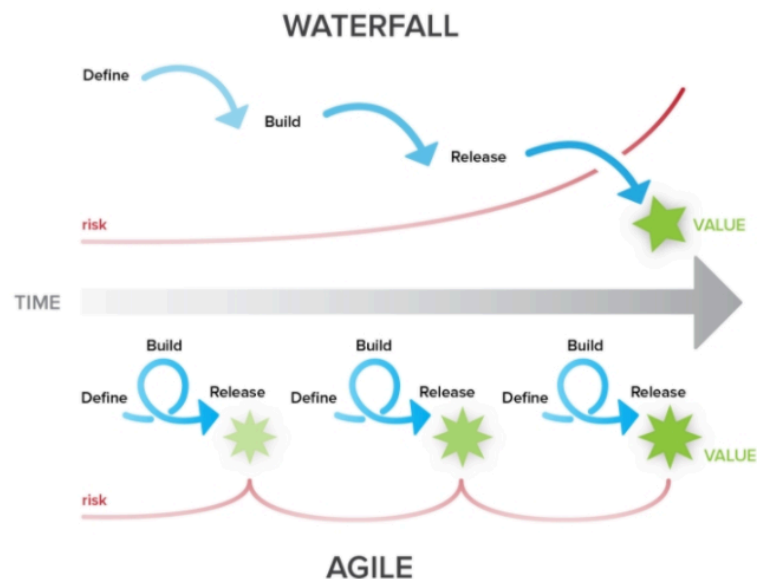**Information Technology**

FIT2001 – Systems Development

Seminar 2: System Development Approaches
　　　　Agile Software Development
　　　　Stakeholder management

Chris Gonsalvez

Our road map:

- Traditional vs. Agile approaches to developing systems

- A Focus on Agile development

- Stakeholder management

- What are Information Systems?

- How do we develop them? Systems Development (SDLC) – key phases

- Some System Development roles and skills

At the end of this seminar you will:

- Be aware of the different approaches to developing information systems

- Understand Agile software development - the Agile manifesto, the12 Agile principles and key concepts

- Be able to identify and understand different kinds of stakeholders and their contributions to requirements definition
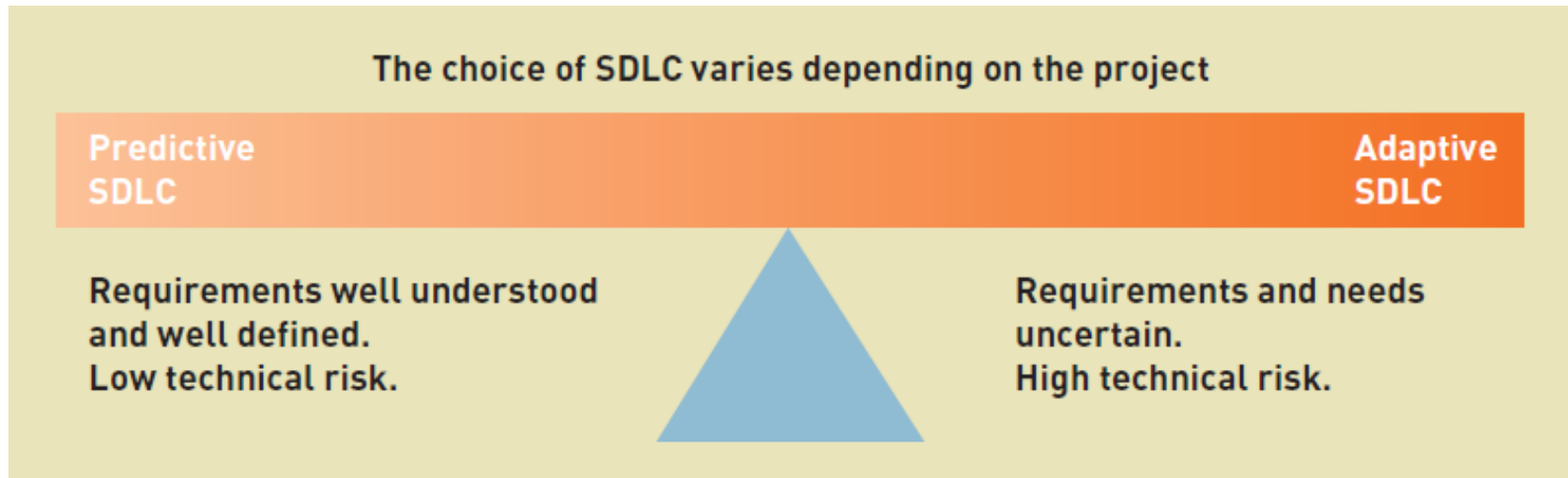
Information systems

"A system which assembles, stores, processes and delivers information relevant to an organisation (or to society) in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens."

Buckingham et al. (1987)

in Avison & Fitzgerald 2006, p. 23

We now have to consider how to develop them

Development approaches

The choice of SDLC varies depending on the project

Predictive SDLC — **Adaptive SDLC**

Requirements well understood and well defined.
Low technical risk.

Requirements and needs uncertain.
High technical risk.

Systems Analysis and Design in a Changing World, 6th Edition – Figure 8.1, p228
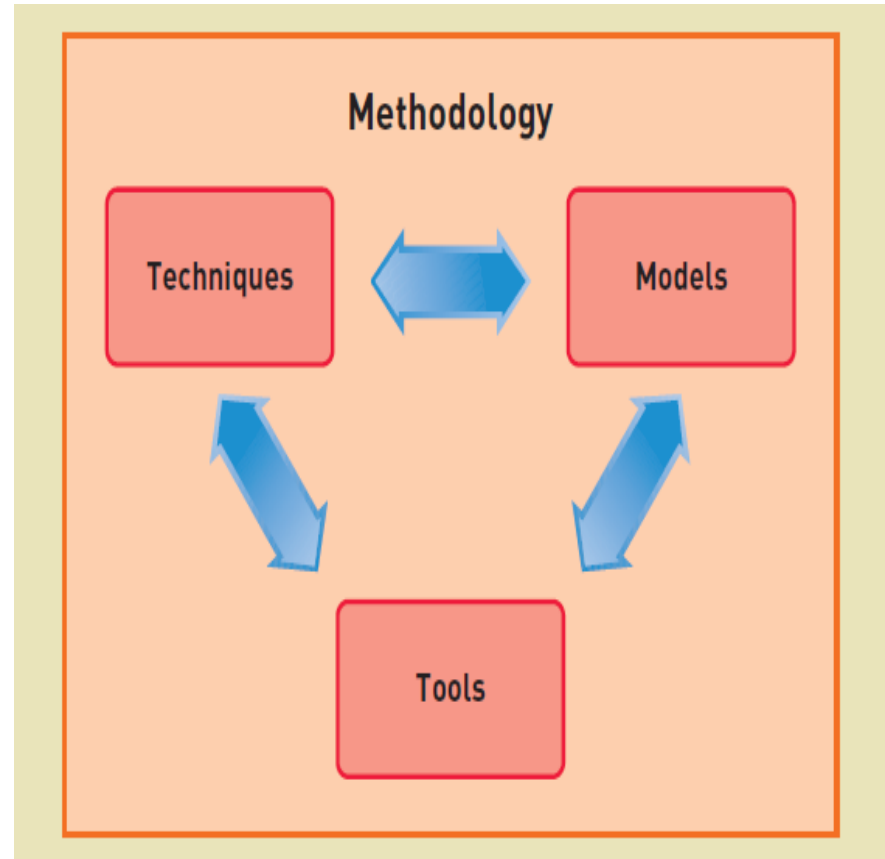
Most projects fall somewhere on this continuum

Frameworks / Methodologies definitions

Frameworks

- Provide structure and direction on a preferred way to do something − guidance while being flexible
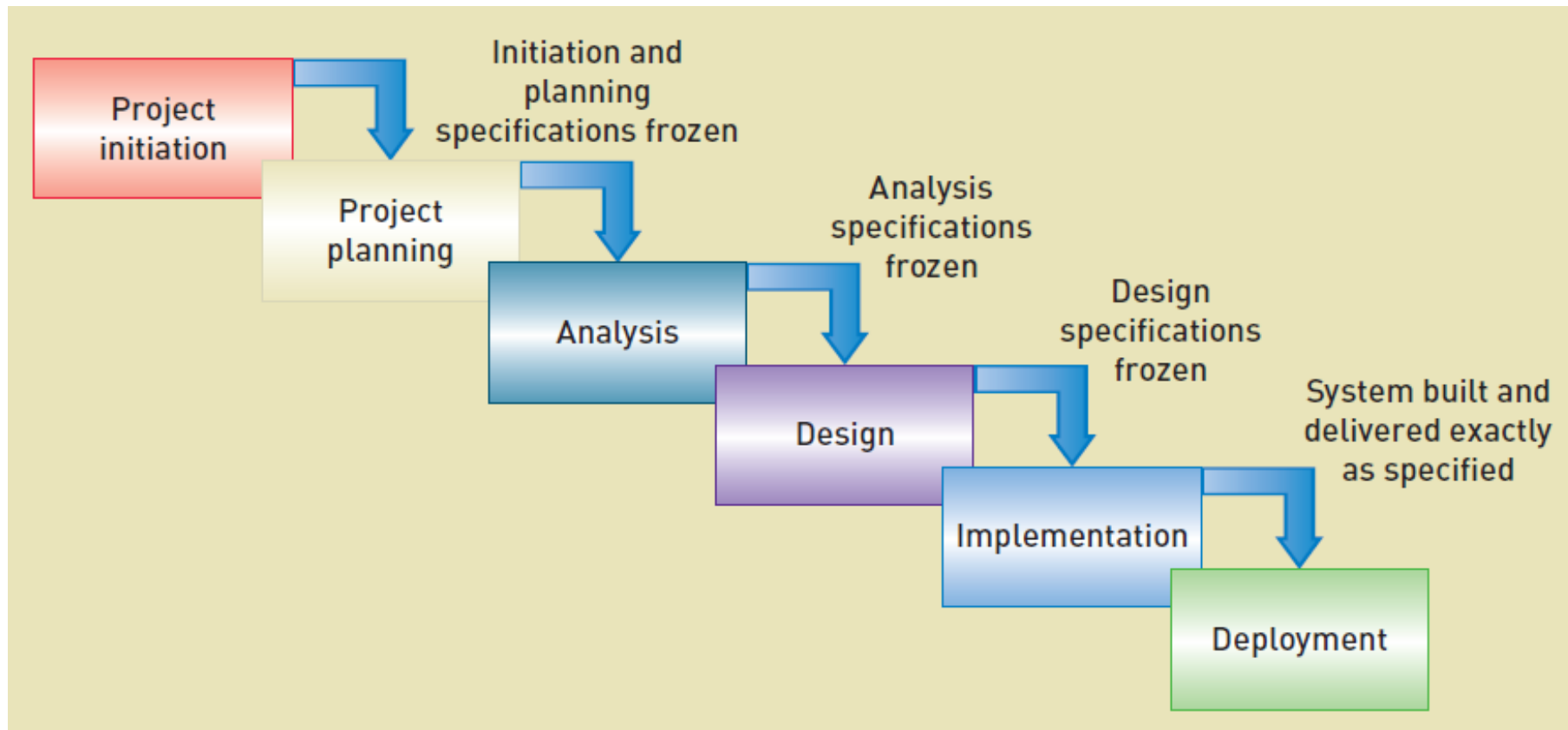
Methodologies

- A set of principles, tools and practices − conventions that an organisation / team agree to follow to achieve a particular goal.

Traditional Predictive thinking Structured Waterfall framework

Systems Analysis and Design in a Changing World, 6th Edition - Figure 8-3, p229

Waterfall framework

- Dominant since development methodology from early 70s to 90s

- Sequential stages − no overlap or iteration

- Strong emphasis on planning and specifications development

- Works well for clearly defined projects - requires thorough planning, extensive project documentation and tight control over the development process.

- Issues - tends to be slow, costly and inflexible.
  - Inability to adjust the product to the evolving market requirements often results in a huge waste of resources and the eventual project failure

Rarely developed this way anymore

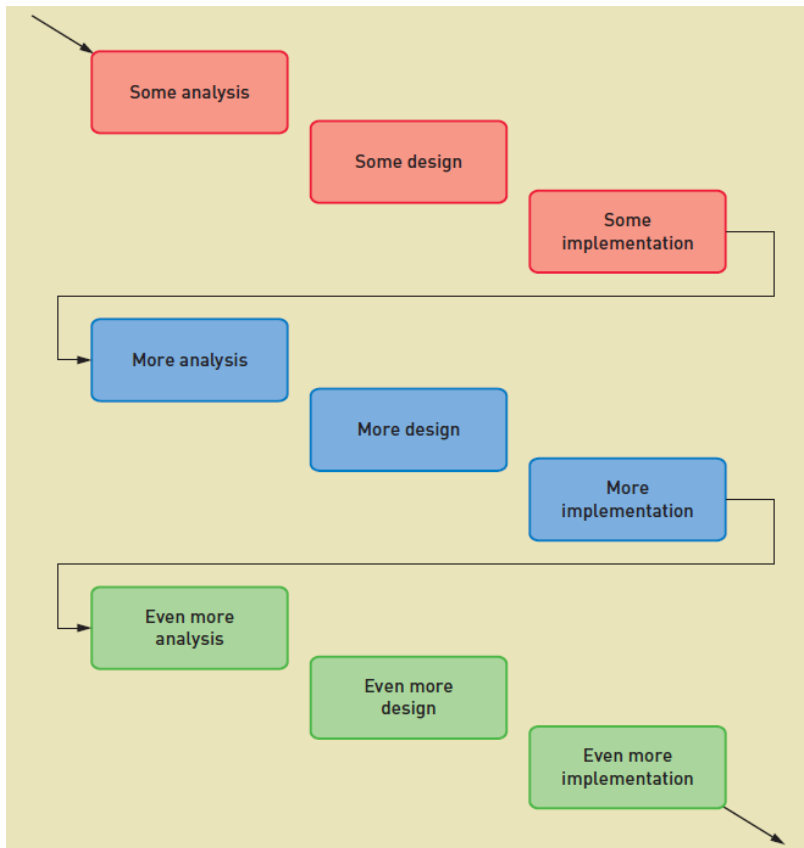# Development often moved towards an Adaptive framework

- More flexibility, but still assumes predictive planning and sequential phases
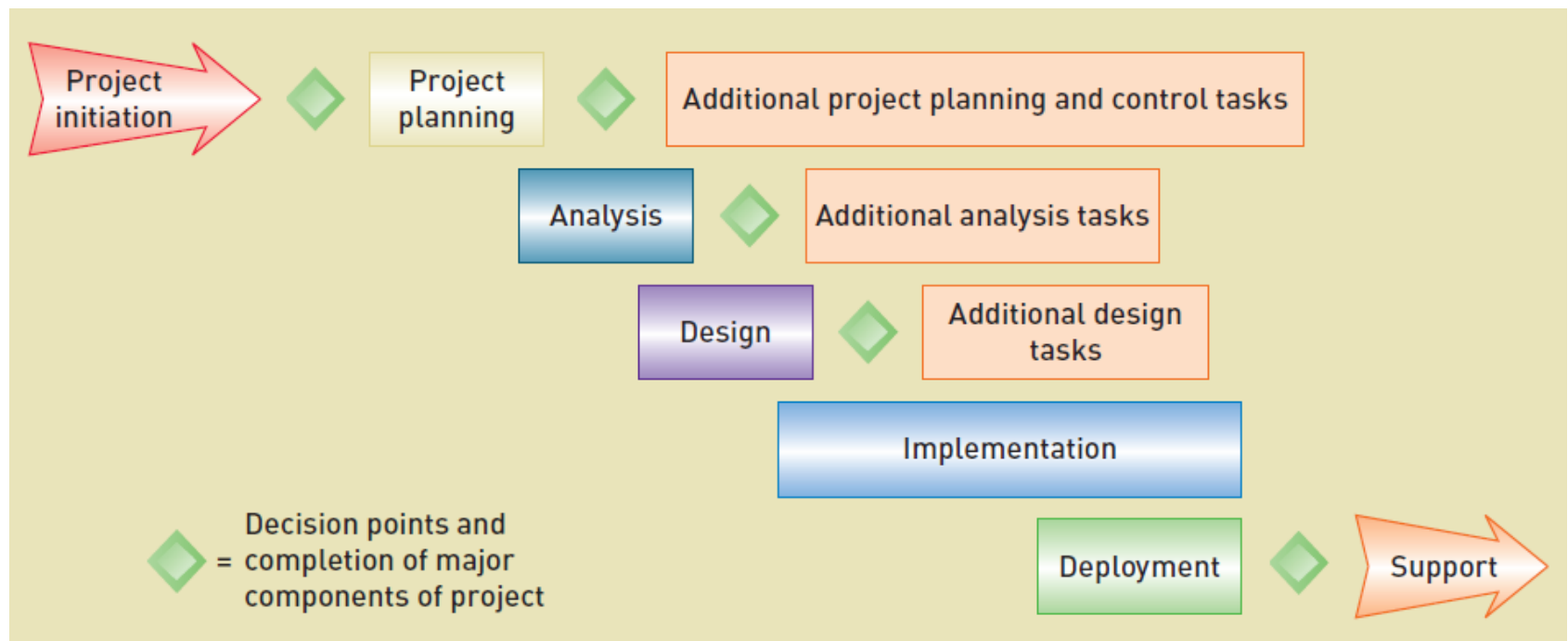
Systems Analysis and Design in a Changing World, 6th Edition – Figure 8.4, p230
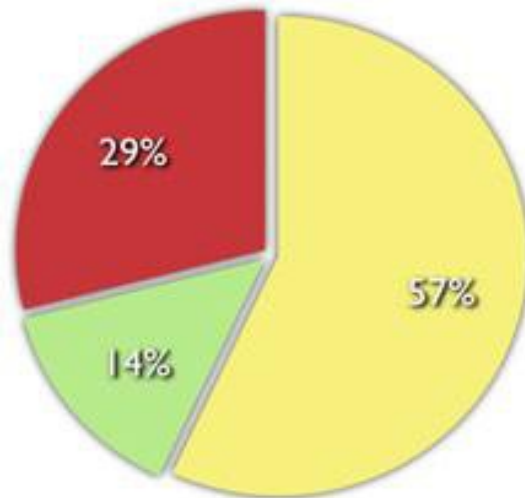
Adaptive thinking
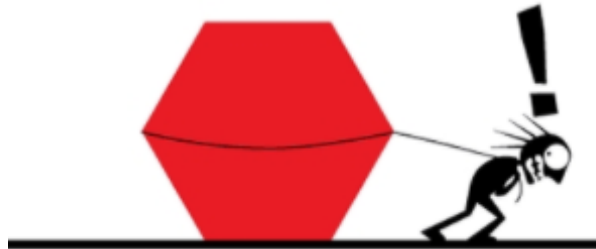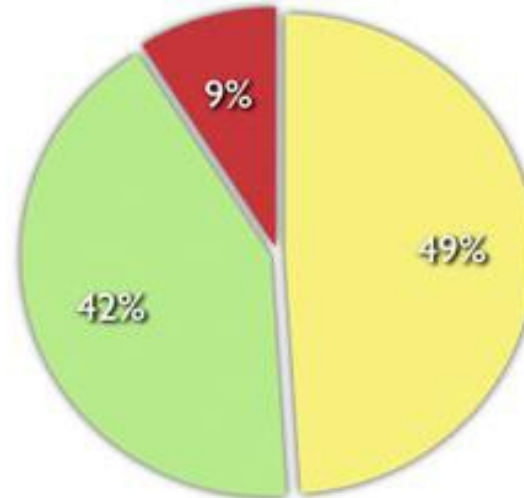# Agile Iterative frameworks

Iterative development

An approach to system development in which the system is "grown" piece by piece through multiple iterations

Systems Analysis and Design in a Changing World, 6th Edition – Figure 8.6, p231

Agile vs. Waterfall

THE WATERFALL PROCESS

THE AGILE PROCESS

29%

14%

57%

9%

42%

49%

Successful

Challenged

Failed

Source: The CHAOS Manifesto, The Standish Group, 2012.

Results of projects conducted by The Standish Group from 2002 to 2010

Agile vs. Waterfall Metrics

Metric	Waterfall	Agile
Planning scale	Long-term	Short-trem
Distance between client and developer	Long	Short
Time from specification to implementation	Long	Short
Time to discover issues	Long	Short
Ability to meet deadlines	Poor	Good
Ability to respond quickly to change	Low	High

MONASH University

AGILE SYSTEMS DEVELOMENT

Agile History

- Incremental approaches started as far back as the late 1950s – building software for IBM

- Mid 90s – issues with developing software, started mixing old and new ideas, focus on close collaboration with users, frequent delivery of business value – frameworks such as SCRUM, Extreme Programming started to appear

- 2001 – 17 software developers caught up to find commonalities in developing software – they disagreed about a lot, what they agreed upon became The Agile Manifesto

What is Agile?

- Agile frameworks take an iterative approach to software development - project consists of small iterations

- Each iteration is a miniature project with a well defined scope

- At the end of each sprint, a potentially shippable product increment is delivered.

- Every iteration sees new features added to the product, which results in the gradual project growth.

- With the features being validated early and regularly, the chances not delivering what the clients wants reduces significantly. Let's summarize the main Agile aspects:

Agile Manifesto - Values

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

MONASH University

12 Agile Principles

Satisfy
The Customer

Welcome Changing
Requirements

Deliver Working
Software Frequently

Collaborate
Daily

Motivated
Individuals

Face-to-face
Conversation

Measure Of Progress
Through Working Product

Promote Sustainable
Development

Continuous Attention To
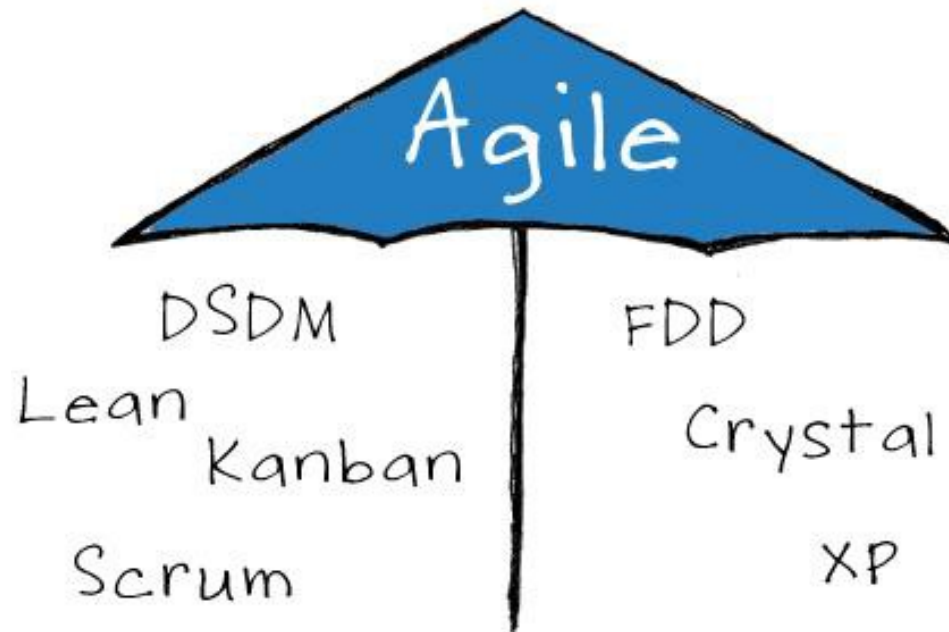Technical Excellence

Simplicity
Is Essential

Self-organizing
Teams

Regularity Reflect On
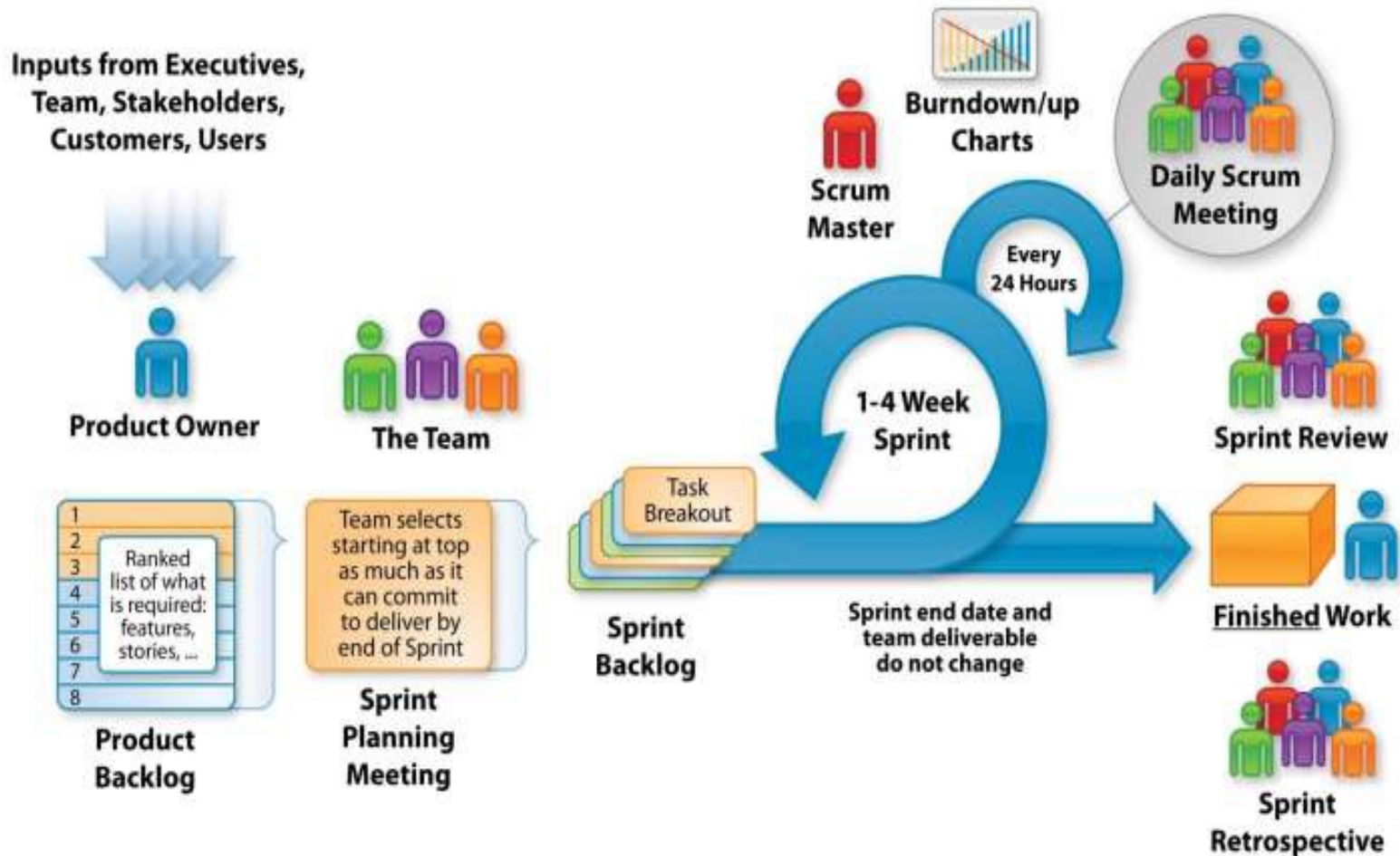Continuously Improving

Agile frameworks

MONASH University

Agile Framework example - SCRUM

- A framework based on agile principles

- Based on continuous improvement in product and process

- Delivers software (value) frequently

- A scrum project is a series of iterations called Sprints – typically 2-4 weeks long, based on an inspect and adapt cycle

- Produces outputs iteratively and incrementally, thus reducing risk and enhancing visibility

https://www.agilelearninglabs.com/resources/scrum-introduction/

SCRUM Framework

SCRUM Roles

Product Owner
The person with the project vision, who breaks down the project into tasks and then prioritizes them, so everything meets the objectives and goals of the customer, market and organization.

ScrumMaster
This is the expert who helps to make sure the project is following the Scrum framework. While they guide and coach the team, their responsibility is to the process rather than the team.

Development Team
These are the people who execute the project. They're autonomous, self-directed and each has a unique set of skills that address the needs and the project and will work together with the whole team.

PROJECT MANAGER

Product owner
Client's representative, defines and prioritises product features, accept or reject work items

Scrum Master
Coach for scrum team, applying agile principles, ensures team's productivity, builds a successful team

Development Team
5-9 members in a self-organizing, high performance, cross-functional team team (Developer, Tester, BA)
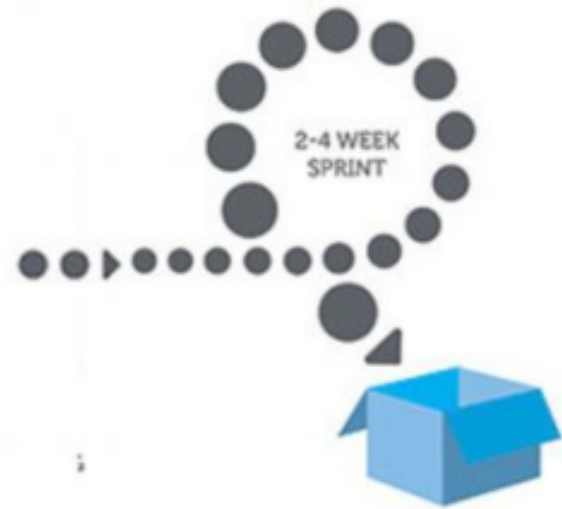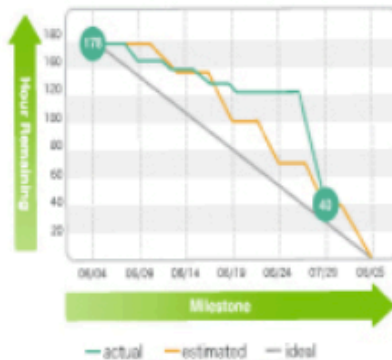
SCRUM Artifacts
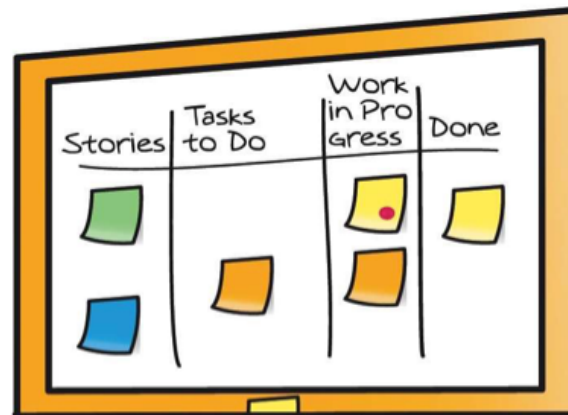

Product Backlog

Sprint Backlog

2-4 WEEK SPRINT

Product Increment

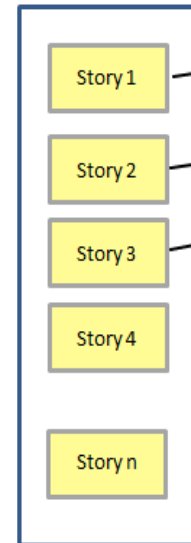**Burndown Chart**

Kanban board

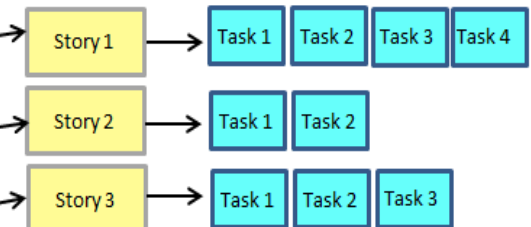# Product & Sprint Backlog

Product Backlog

- The single source of requirements

- Cumulative list of desired deliverable for the project – every feature, enhancement, bug fix, documentation requirement, every bit of work required by the team

- Prioritised to maximise value

Product Backlog

Story 1
Story 2
Story 3
Story 4
Story n

Sprint Backlog

Story 1 → Task 1 | Task 2 | Task 3 | Task 4

Story 2 → Task 1 | Task 2

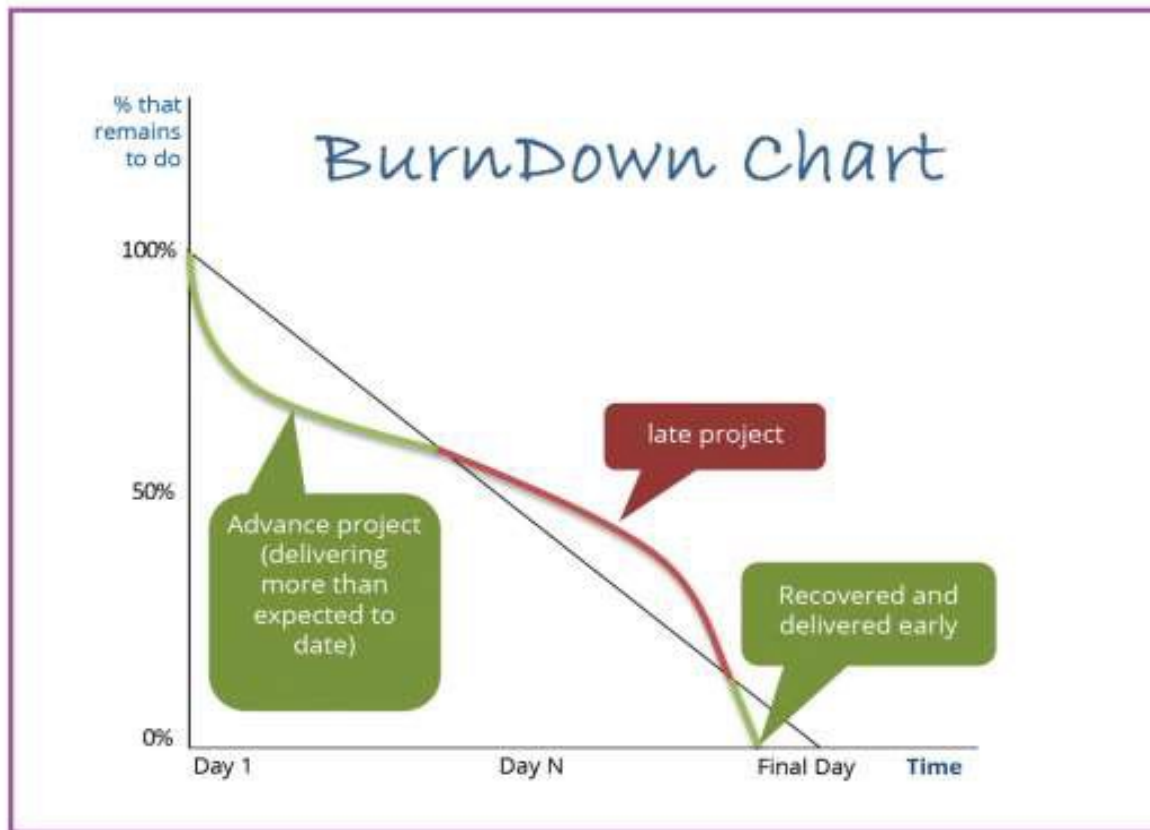Story 3 → Task 1 | Task 2 | Task 3

Sprint Backlog

- A list of tasks the team must complete to deliver an increment of functional software at the end of each Sprint.

- Once decided Team owns the Sprint Backlog – only they can decide on scope change

Sprint BurnDown Charts

▸ Shows the total estimated work remaining for the entire forecasted sprint backlog against time

Task Board (Kanban)

- Allows visibility, transparency across the project

- Displays the live status of team work and focus

- Most have – Backlog, To-do, In Progress (Doing) and Done status.

- Boards can be physical or digital

- They are often physical providing strong motivation for the team

MONASH University

Product Increment

- A Product Increment is the end product for each sprint. It must:

  – Be of high enough quality to be given to users
  – Meet the Scrum team's current definition of DONE
  – Be acceptable to the product owner.
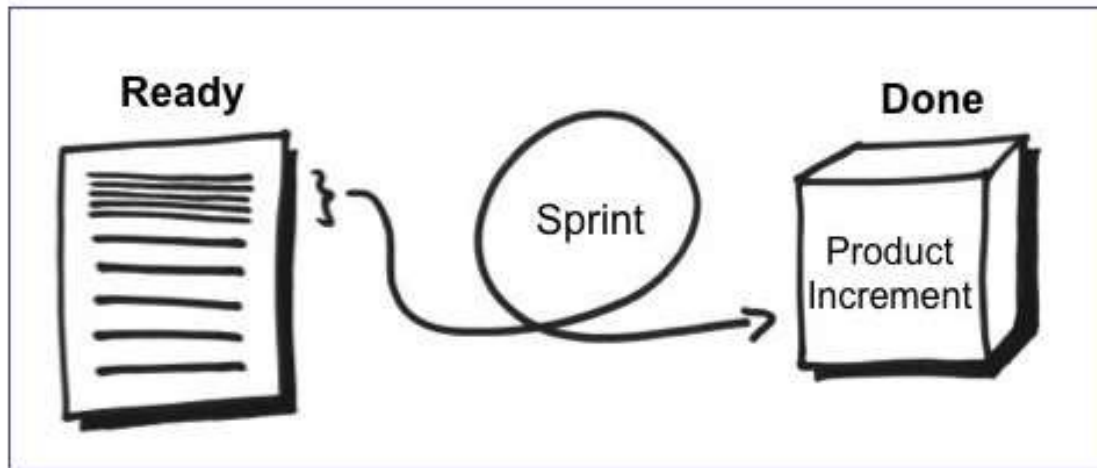
SCRUM Activities − A sprint cycle

The Sprint is a timebox of 2-4 weeks during which the team produces a potentially shippable Product Increment.
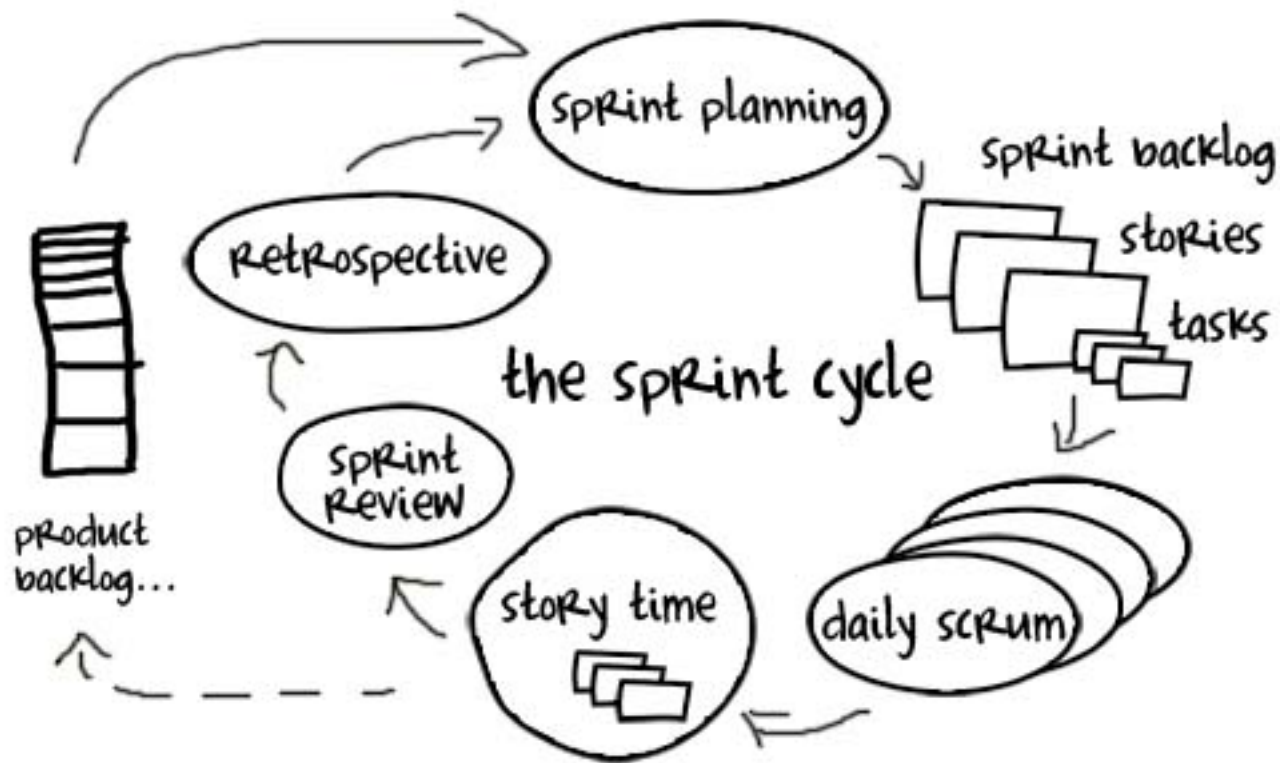
Sprint Activities

Start of the Sprint - Sprint Planning

Determine which items from the product backlog they will work on during the Sprint.
The end result the Sprint Backlog – defines the scope of the sprint
1. Discussion with product owner – WHAT will we do
2. Team does the detailed plan – HOW will we do it

During the Sprint - Daily Stand Up

Short (usually limited to 15 minutes) discussion where the team coordinates their activities for the following day. The only focus of the Daily Stand Up:

1. What I did since last daily scrum meeting
2. What I am planning to work on today
3. Impediments (Issues/blockers) if any?

Sprint Activities - At the end of the Sprint

Sprint Review

- The entire team does a review to get 'Product Increment' feedback from the Stakeholders

- Feedback goes into the 'Product Backlog' for future consideration.

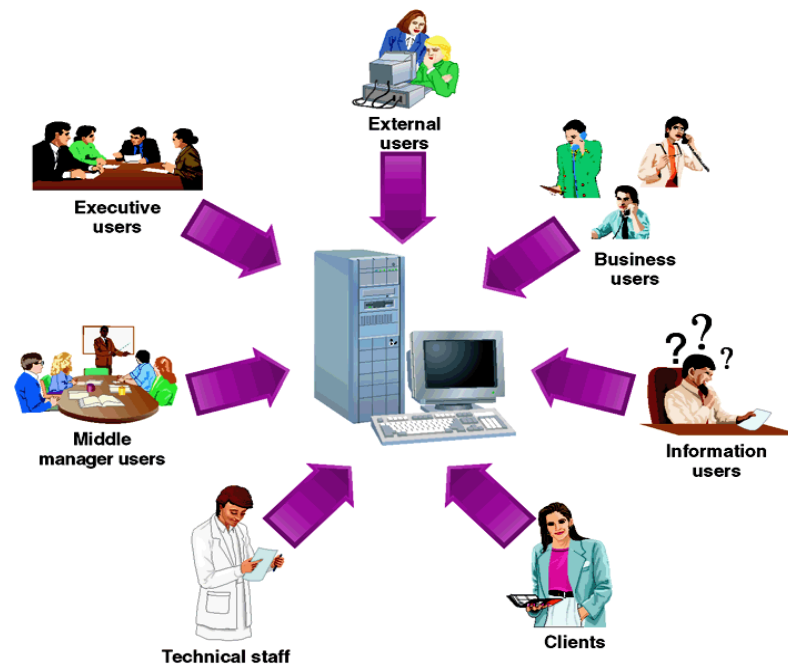- Not intended to provide a status report

Sprint Retrospective

- The team (including product owner) reflect upon how things went during the previous sprint
    - What went well
    - What could be improved

- They identify adjustments they can make moving forward
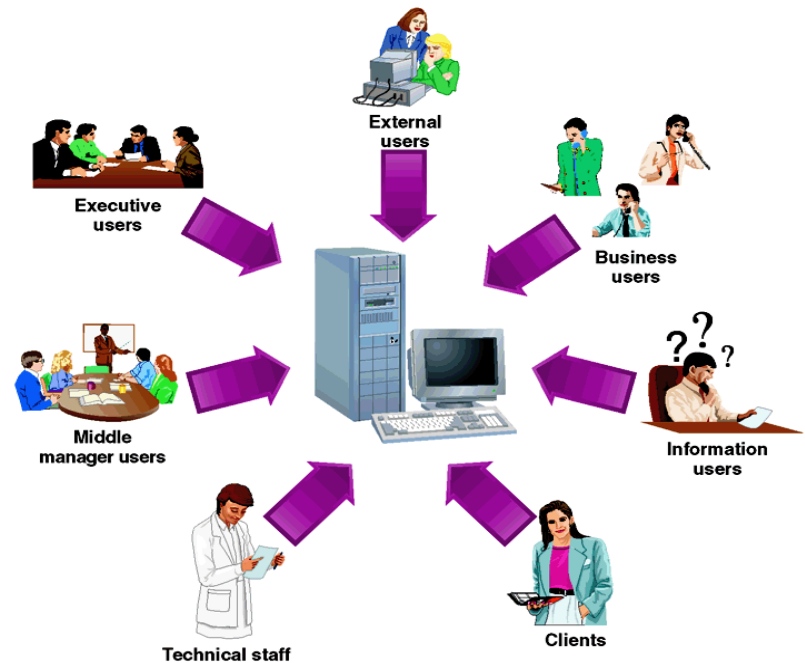
STAKEHOLDER MANAGEMENT

Who do you get these requirements from?

…. a range of stakeholders

People interested in the success (sometimes failure) of a system

It might include senior executives, project organisation roles, client organisation roles, system developers, IT operations, customers, etc.

MONASH University

Identify stakeholders

Need to identify the correct people:

Identify stakeholders

Need to find out:

- Who gains and who loses from this development?

- Who controls change management of processes?

- Who will make the decisions?

- Who procures IT systems and who decides what to buy?

- Who controls resources?

- Who has specialist skills the project needs?

- Who has influence?

- *Project owner (facilitates progress) &*

  Project sponsor (sells change to users) – both very important

  - Can have a major impact on project success

Prioritise and understand your stakeholders

- Someone's position on the grid shows you the actions you have to take with them

- Influence = Power
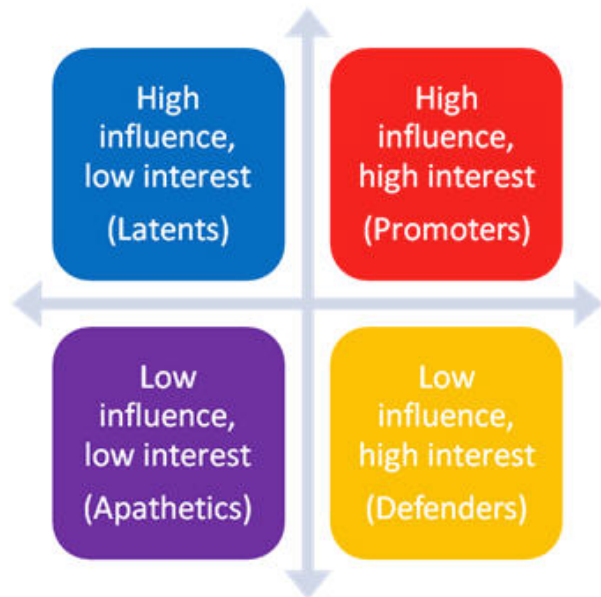
- You need to understand how they feel about the project

- Determines how you engage / communicate with them

Can block or advance your project

Managing stakeholder expectations?

- Systems fail if they don't meet expectations

Requirement:
Create a means to transport a single individual from home to place of work.

Management Interpretation	I T Interpretation	User Interpretation

Your role …

… to get the disparate views

… create a shared vision

Workshop Preparation

- Watch Seminar 2

- Watch the Industry Seminar

- Watch the Agile video in the resources

Thanks for watching

See you next week

Resources:

Prescribed text:

– Satzinger, J. W., Jackson, R.B., and Burd, S.D.(2016) Systems Analysis and Design in a Changing World, 7th Edition, Cengage Learning, Chapter 10

Agile basics:

Agile Alliance - https://www.agilealliance.org/agile-essentials/

http://www.agilelearninglabs.com/resources/scrum-introduction/

Agile Video - https://www.youtube.com/watch?v=OJflDE6OaSc