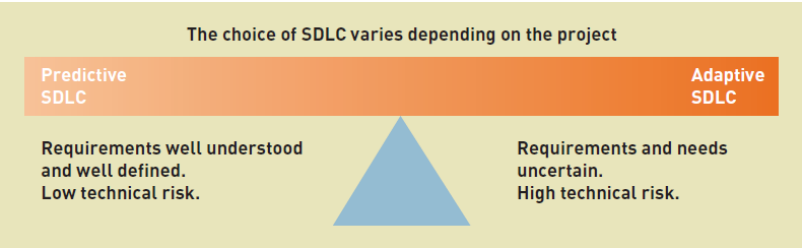
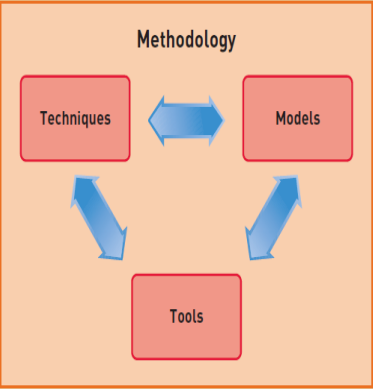
**Development approaches:**



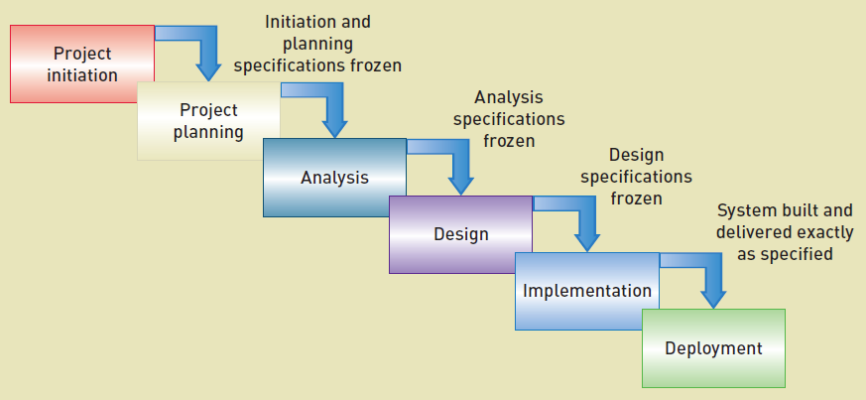
Most projects fall somewhere on this continuum

**Frameworks / Methodologies definitions:**

****

* Frameworks- Provide structure and direction on doing something
* Methodologies- A set of principles, tools and practices the team agree to follow to achieve a particular goal

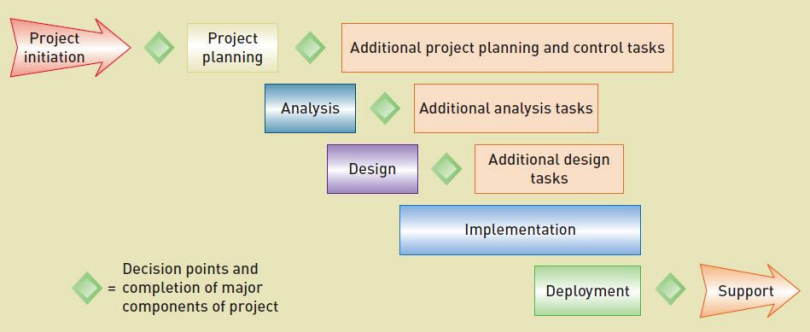
**Traditional Waterfall framework:**

****

* Dominant 70s to 90s
* Sequential stages – no overlap or iteration
* Strong emphasis on planning and specifications development
* Works well for clearly defined projects - 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 market requirements often results in waste of resources

**Development often moved towards an Adaptive framework**

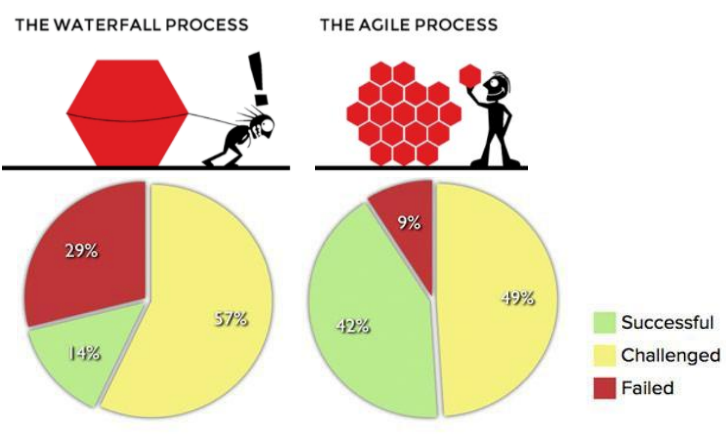
More flexibility but still assumes predictive planning and sequential phase

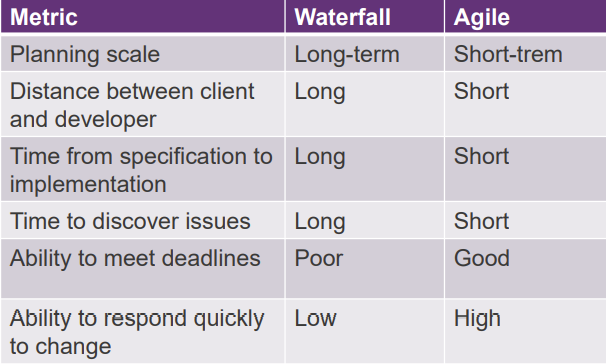
****

**Agile Iterative frameworks**

The system is “grown” piece by piece through multiple iterations

Agile vs. Waterfall:

****

****

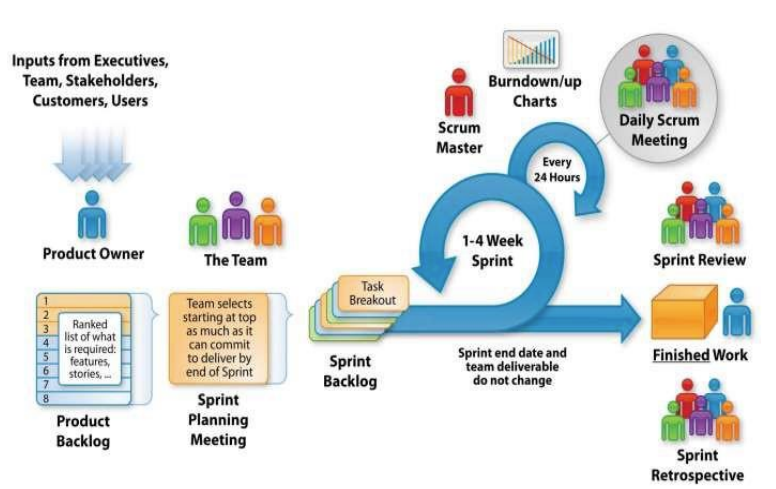
* Agile frameworks take an iterative approach
* Each iteration is a small project with the defined scope
* At the end of each sprint, a potentially shippable product increment is delivered.
* Every iteration sees new features added to the product
* Agile Manifesto - Values
  + We are uncovering better ways of developing software by doing it and helping others do it
  + **The four foundational values of the Agile Manifesto are a guide to the development and delivery of high-quality, working software.**
    1. Individuals and interactions over processes and tools - Understanding the requirement of the clients is essential to developing great products. Individual and interactions help create customer-focussed products. Developing good products requires effective team-work and interactions, irrespective of the processes and tools used. Communication is an example of the difference between valuing individuals versus process. In the case of individuals, communication is flexible and happens when a need arises. In the case of process, communication is scheduled.
    2. Working Software Over Comprehensive Documentation - The amount of time spent on documenting user and technical requirements in tradition development often led to project delays and unmet client expectations. Agile focuses on providing enough documentation to get the job done, without getting bogged down in excessive detail.
    3. Customer Collaboration Over Contract Negotiation - This value describes a customer who is engaged and collaborative throughout the development process, making it far easier to meet their requirements. With traditional development, customers negotiate the requirements for the product, often in great detail, prior to any work starting, with contracted delivery points creating a wall between the customers and developers.
    4. Responding to Change Over Following a Plan - Traditionally change was considered a cost to be avoided. The focus was on following a detailed plan with a defined set of features to deliver. Agile considers change as a valuable agent and responds and adapts to it with the iterative model.
    5. Changes are accepted at any time during the development effort depending on the business value of the change, the Product Owner's acceptance, and the ability of the Dev Team to respond in a timeframe acceptable to the Product Owner.

Framework example – SCRUM(?)

A framework based on agile principles

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

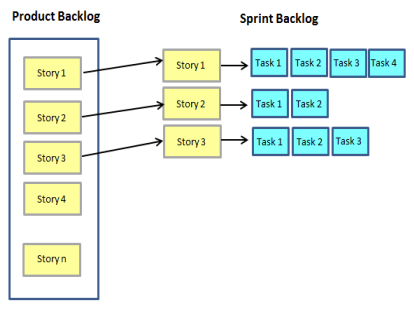
****

Product owner- defines and prioritises product features, accept or reject work items

Scrum Master- applying agile principles, ensures team's productivity, builds a team

Development Team- 5-9 members, high performance, cross-functional team

**Backlog & Sprint Backlog**

****

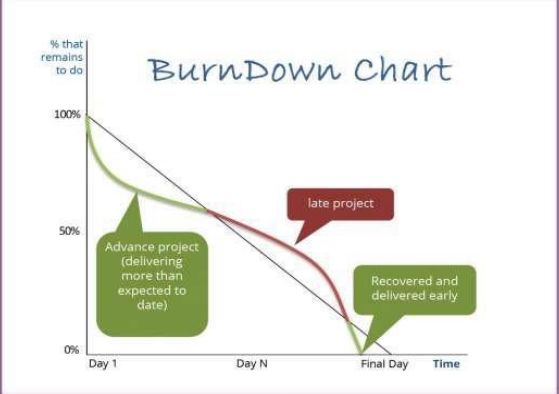
Product Backlog:

* single source of requirements
* every feature, enhancement, bug fix, documentation requirement, every bit of work required by the team
* Prioritised to maximise value

Sprint Backlog

list of tasks the team must complete delivering an increment of functional software at the end of each Sprint

**Sprint Burndown Charts**

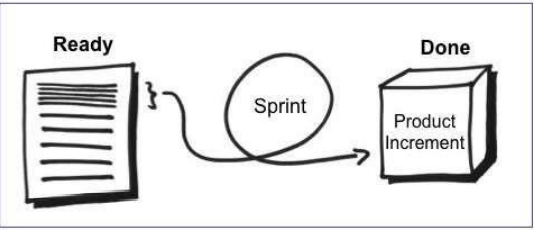


total estimated work remaining for the entire forecasted sprint backlog against time

**Task Board (Kanban)**

* visibility, transparency across the project
* Displays the live status of team work and focus
* Backlog, To-do, In Progress (Doing) and Done

**Product Increment**

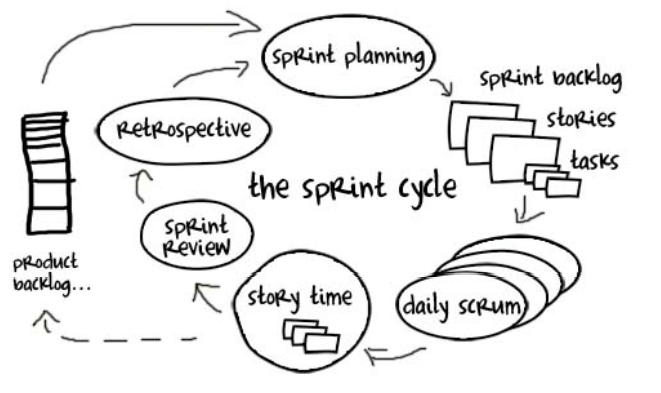
****

end product for each sprint：

* high enough quality to be given to users
* Scrum team's current definition of DONE
* acceptable to the product owner

**SCRUM Activities：**

A sprint cycles

****

Start of the Sprint - Sprint Planning：

* Determine which items from the product backlog they will work
* The result of Sprint Backlog – defines the scope of the sprint
  + Discussion with product owner – WHAT will we do
  + Team – HOW will we do it

During the Sprint - Daily Stand Up：

* Short discussion following day activities:
  + What I work on today
  + Any Issues/blockers
  + What I did since last daily scrum meeting

End of the Sprint – Review and Retrospective：

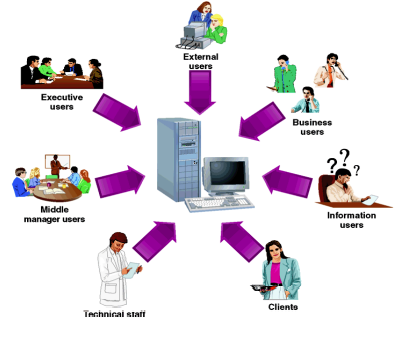
Review:

* 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

Retrospective:

* The team reflect upon how things went during the previous sprint
  + What went well
  + What could be improved

# STAKEHOLDER MANAGEMENT

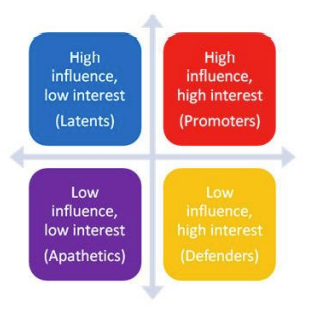


Stakeholders- People interested in the success

**Identify stakeholders:**

* Who gains and who loses?
* Who controls change management of processes?
* Who will make the decisions?
* Who controls resources?
* Who has influence?
* Who has specialist skills the project needs?
* who decides what to buy?

**Prioritise and understand your stakeholders:**



* Influence = Power
* Determines how you engage / communicate with them
* Systems fail if they don’t meet expectation