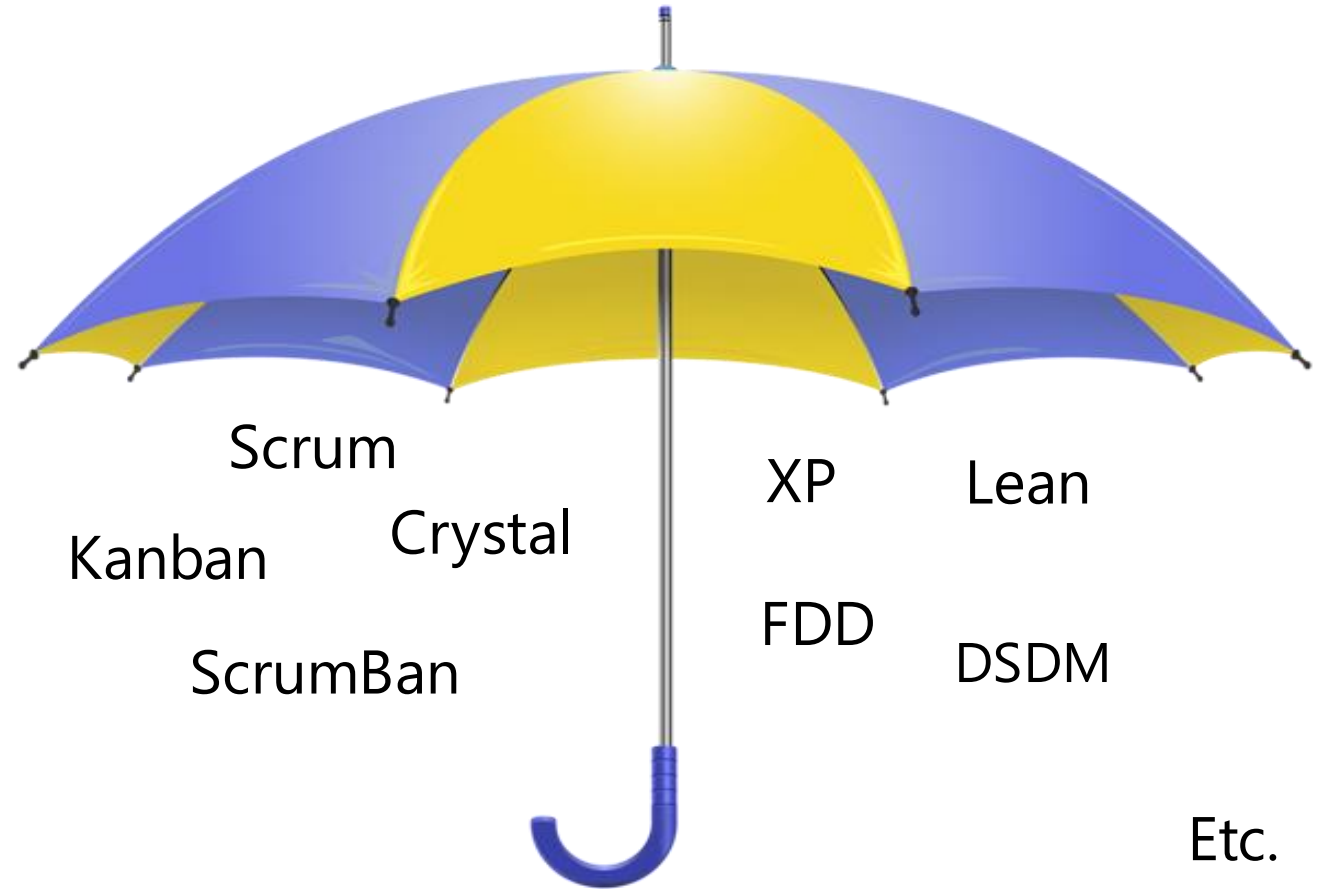
- Quality software due to the iterative development and continuous testing and improvement
- Clarity between team members due to transparent development process
- Changes in the requirements are easily addressed

Disadvantages

- Team can become overwhelmed with a large number of changing requirements
- Lack of documentation can become a problem later in the development
- Discussions and workshops can be time-consuming for the team



The Agile Umbrella



Scrum

What is it?

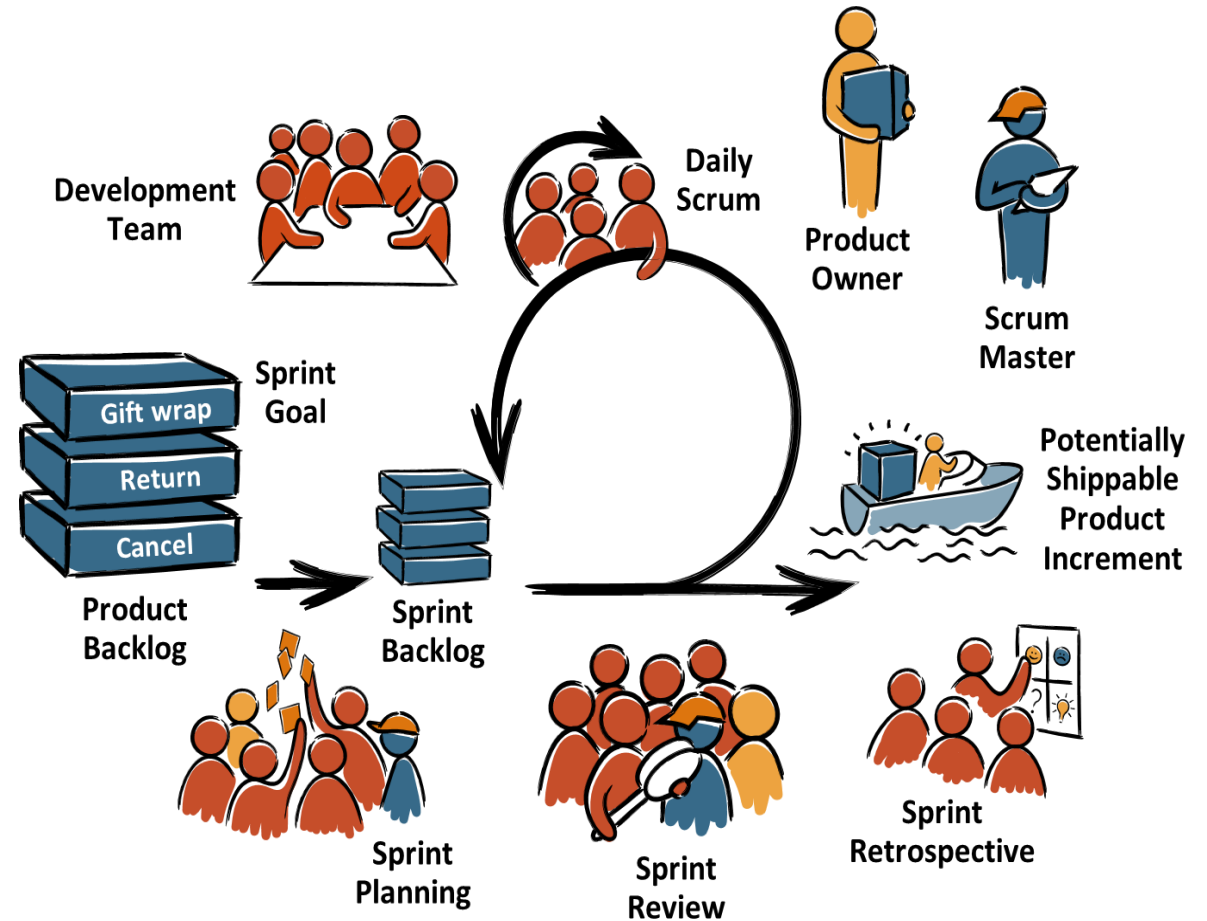
A lightweight empirical framework that helps people, teams and organisations generate value through adaptive solutions for complex problems

What are the advantages of using it?

Conceptually simple

What are the disadvantages of using it?

Difficult to master



Scrum Roles



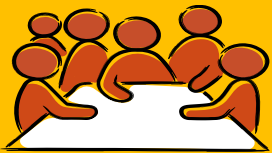
Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team.



Scrum Master

A servant-leader that does whatever it takes to make the Scrum Team successful, such as removing organizational impediments, facilitating Scrum events, protecting the team.



Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable increment of “Done” product at the end of each Sprint.

Kanban

What is it?

A visual method used to track tasks and reduce inefficiencies in a project

What are the advantages of using it?

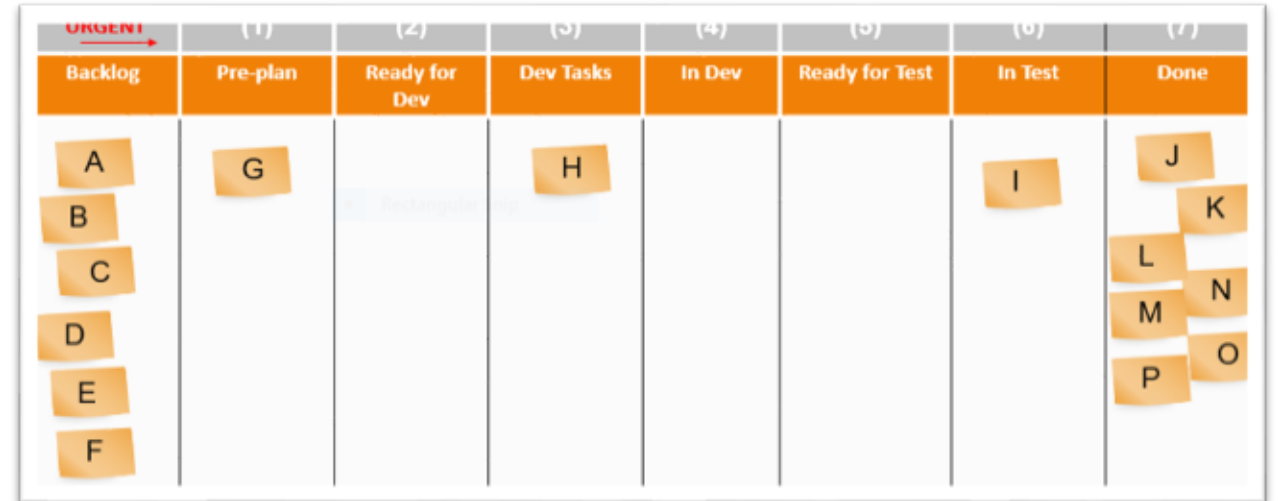
Provides a simple visualisation of the workflow

Bottlenecks are easy to spot

What are the disadvantages of using it?

Overly complex boards

Keeping the board up-to-date

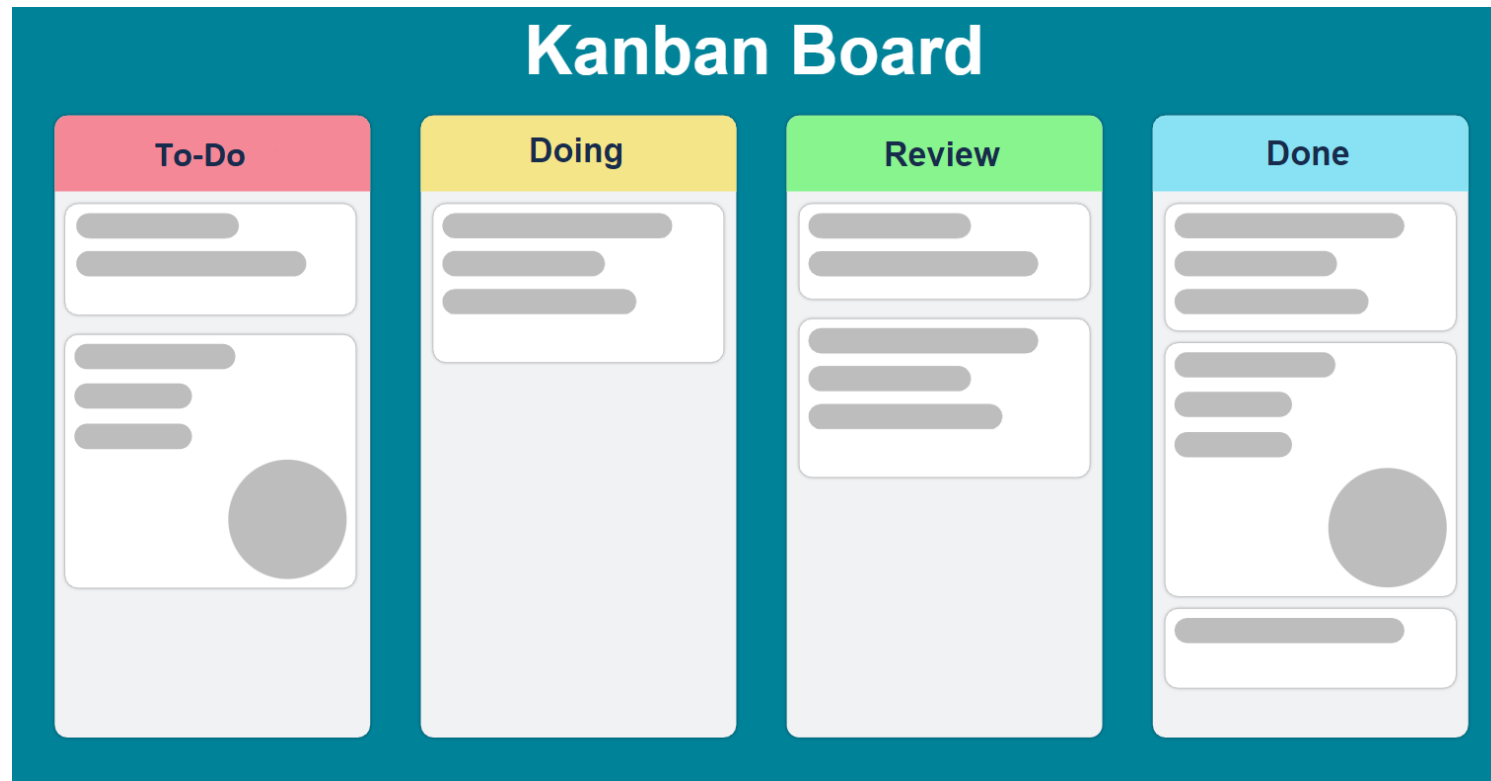


The heart of the Kanban method is the Kanban board

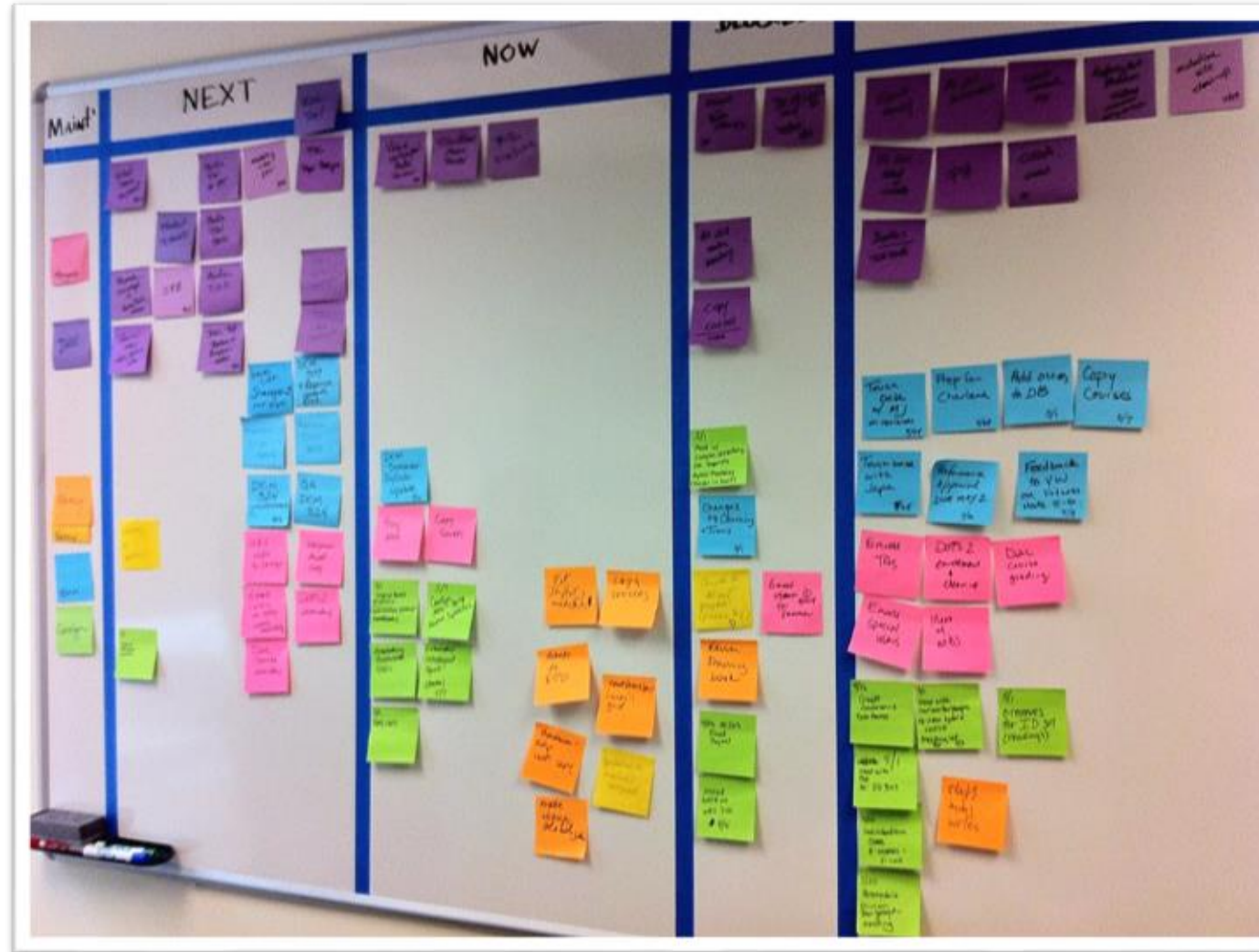
Kanban Board

Kanban a Japanese word meaning **visual card**

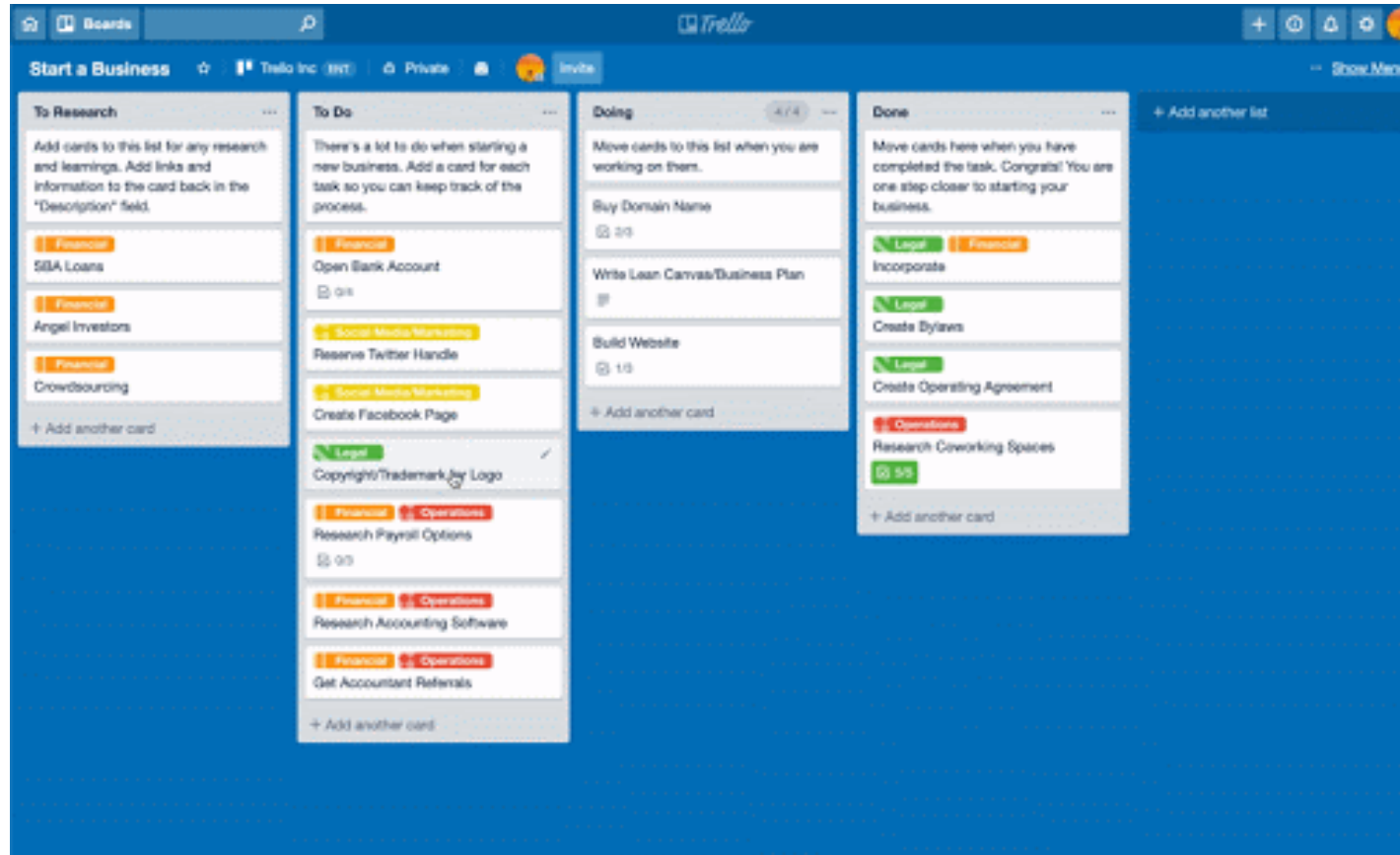
A project management tool designed to help visualize work



Physical Board



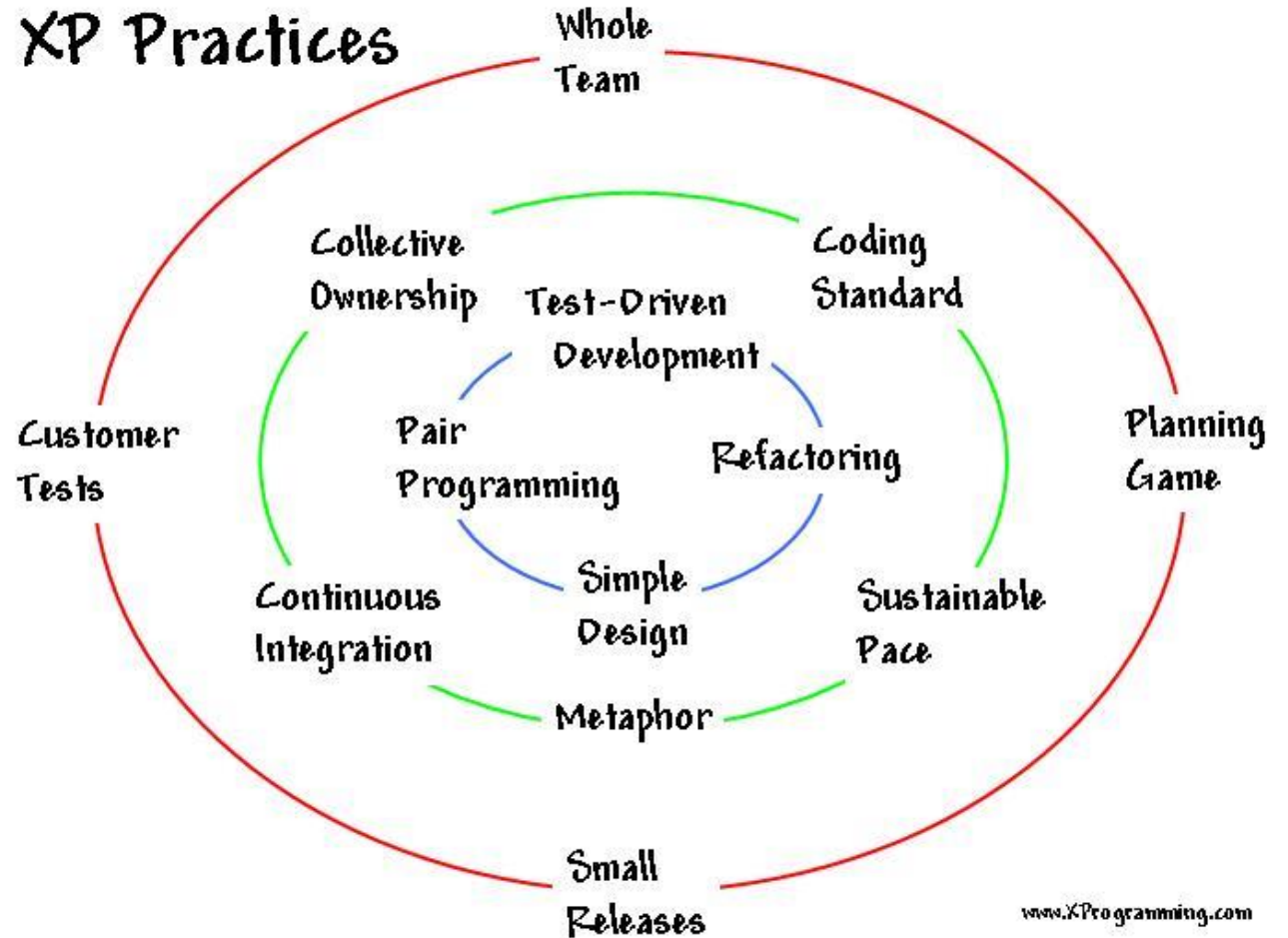
Digital Board



Extreme Programming - XP

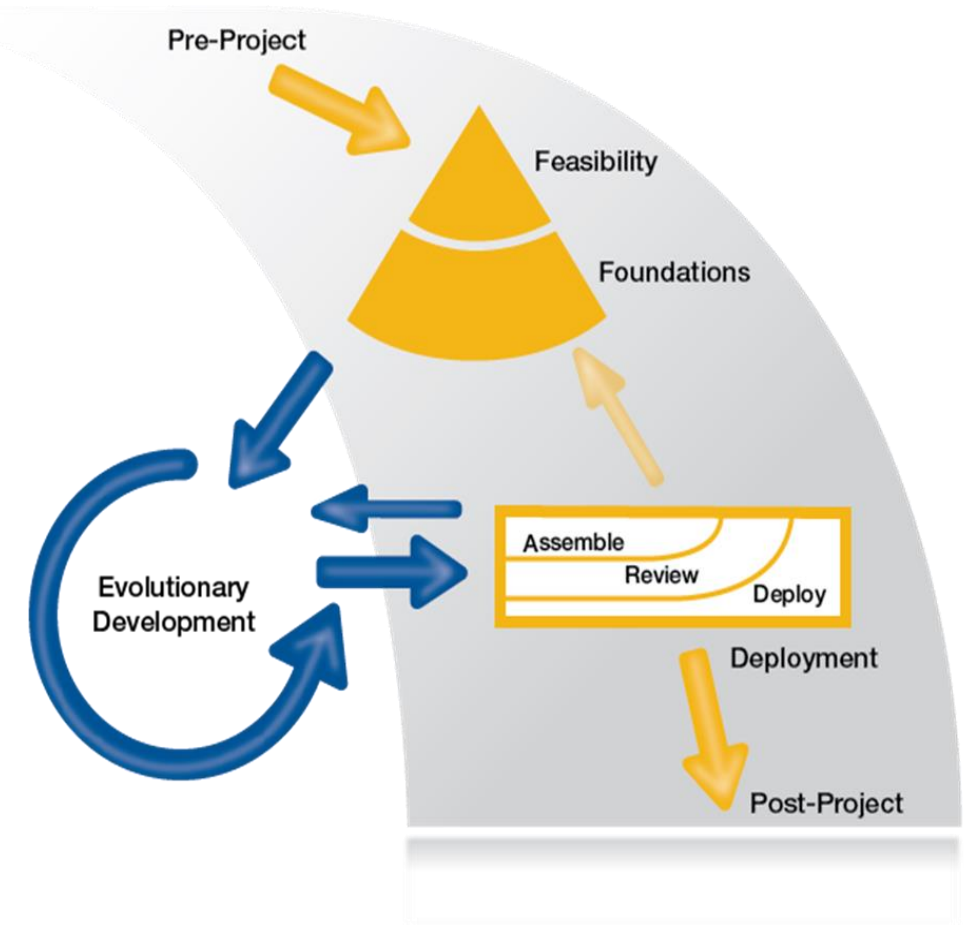
Values :

- Communication: Everyone on a team works jointly at every stage of the project.
- Simplicity: developers strive to write simple code
- Feedback: deliver software frequently, get feedback, and improve
- Courage: developers evaluate their own results and respond to changes as needed
- Respect: everyone contributing to a common goal



DSDM: Dynamic System Development Method

- Created in 1994 to address the problems of the traditional Waterfall approach, as well as the problems introduced by RAD (Rapid Application Development).
- DSDM achieved this by recognising that both approaches had strengths and areas for improvement,
- DSDM brought together the best parts from a traditional approach: control and quality, and from RAD: good communication, business involvement, and transparency

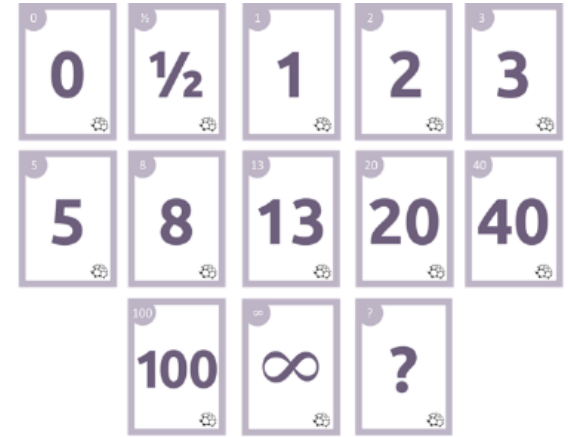
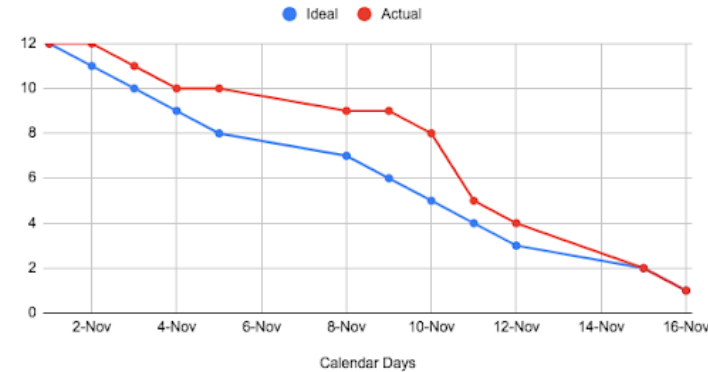


Agile Business Consortium

DSDM Practices

1. Facilitated Workshops
2. User Stories
3. Prioritisation
4. Timeboxing and Sprints
5. Relative Sizing
6. Planning Poker
7. Burndown Charts

Burndown Chart

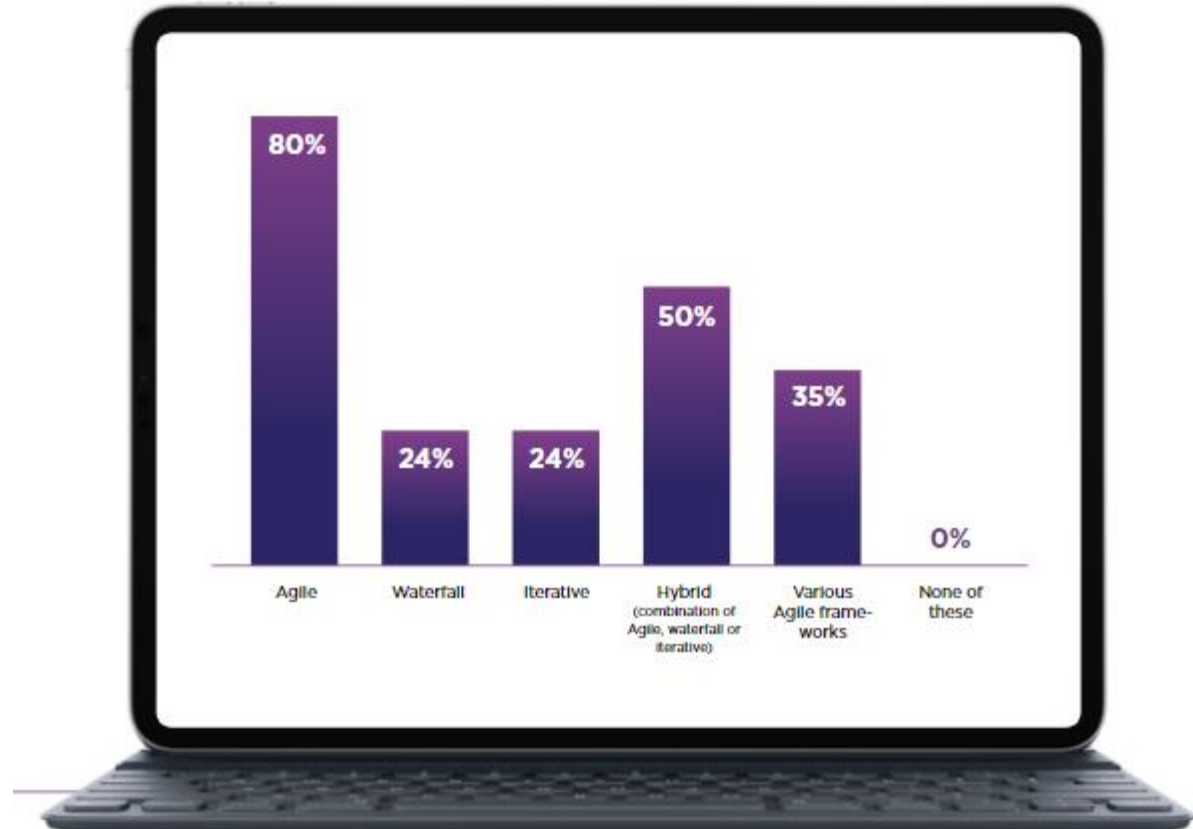


*As a: Customer
I need: To cancel a
hotel booking
So that: I can get a
refund*



State of Agile Report

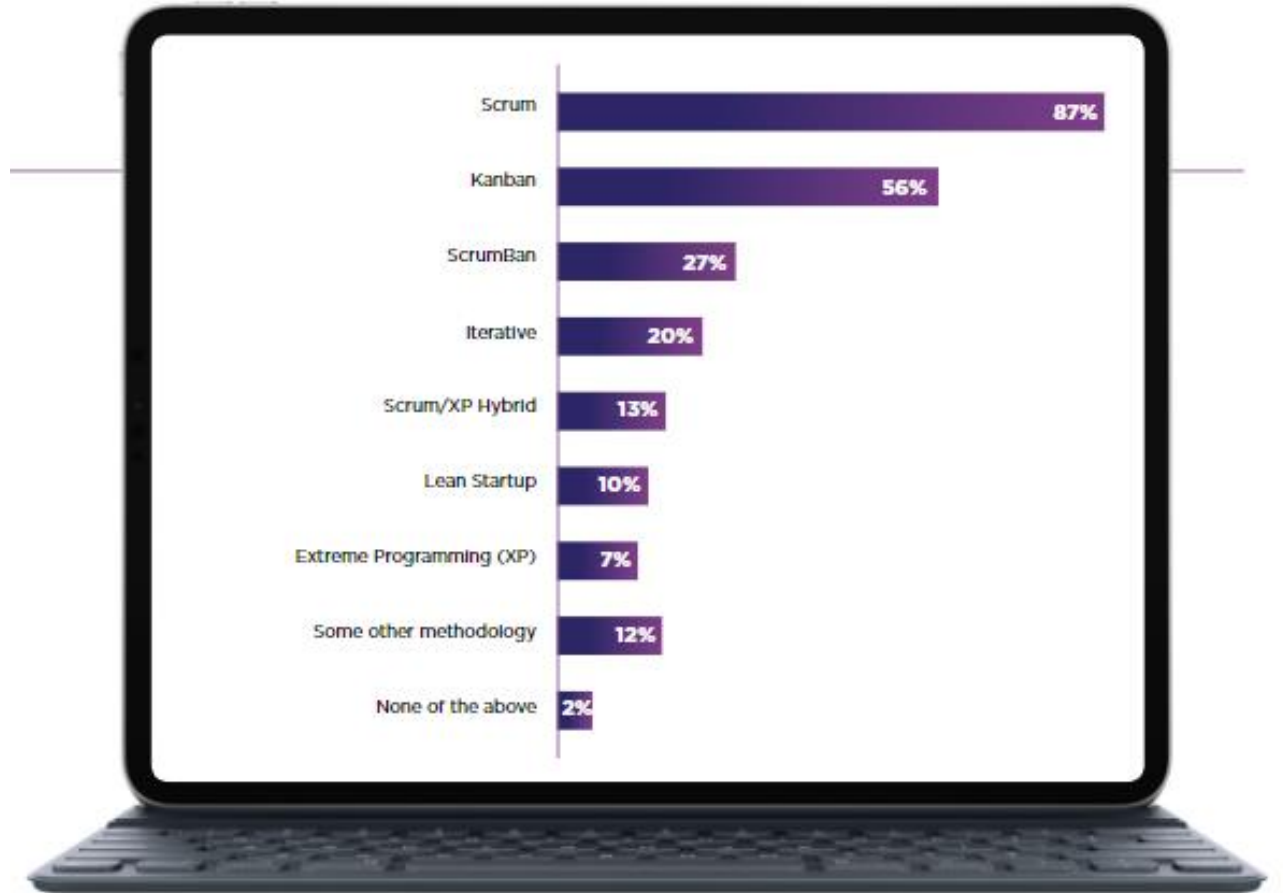
- 4 out of 5 software development teams are using Agile as their predominant approach
- Half are using a combination of Agile, Waterfall, and/or Iterative



State of Agile 2022

State of Agile Report

- More than 4 out of 5 software development teams are using Scrum
- More than half are using kanban
- Over a quarter are using a hybrid of Scrum and Kanban called ScrumBan



State of Agile 2022

Systems Analysis - we have covered:

- UML: Use Case and Activity Diagrams
- Use Case textual Descriptions
- Requirements: Functional and Non-Functional
- User stories
- Requirement Elicitation: Questionnaires/Surveys, Observation, Shadowing, Document Analysis, Workshops, etc.
- Life Cycles: Waterfall, Agile

Semester 2:

Database Design

Any Questions?