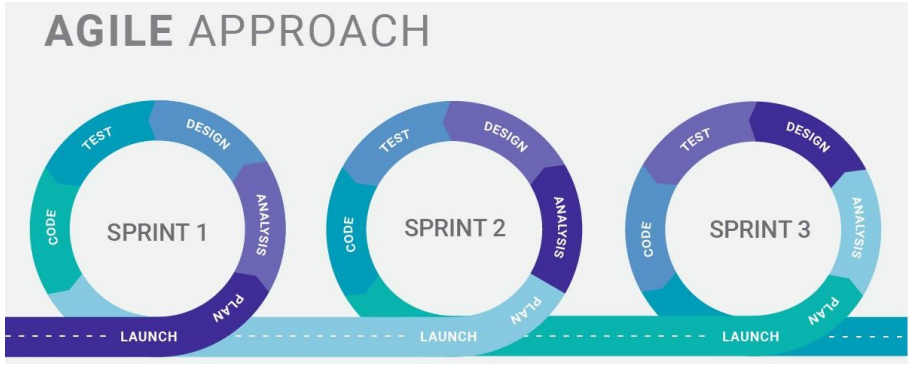
Agenda

- Agile Methodology
- Empirical Process
- Scrum Framework
- Scrum Roles
- Scrum Ceremonies
- Scrum Artifices
- User Stories & Estimate
- Planning & Monitoring



Agile Methodology

- Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
- The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change.
- Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams

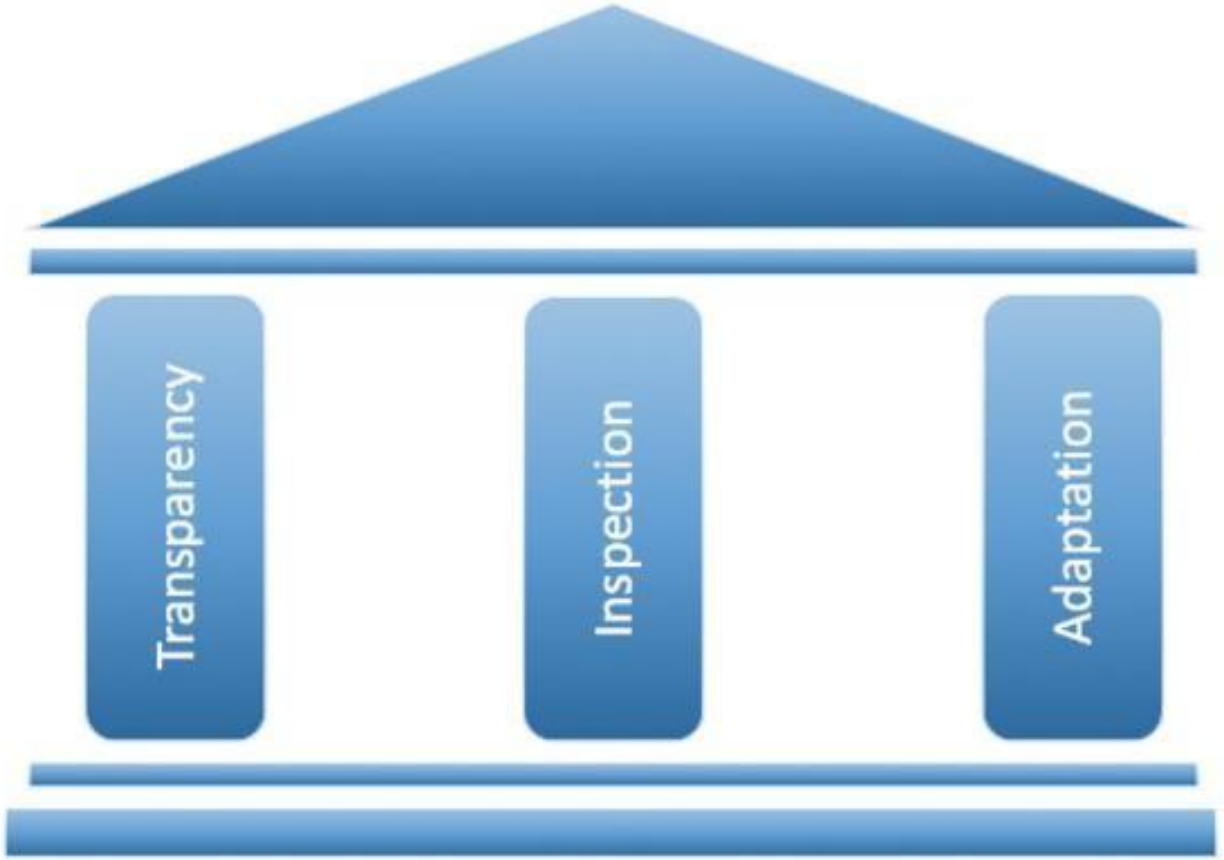
Definition

- **A framework within which people can address complex adaptive problems while productivity and creativity delivering the products of the Highest possible values** in shortest possible time.
- Scrum is simple but **not** easy
- Light weight , simple to understand **difficult** to master
- About People – working together

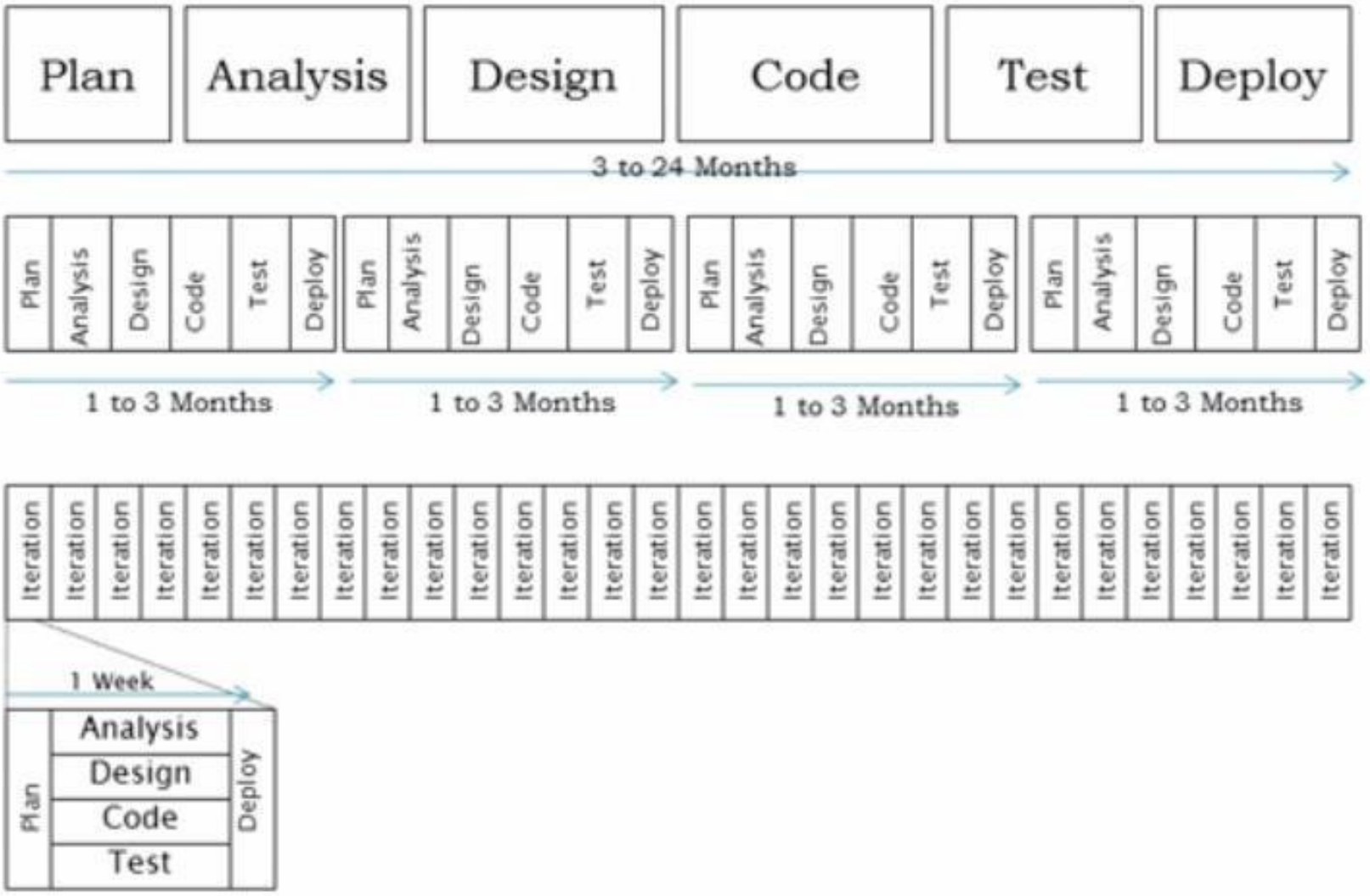
Organization own Processes Tools & Techniques		
Roles	Ceremony	Artifacts
Rules		

Empirical Process

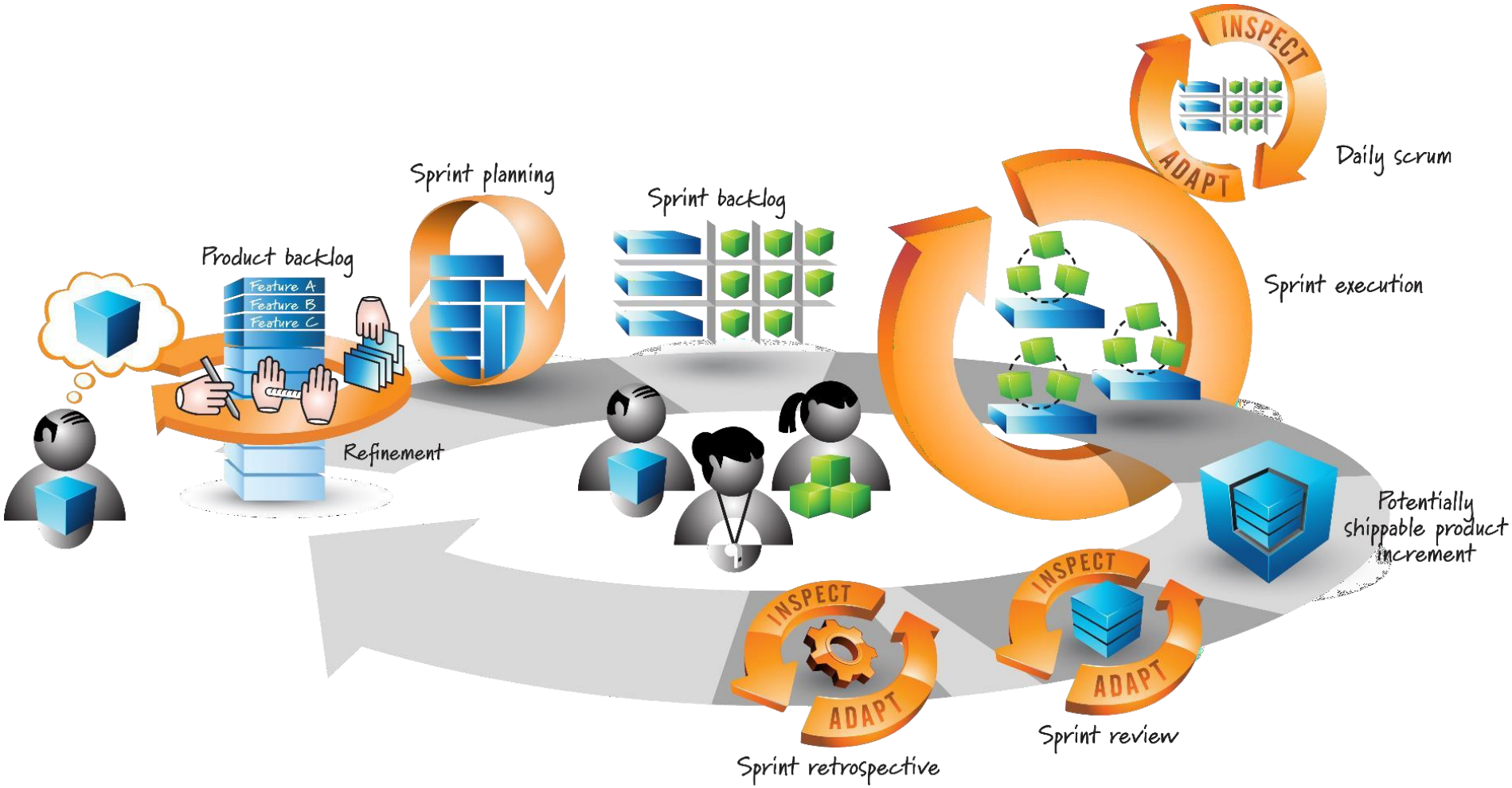
Practical rather theoretical



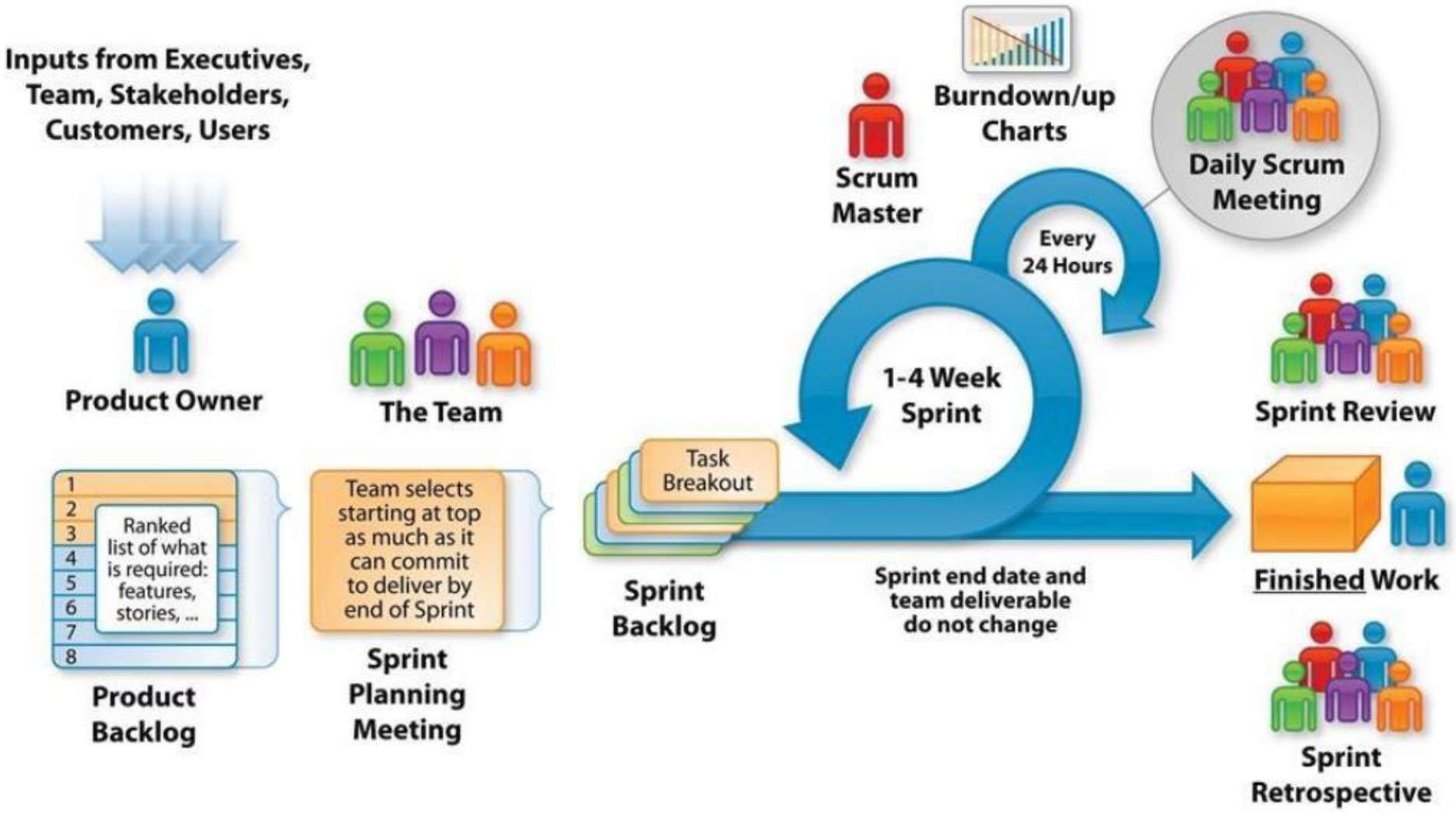
Waterfall Vs Scrum Life cycle



Scrum Framework



Scrum Framework



Scrum Practices

Roles

- Product Owner
- Scrum Master
- Development Team

Activities

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint execution
- Sprint Review
- Retrospective
- Backlog Grooming

Artificats

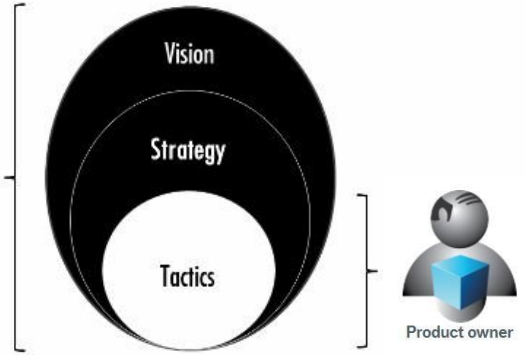
- Product Backlog
- Sprint Backlog
- Shippable product

Scrum Role : Product Owner

An Individual who owns the product on behalf of the organization

Product Owner Responsibilities

- Big Product Owner
  - Strategy and Roadmap
  - Prioritize Feature & Collaborate
  - Financial Forecast
  - ROI
- Small Product Owner
  - Manage Product Backlog
  - Release & Sprint Planning
  - Write User Stories & Acceptance Criteria
  - Work with the development team and Scrum Master



Who will be the product Owner ?

- Internal Product
  - Business Representative
- Outsource Development
  - Customer Representative
- Architecture Development
  - Technical Person who can priorities the backlog
- Commercial Product
  - Product Management or Product Marketing Team

Scrum Role : Scrum Master

Scrum Master Role is Simple But It is **not** Easy

Scrum Role : Scrum Master

- Servant Leader to Product owner, Development Team and Organization
- Responsible for scrum values and principles being understood and enacted
- Remove Impediments & Shield the team from interferences
- Active Facilitator
- Coaching the Product owner on Product Backlog Refinements
- Coaching the team on self organizing and self motivating
- Leading the organization in its scrum adoption
- Look for people aspects of the job
- Arm chair Psychologists

Scrum Master Skills/Characteristics

- People Management
- Training/ Mentoring / Coaching
- Active Listening
- Emotional Intelligence
- Powerful Questioning
- Proactive
- Patient

Who Will be the scrum Master ?

- Resource Manager
- Project Manager
- Team leader

Scrum : Development Team

- Typically 5-9 People
- Cross Functional
- T Shaped
- Members should be Full Time
- Self organizing
- Self Directing
- Membership change not recommended in between Sprint
- Work At sustainable pace
- Focus & Committed

Development Team Responsibility

- Plan the sprint
- Perform Sprint Execution
- Daily Scrum
- Inspect & Adapt Product & Process
- Product Backlog Grooming

Artifact : Product Backlog

- The Requirements
- List of desired work Items
- Definition of Ready

Product Backlog Characteristics

- Good product backlogs should be DEEP (Coined by Roman Pichler and Mike)

- **D**etailed appropriately
- **E**mergent
- **E**stimated
- **P**rioritized

Artifact : Sprint Backlog

- Decompose the Product Backlog Items
  - Engineering Tasks
  - Acceptance Criteria
- Language which development team understood
- List emerges during the Sprint.
- Transparent & Visible to the Team
- Each ongoing task identifies those responsible for doing the work
- Each Tasks has information about estimated amount of work remaining on the task on any given day during the Sprint.

Artifact : Increment

- Increment is the sum of all Product Backlog items completed during a sprint and all previous sprints
- At the end of a Sprint, the new Increment should “Done”
- It must be useable condition (Potentially Shippable Product)
- Release increments early and frequently Vs deliver the finished product in one go

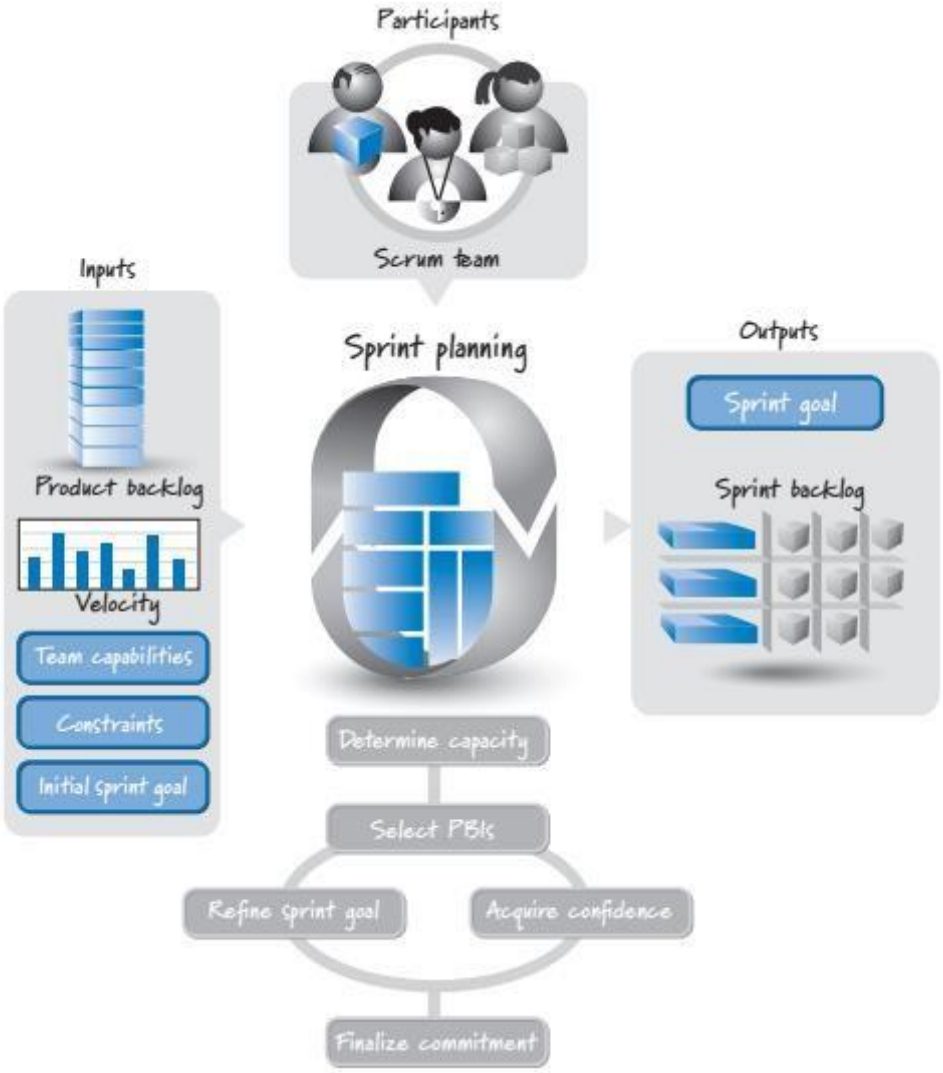
Definition of Done

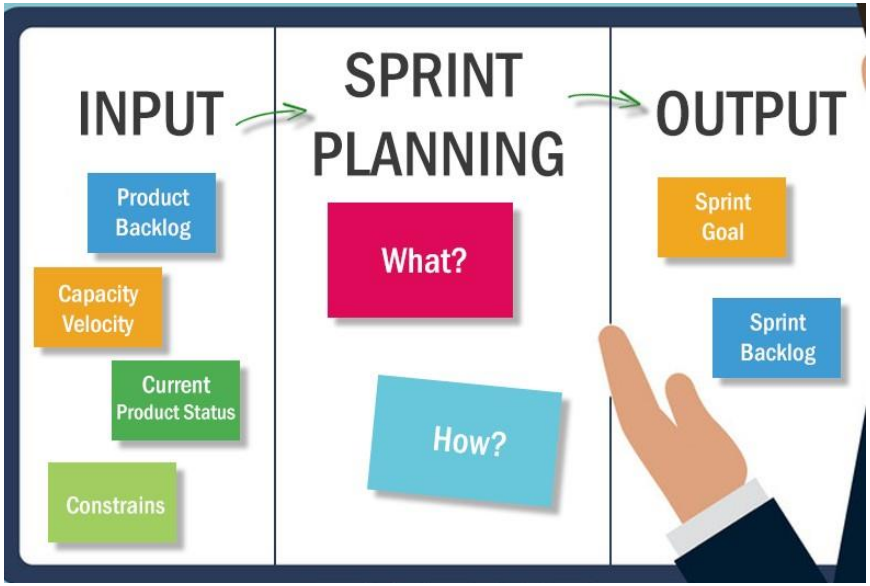
- DoD is a Check list to understand what done means.
- Varies significantly Per Scrum Team
- Sprint delivery adheres to the Definition of Done
- Definition of Change during the project.

Sprint

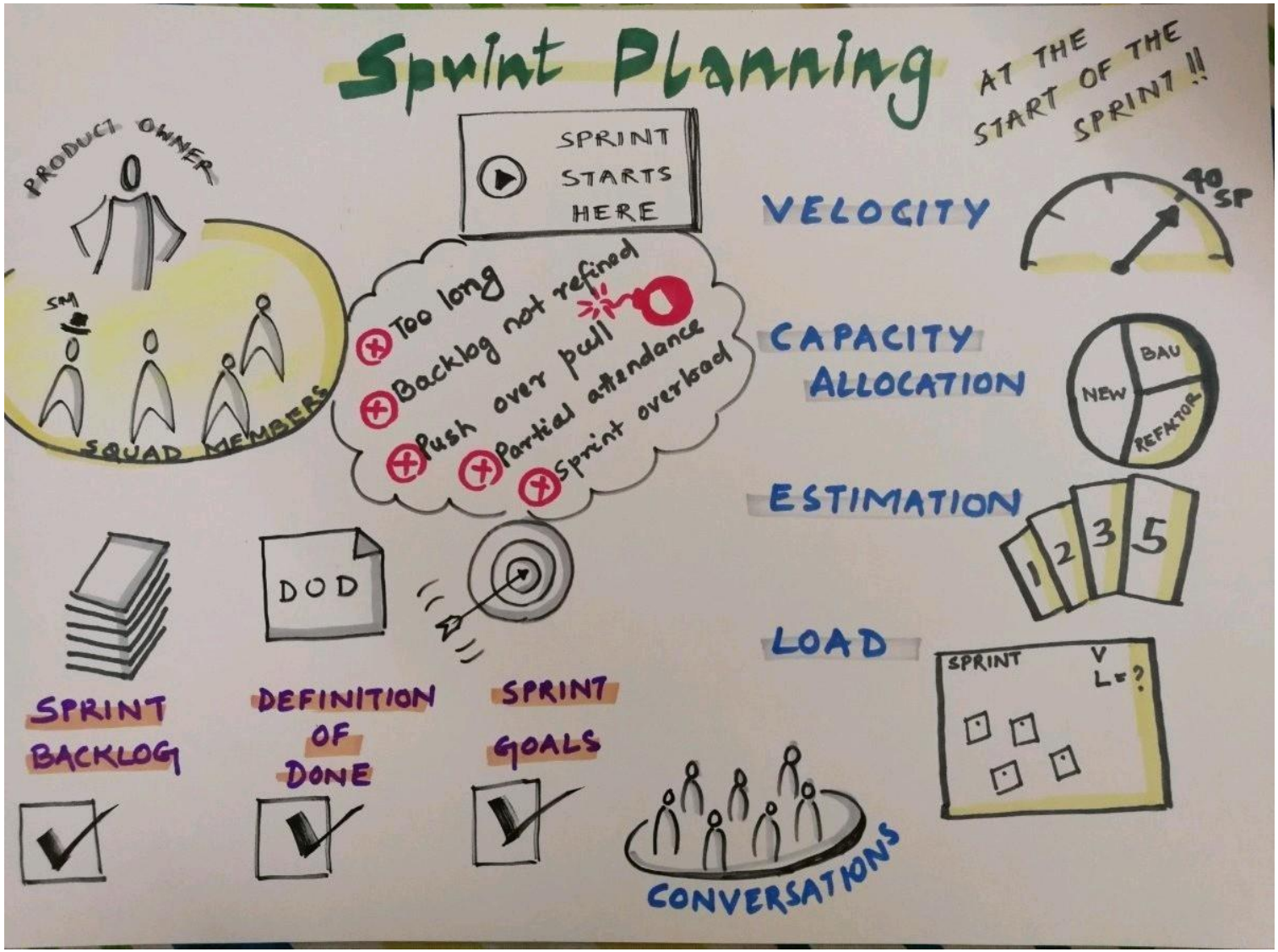
- Time boxed
- Short Duration
- Consistent Duration
- No Goal Altering Changes
- Definition of Done

Sprint Planning Meeting



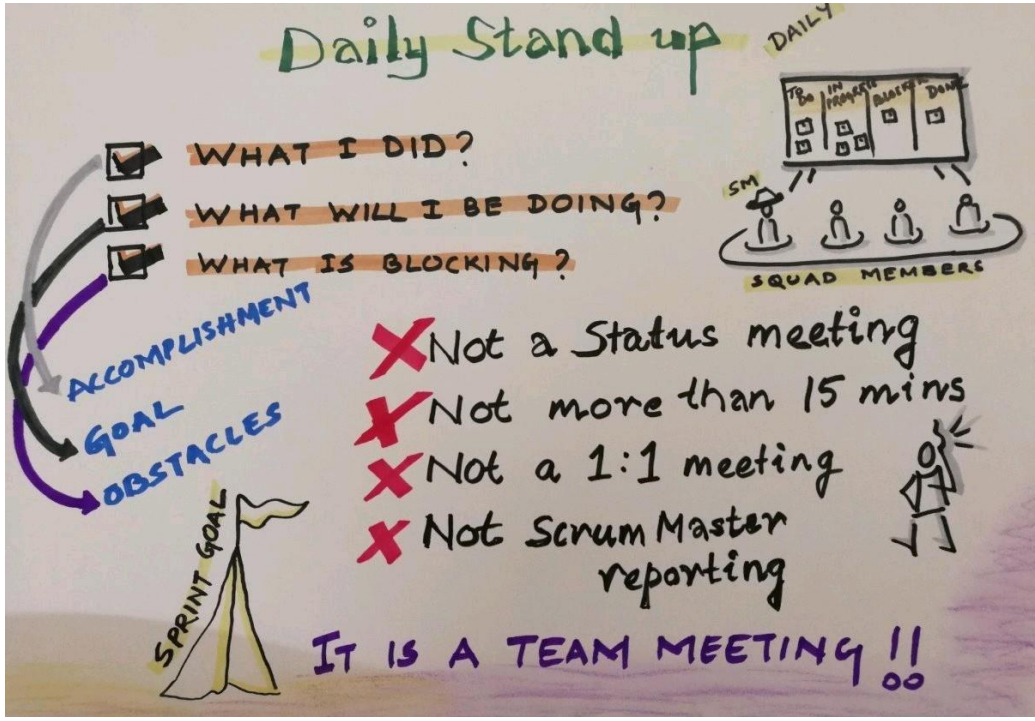


Sprint Planning Meeting



Daily Scrum

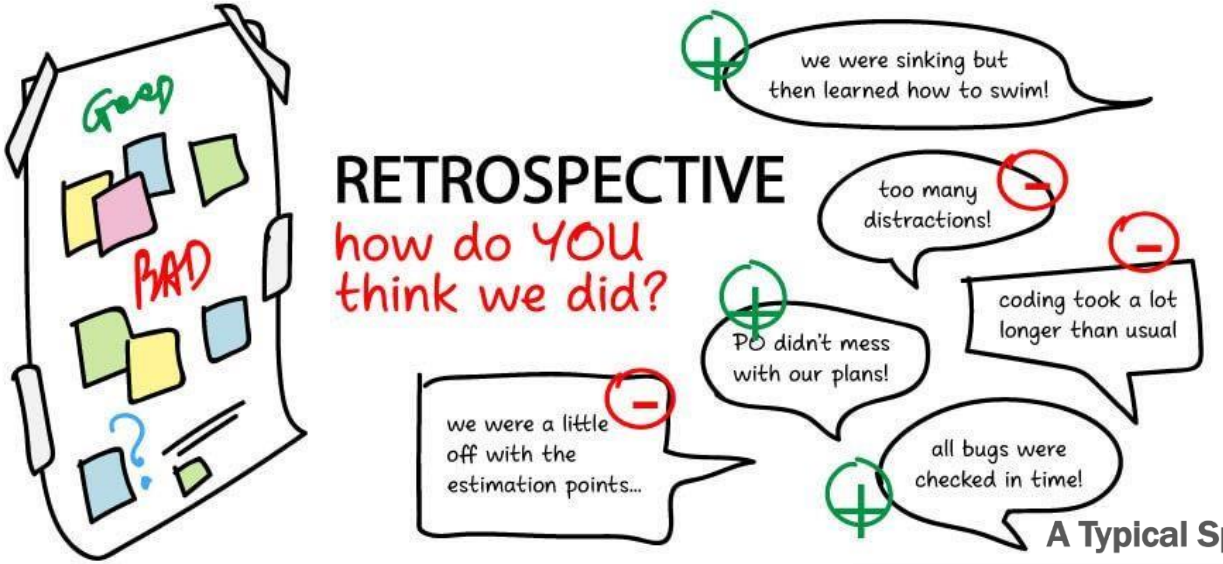
- Daily 15 Minutes Stand Up Meeting
  - What did you do yesterday to achieve the Sprint Goal ?
  - What will you do today to achieve the Sprint Goal ?
  - Is there any Impediments ?
- Not Problem Solving Meeting
- Not Status for the Scrum Master
- Only Team members can talks



Sprint Review Meeting

- 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

Sprint Retrospective



A Typical Sprint Retrospective Model

What worked well?

What could be improved?

What will we commit to doing in the next Sprint?

Scrum Team members make actionable commitments

QUICK TIPS TO ELEVATE YOUR NEXT

SPRINT RETROSPECTIVE

1. **Keep It Simple**

   Ask the team what they'd like to start, stop, and continue doing.

2. **Incorporate Novelty**

   Incorporate games & other varying tactics into your sprint retrospectives.

3. **Stay Focused**

   Build your retrospective using kanban boards that are democratically generated.

4. **Make It Action-Oriented**

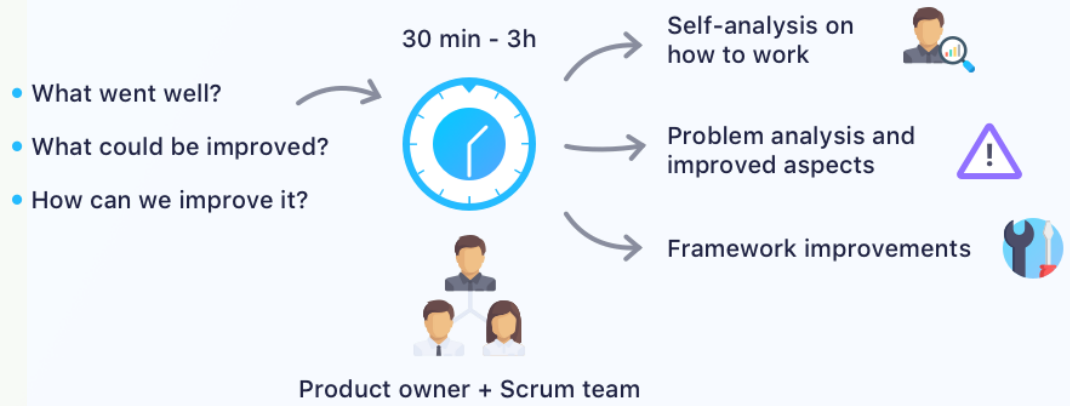
   Make sure you're assigning anything actionable to someone on the team.

5. **Bring In Outside Perspective**

   It could be wise to bring in an agile coach to help with retrospective facilitation.

Sprint Retrospective

Meeting after Sprint Review to review processes



Backlog Refinement

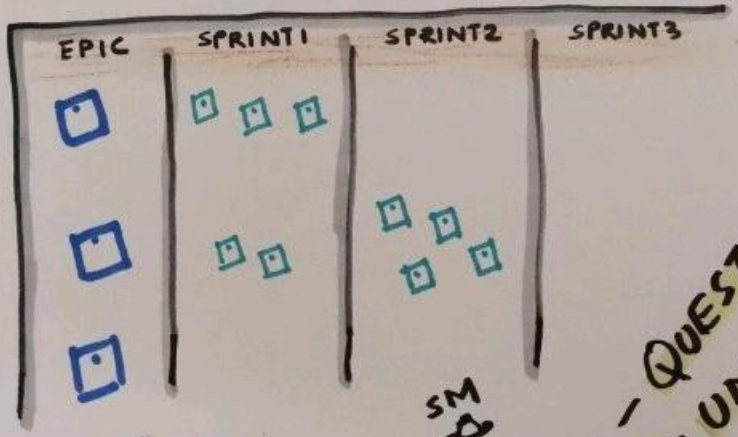
AT LEAST ONCE PER SPRINT !!

- IT IS A CONTINUOUS PROCESS !!
- WHY OVER HOW !!
- DRILL DOWN REQUIREMENTS !!
- IDENTIFY RISKS & DEPENDENCIES !!
- STORY SPLITTING & ESTIMATION !!

- ✗ No Defn. of. ready
- ✗ No 3C's
- ✓ No 'HOW' by PO
- ✓ No over-sized stories (SPLIT)

STAKEHOLDERS

ACCEPTANCE CRITERIA



PULL

SPRINT BACKLOG

PRODUCT BACKLOG

PRODUCT OWNER

CONTENT AUTHORITY & PRIORITIZATION

SM

SQUAD MEMBERS

- QUESTION
- UNDERSTAND
- SPLIT
- ESTIMATE
- IDENTIFY RISKS & DEPENDENCIES

Scrum Time Boxes

- Sprint Planning Meeting: This is time-boxed to eight hours for a one-month Sprint. For shorter Sprints, the event is proportionately shorter.
- Daily Scrum: The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.
- Sprint Review: This is a four-hour time-boxed meeting for one-month Sprints. Proportionately less time is allocated for shorter Sprints.
- Sprint Retrospective : This is a three-hour time-boxed meeting for one-month Sprints. Proportionately less time is allocated for shorter Sprints.

User Stories (INVEST)

Independent

- Avoid Introducing dependencies between stories

Negotiable

- Stories are negotiable. They are not written contracts or requirements

Valuable to users or customers

- The story must be valued by the users / Purchaser

Estimatable

- Developers should be able to estimate it.

Small

- Stories should be rightly sized , Too large or too small cannot be used in planning.

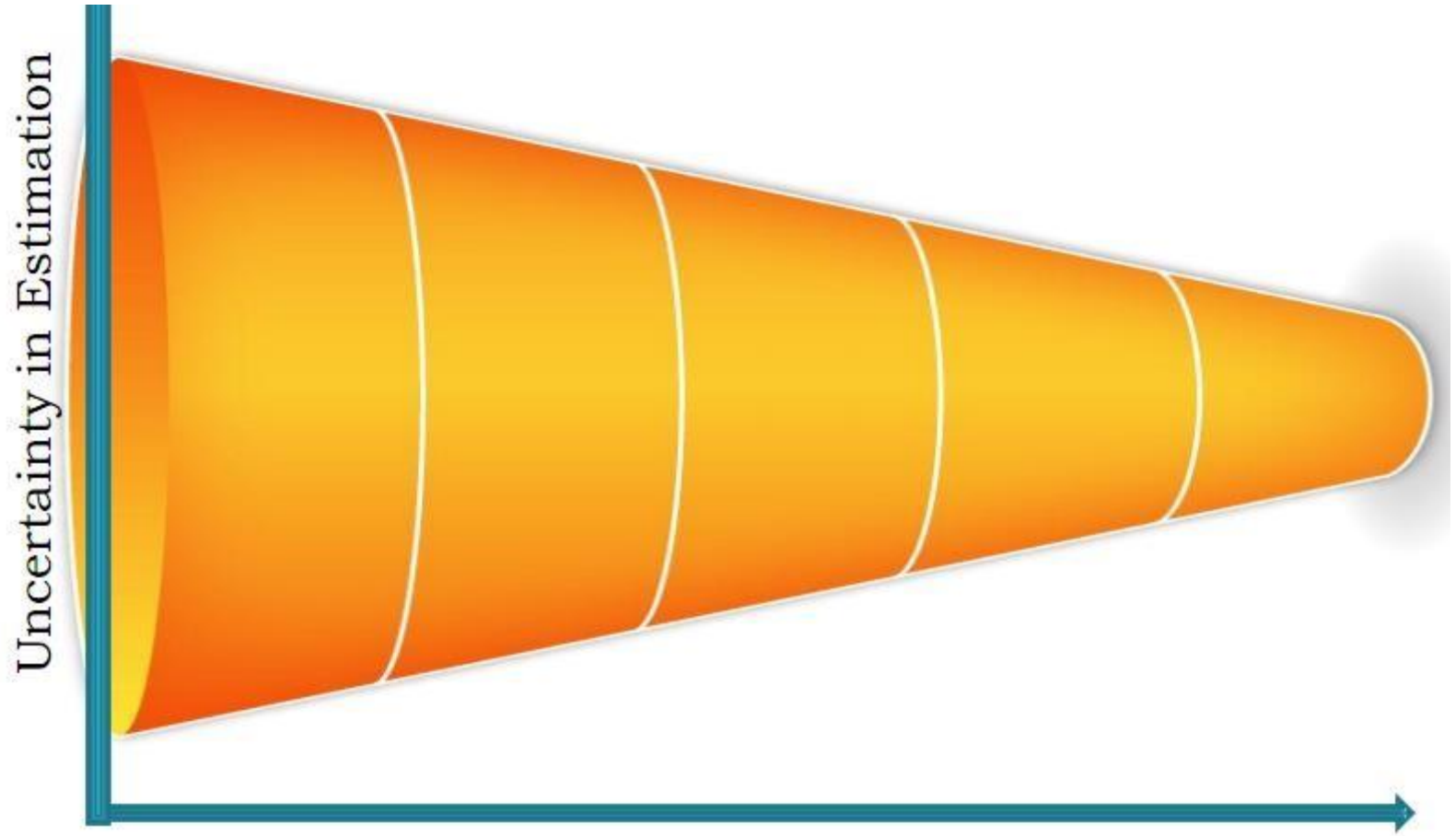
Testable

- Stories must be written so as to be testable. Successfully passing its tests proves that a story has been successfully developed.

Good User Story

- Closed Story
- Has User Role
- Written for one User
- Does not focus to much on the UI
- Has the acceptance tests

Estimation



What is User Stories ?

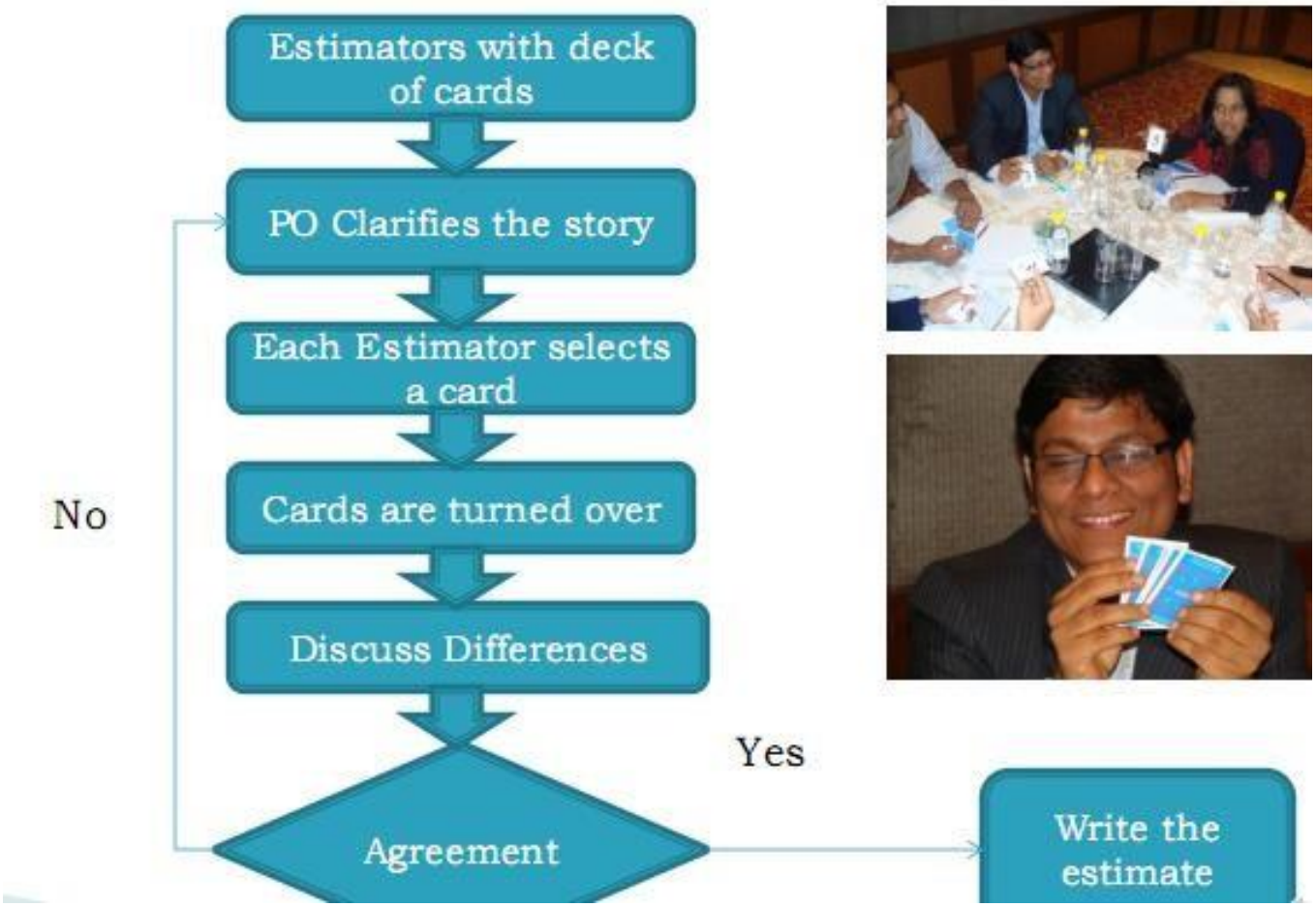
A User Story describe functionality that will be valuable to either a **user** or **purchaser** of a system.

As a
Who wants this piece of functionality
I want
What the user wants
So that
Why the user wants it

Story Point Estimate

- Estimate with in one order of magnitude
- Estimate scale should in scale
- Frequency used in scale
- Fibonacci series
  - 1, 2, 3, 5, 8, 13….

Planning Poker



Monitoring Release/Sprint

- Velocity
- Release Burn down Chart
- Burndown Bar Charts
- Velocity Chart
- Parking Lot Chart

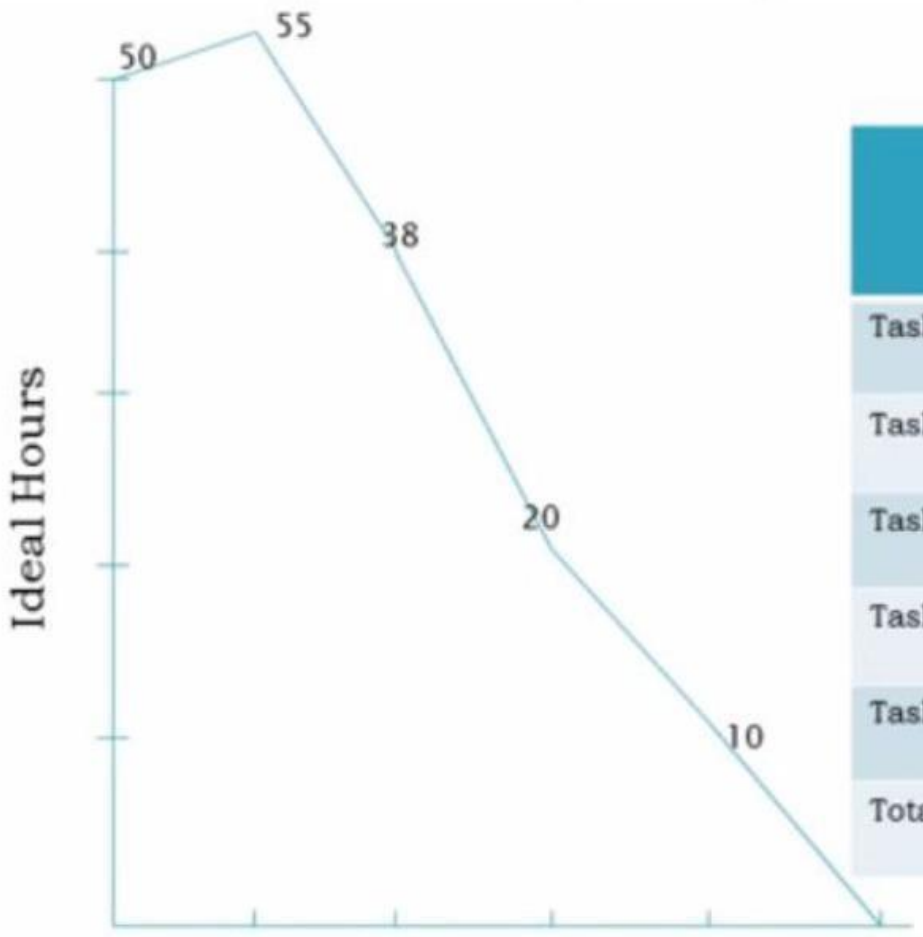
Velocity

Story	Story Point	Ideal Hours Planned	Hours Spent	Complete
Story A	3	33	35	Done
Story B	2	18	15	Done
Story C	1	10	12	Done
Story D	5	57	65	50%

What would be velocity? 6 or 8.5 or 11

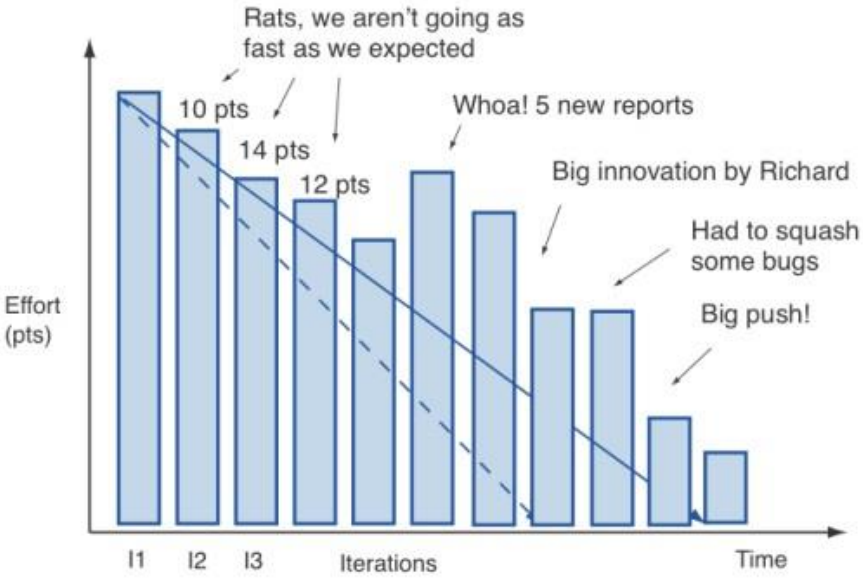
Release Planning – User Story Mapping

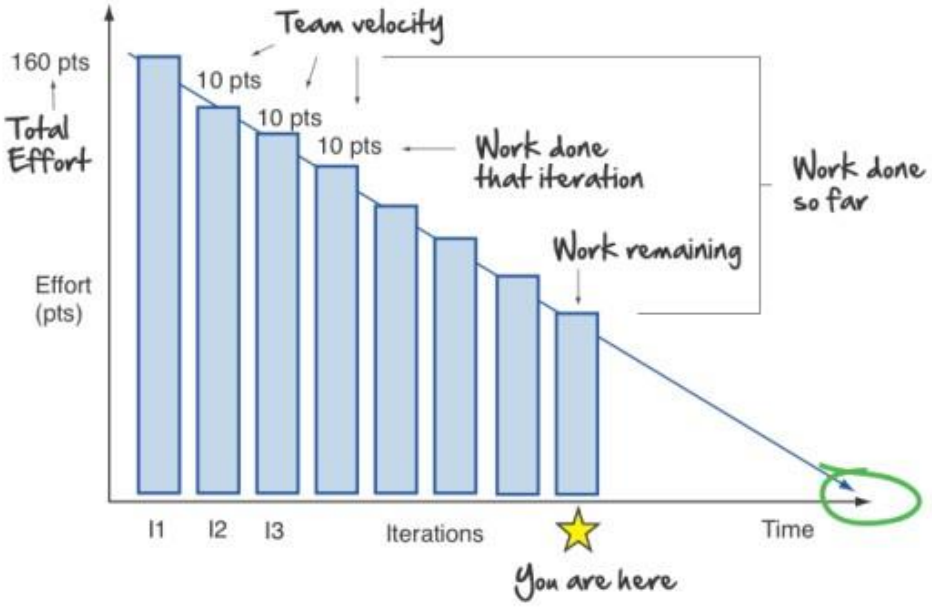
Sprint Burndown Chart



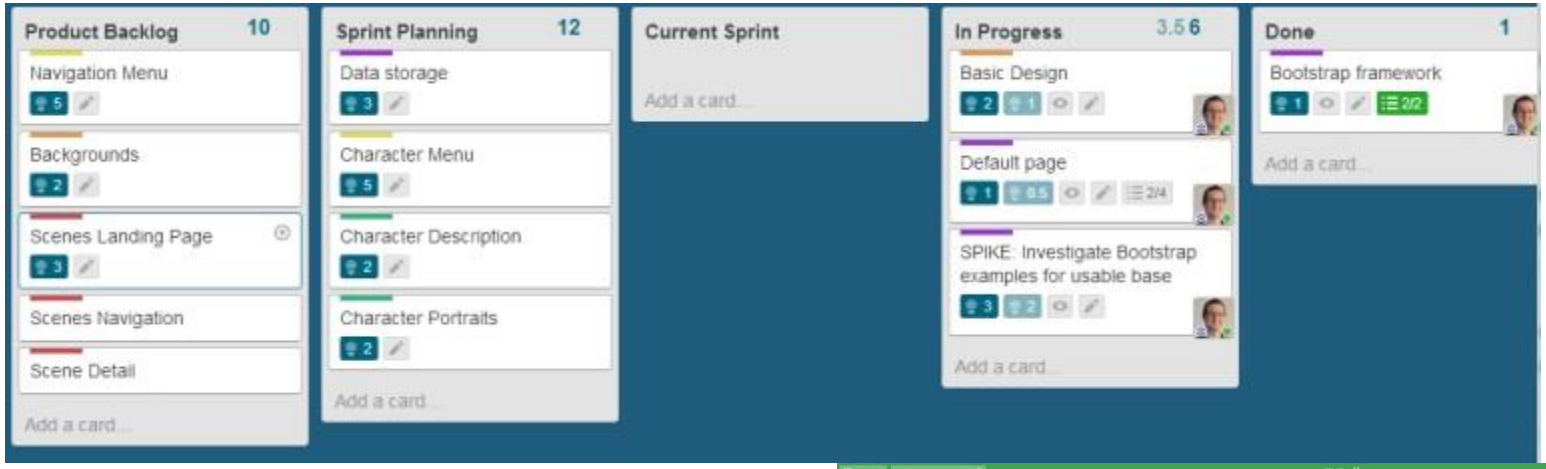
	#0	#1	#2	#3	#4	#5
Task 1	10	15	5	0	0	0
Task 2	8	6	5	0	0	0
Task 3	12	14	10	8	5	0
Task 4	12	12	10	8	5	0
Task 5	8	8	8	4	0	0
Total	50	55	38	20	10	0

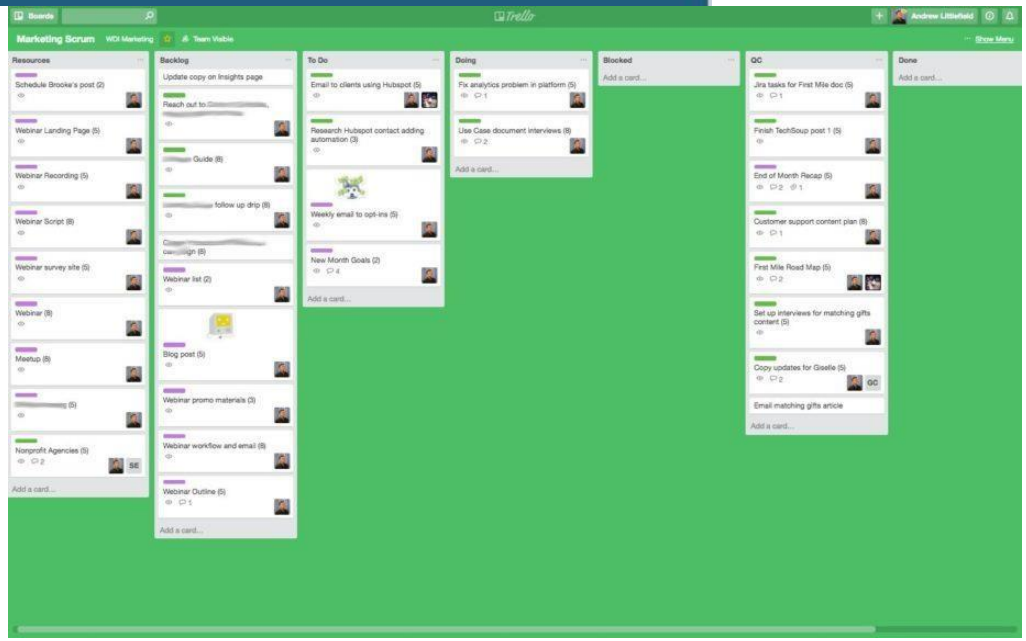
Sprint Burndown Chart





Scrum Board & Trello



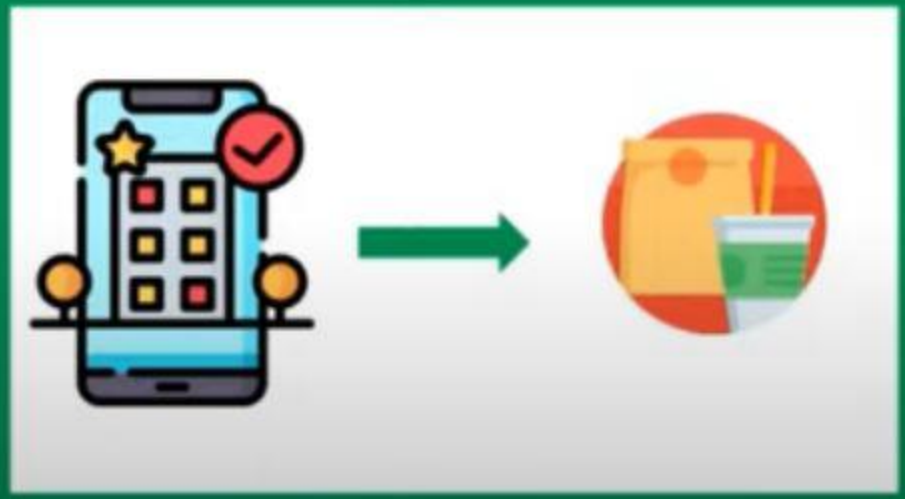


Agile implementation - Software Scenario

CASE STUDY

Restaurant called "**ABC Cuisines**" is looking for an **Online Home Delivery Service** for her Restaurant.

Raj is the Business Analyst part of the Consulting firm who is hired to set this up. He is going to use the **User Story Mapping Technique** to gather Business Requirements and for release prioritisation.



Agile implementation - Software Scenario