



# Software Project Management



# Agile is a Team Effort

All members of the team can view and use boards.

SCRUM board focuses on completing deliverables.

KANBAN board is great for teams whose main objective is managing a continuous flow of work, like a support team

# JIRA

- ▶ Review the two JIRA Software Boards
- ▶ SCRUM BOARDS
- ▶ KANBAN BOARDS

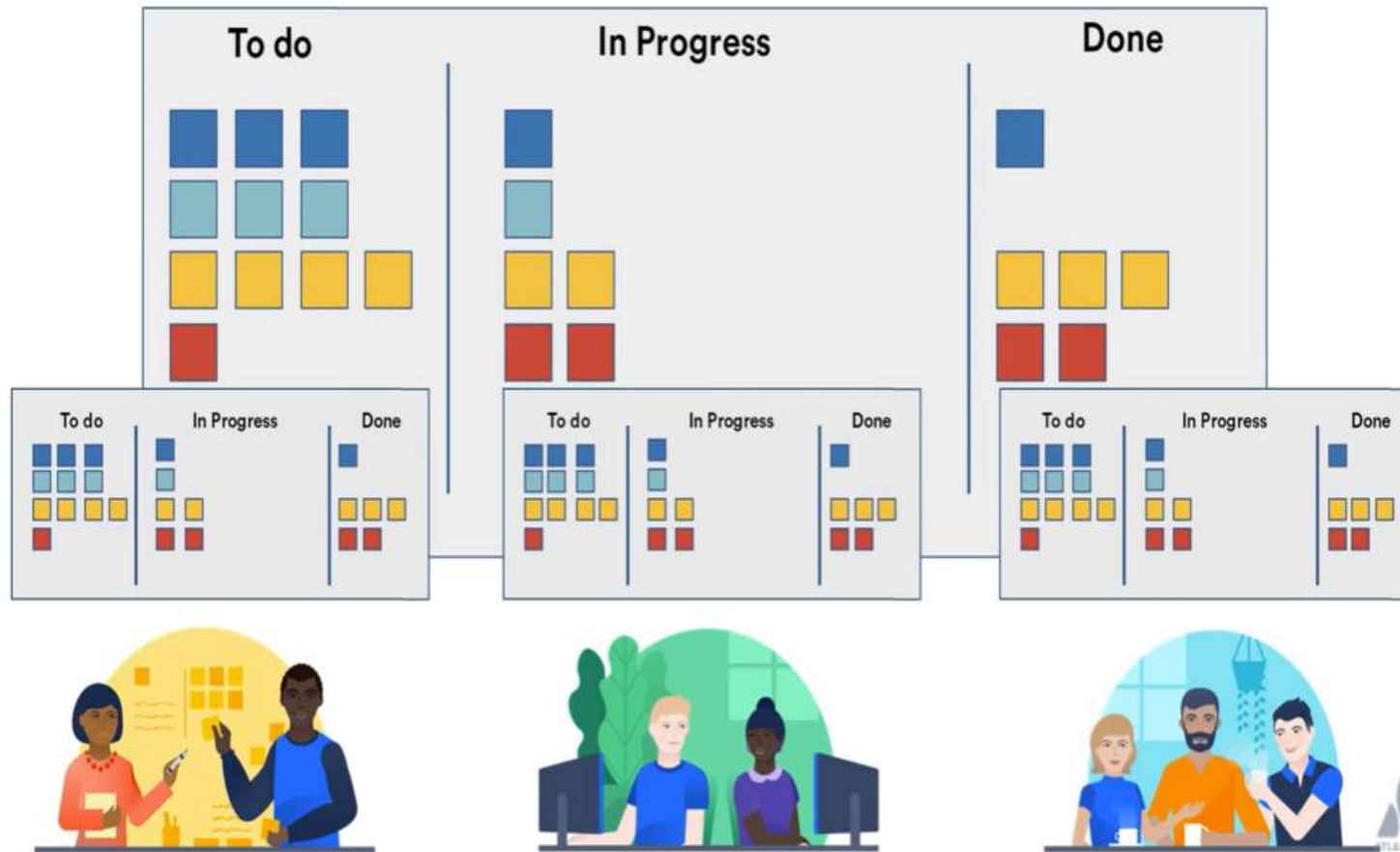


# Introduction

Teams spread across the world



# Introduction



- Each team has its own version of poster boards and can attach notes
- After few days, they will have video conference calls

# JIRA Software

The Jira board displays the following issues:

Column	Issue ID	Title	Status	Assignee
TO DO	TIS-130	Planet images are fuzzy	Not Started	[Assignee]
	TIS-136	Update planets available depending on galaxy selected	Not Started	[Assignee]
	TIS-145	Record audio for Identify Planet button using Spock, Worf, Dalek, C-3PO and Terminator voices	Not Started	[Assignee]
IN PROGRESS	TIS-135	Create 3D imagery of each planet	In Progress	[Assignee]
	TIS-139	Write test script	In Progress	[Assignee]
	TIS-132	TARDIS function is not correctly showing planets in future/past locations	In Progress	[Assignee]
CODE REVIEW	TIS-133	Create main Locate a Planet screen	Code Review	[Assignee]
	TIS-137	Implement red and green night mode filters	Code Review	[Assignee]
	TIS-131	Remove planet Alderaan from available planets as it no longer exists	Code Review	[Assignee]
DONE	TIS-129	Find button is not identifying correct planets in galactic quadrant	Done	[Assignee]
	TIS-134	Create compass to point to target planet	Done	[Assignee]
	TIS-138	Update Design Specs	Done	[Assignee]

A board displays Jira issues. Issues are any type of work item.





# JIRA Software

The screenshot displays a JIRA Kanban board for the 'Planet Locator App' project. The board is organized into four columns: TO DO, IN PROGRESS, CODE REVIEW, and DONE. Each column contains several task cards, each with a title, a status bar, and a priority icon. Handwritten sticky notes are attached to some of the task cards.

**TO DO Column:**

- Planet images are fuzzy (Planet Locator App, TIS-130)
- Update planet data depending on user input (Planet Locator App, TIS-136)
- Record audio for Identify Planet button using Spock, Worf, Dalek, C-3PO and Terminator voices (Planet Locator App, TIS-145)

**IN PROGRESS Column:**

- Create 3D imagery of each planet (Planet Locator App, TIS-135)
- Write test script (Planet Locator App, TIS-132)
- TARDIS function is not correctly showing planets in future/past locations (Planet Locator App, TIS-132)

**CODE REVIEW Column:**

- Create main Locate a Planet screen (Planet Locator App, TIS-133)
- Implement red and green night mode filters (Planet Locator App, TIS-137)
- Remove planet Alderaan from available planets as it no longer exists (Planet Locator App, TIS-131)

**DONE Column:**

- Find button is not identifying correct planets in galactic quadrant (Planet Locator App, TIS-129)
- Update Design Specs (Planet Locator App, TIS-138)

**Handwritten Sticky Notes:**

- As a marketing Analyst, I would like to click through reports (attached to TIS-136)
- As a developer, I want more information about usage (attached to TIS-132)
- As a user, I want to know my travel balance (attached to TIS-129)

# JIRA BOARDS

## Scrum Boards

Great for teams that are completing a deliverable

Fixed-length development timeframe

Frequent check-ins, communications and ceremonies

## Kanban Boards

Great for managing a continuous flow of work

Less formal structure than Scrum

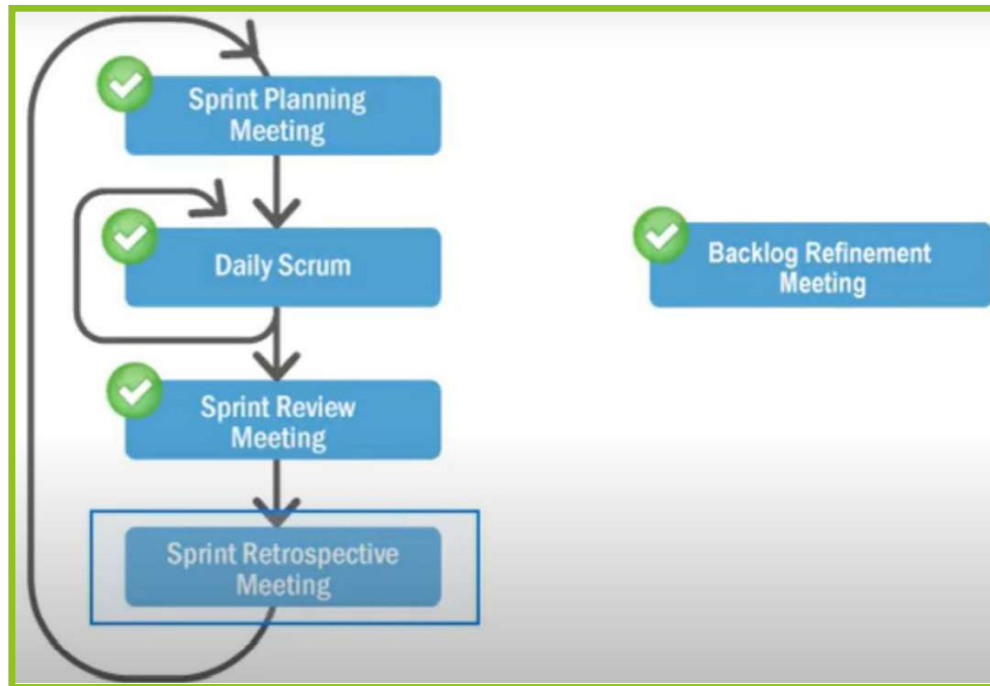




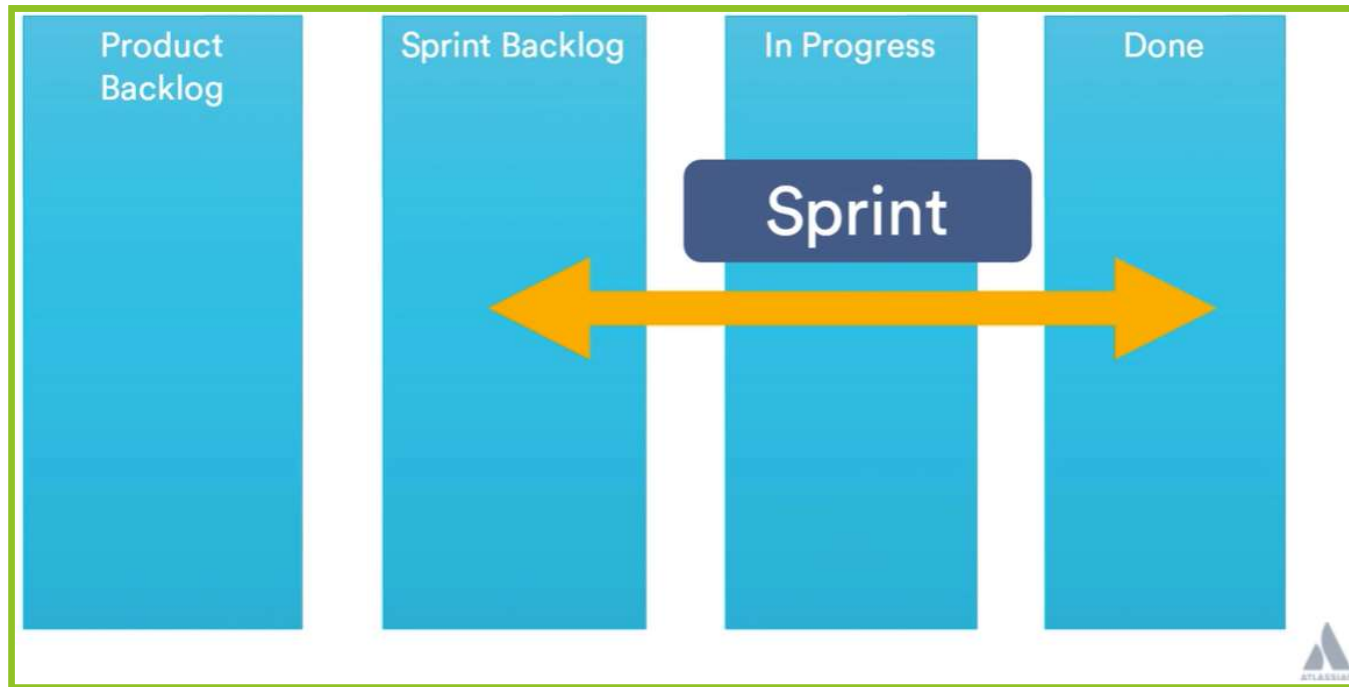
# JIRA ROLES



# Sequence of Scrum Ceremonies

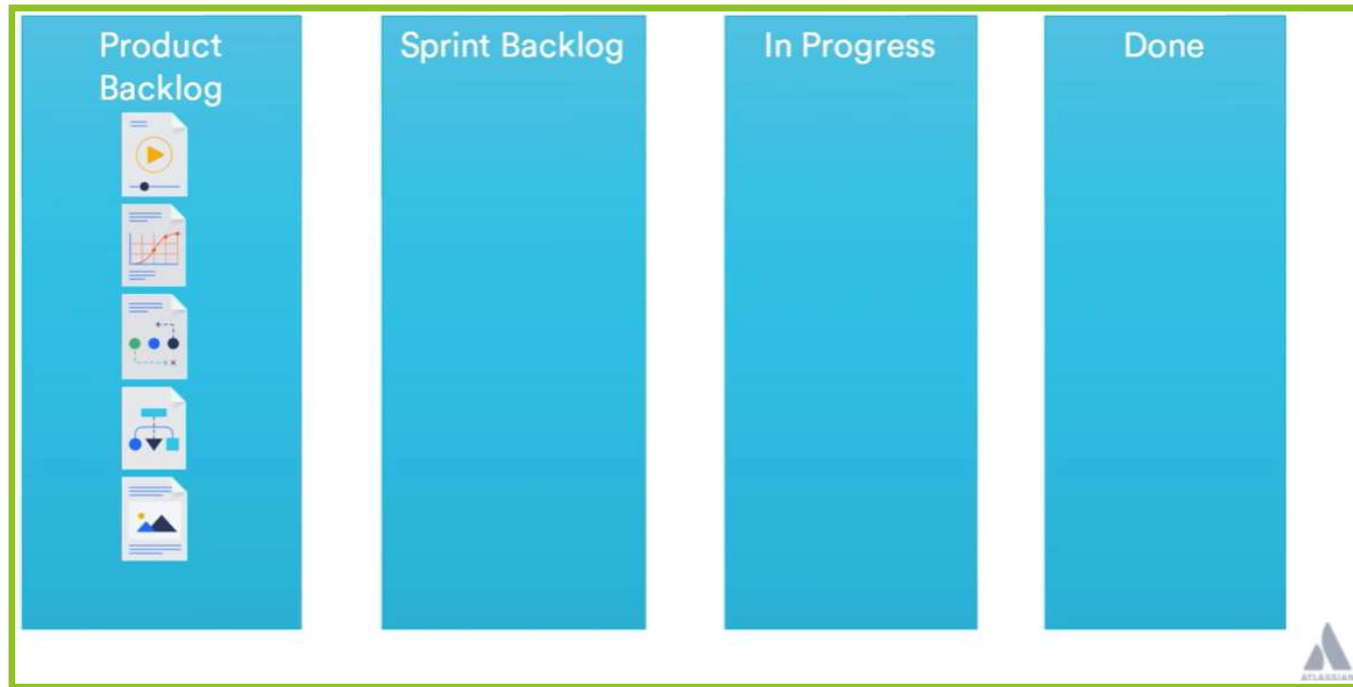


# SCRUM BOARD



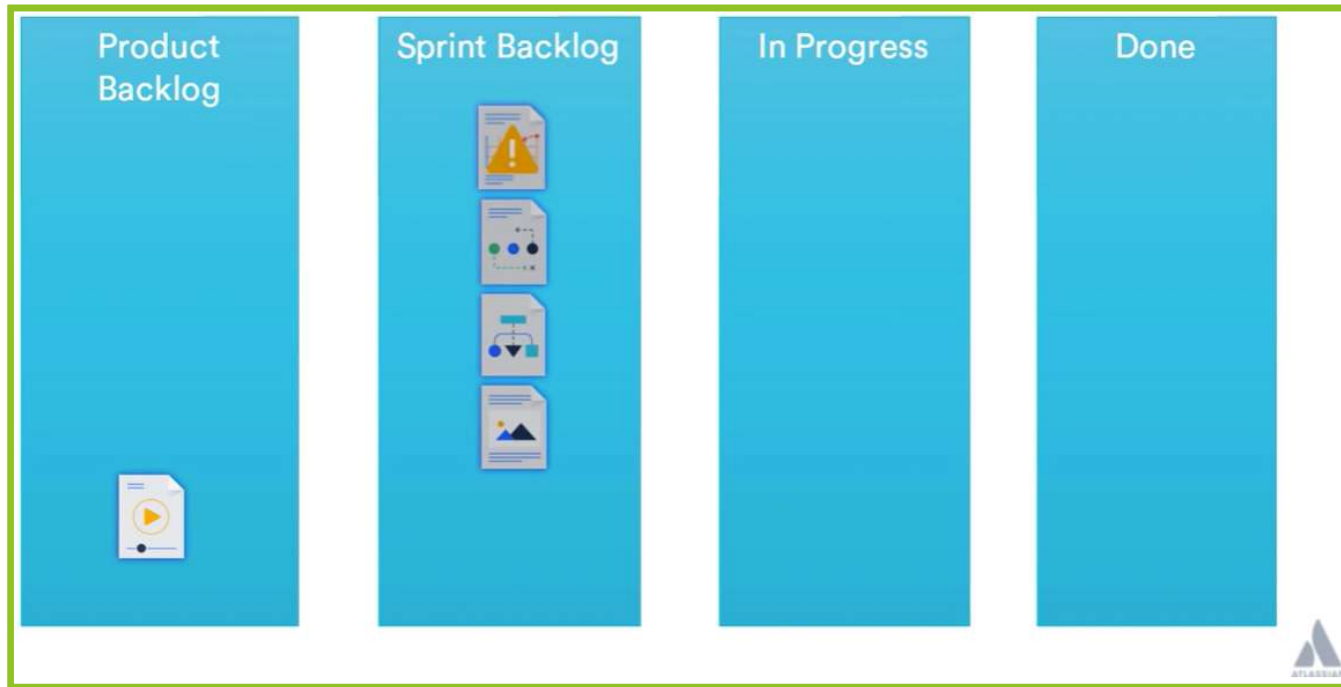
- SCRUM Board focuses on planning, committing and delivering bodies of work over fixed length period of time called Sprint
- Team decides how long sprint lasts
- It's a team effort

# SCRUM BOARD



- Product owners defines, add and prioritize work items/issues/features (giant to do list)
- Urgent items appear at top
- Team grooms the product backlog and developing finer details for the highest priority work item and estimating how long each item might take to complete

# SCRUM BOARD



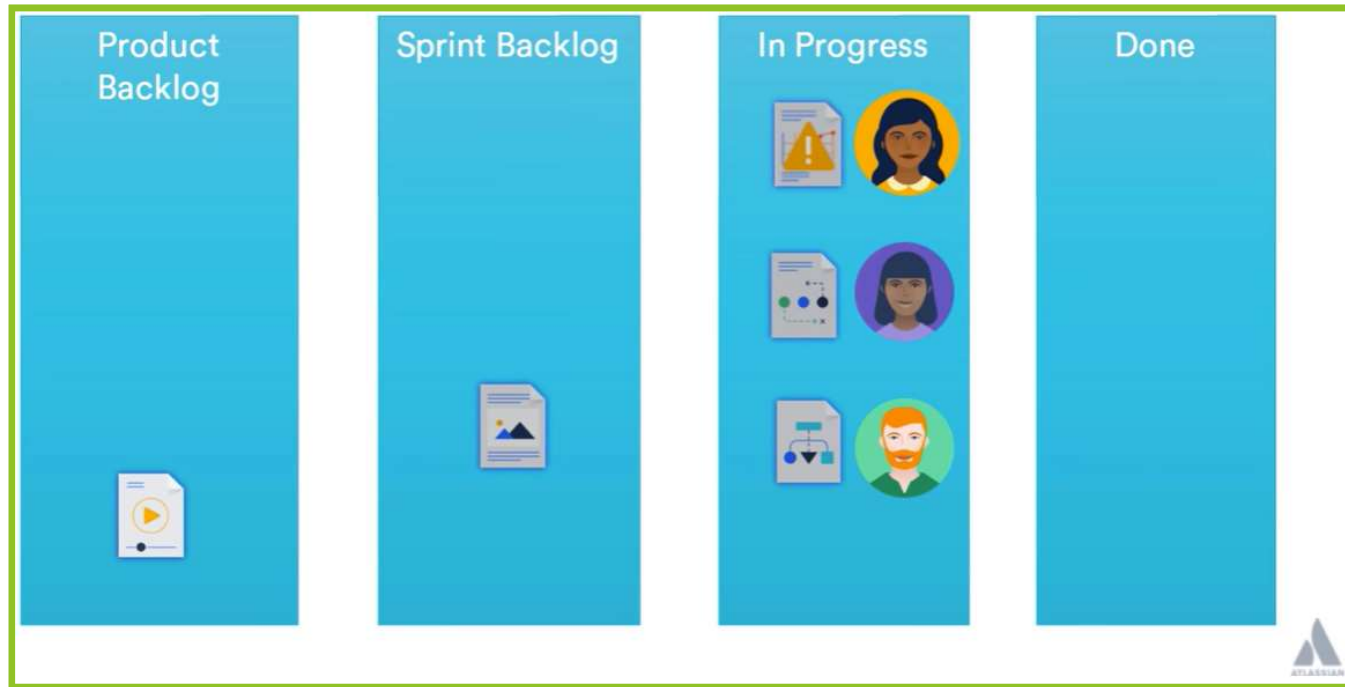
- Developers put the items to sprint backlog

# Sprint Planning

- ▶ **Attendees:** Development team, scrum master, product owner
- ▶ **When:** At the beginning of a sprint.
- ▶ **Duration:** Usually up to two hours per week of iteration. e.g. a two-week sprint kicks off with a four-hour planning meeting.
- ▶ **Purpose:** Sprint planning sets up the entire team for success throughout the sprint. Coming into the meeting, the product owner will have a prioritized product backlog. They discuss each item with the development team, and the group collectively estimates the effort involved. The development team will then make a sprint forecast outlining how much work the team can complete from the product backlog. That body of work then becomes the sprint backlog.



# SCRUM BOARD

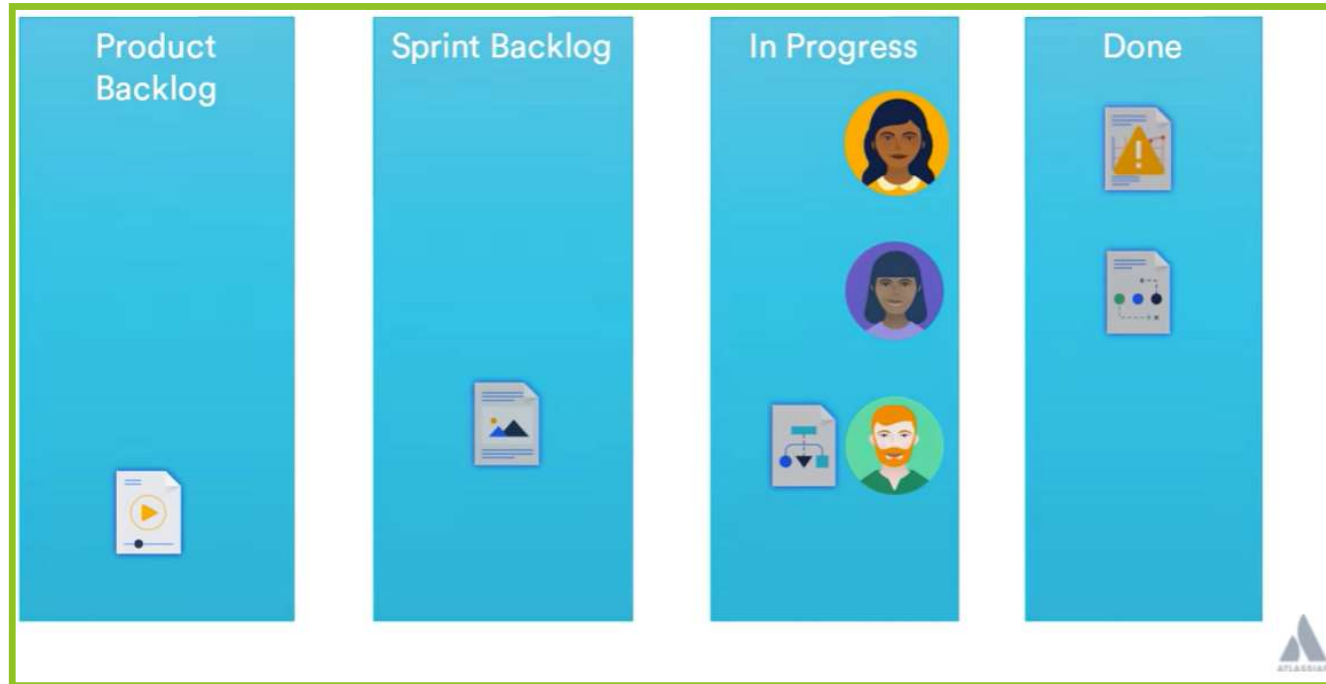


Team starts working on the sprint backlog

# Daily Stand-up

- ▶ **Attendees:** Development team, scrum master, product owner
- ▶ **When:** Once per day, typically in the morning.
- ▶ **Duration:** No more than 15 minutes. Don't book a conference room and conduct the stand-up sitting down. Standing up helps keep the meeting short!
- ▶ **Purpose:** Stand-up is designed to quickly inform everyone of what's going on across the team. It's not a detailed status meeting. The tone should be light and fun, but informative. Have each team member answers the following questions:
  - ▶ What did I complete yesterday?
  - ▶ What will I work on today?
  - ▶ Am I blocked by anything?
  - ▶ There's implicit accountability in reporting what work you completed yesterday in front of your peers. No one wants to be the team member who is constantly doing the same thing and not making progress.

# SCRUM BOARD



- Team holds sprint review meeting to assess the work completed

# Sprint Review

- ▶ **Required Attendees:** Development team, scrum master, product owner
- ▶ **Optional Attendees:** Project stakeholders
- ▶ **When:** At the end of a sprint or milestone.
- ▶ **Duration:** Typically 60 minutes per week of iteration-e.g. a two-hour review following a two-week sprint.
- ▶ **Purpose:** Iteration review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

# SCRUM BOARD



- Team holds **Retrospective Meeting** to assess the work completed
- Reflect findings such as incorrect estimations or other inefficiencies before starting another product iteration in the next sprint

# Retrospective

- ▶ **Attendees:** Development team, scrum master, product owner
- ▶ **When:** At the end of an iteration.
- ▶ **Duration:** Typically 45 minutes per week of iteration-e.g. a 90-minute retrospective after a two-week sprint.
- ▶ **Purpose:** Agile is about getting rapid feedback to make the product and development culture better. Retrospectives help the team understand what worked well-and what didn't. Retrospectives aren't just a time for complaints without action. Use retrospectives to find out what's working so the team can continue to focus on those areas. Also, find out what's not working and use the time to find creative solutions and develop an action plan. Continuous improvement is what sustains and drives development within an agile team, and retrospectives are a key part of that.



# Backlog Grooming (Refinement) Meeting

- ▶ **Attendees:** Development team, scrum master, product owner
- ▶ **When:** Mid/end of an iteration.
- ▶ **Duration:** Typically 30-60 minutes
- ▶ **Purpose:**
  - ▶ Break down large user stories into smaller tasks.
  - ▶ Discuss user stories with the team, answer any related questions to smooth out any ambiguity.
  - ▶ Ensure upcoming user stories meet the team's "definition of ready" by adding key contextual information and acceptance criteria.

# Benefits of using Story Points

- ▶ No correlation with skills and experience of the estimator
- ▶ Estimates independent of assignee
- ▶ High-level estimation
- ▶ Velocity is Tracked

# Sprint Planning using 'Planning Poker'

- ▶ Planning Poker is an agile estimating and planning technique that is consensus based.
- ▶ To start a poker planning session, the product owner or customer reads an agile user story or describes a feature to the estimators.
- ▶ Each estimator is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100, which is the sequence we recommend.
- ▶ The values represent the number of story points, ideal days, or other units in which the team estimates.
- ▶ The estimators discuss the feature, asking questions of the product owner as needed.
- ▶ When the feature has been fully discussed, each estimator privately selects one card to represent his or her estimate.
- ▶ All cards are then revealed at the same time.
- ▶ If all estimators selected the same value, that becomes the estimate, If not, the estimators discuss their estimates.
- ▶ The high and low estimators should especially share their reasons.
- ▶ After further discussion, each estimator reselects an estimate card, and all cards are again revealed at the same time.
- ▶ The poker planning process is repeated until consensus is achieved or until the estimators decide that agile estimating and planning of a particular item needs to be deferred until additional information can be acquired

# Planning Poker

- ▶ Suppose:
  - ▶ We have a product backlog with 12 stories
  - ▶ 3 stories are not groomed and need estimation
  - ▶ Let's follow 5 step workflow process like below



## Product Backlog

Backlog	Story 13
	Story 12
	Story 11
Groomed	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]
	Story 2 [3]
	Story 3 [2]

3 Stories in Backlog stage, will groom 2 stories today

Three Stories are in Groomed state , and ready to plan for future sprint, having estimated story point as 3, 5 and 3

2 Stories are in development with estimated story point as 8 and 5

1 Story is completed by developers, with estimated story point as 2, and ready to review and accept by product owner

3 stories are in Accepted state, with estimated story point as 5, 3 and 2

OUR FOCUS: To estimate its story points using planning poker

## Grooming Story 11

Developers and testers, mutually discussing

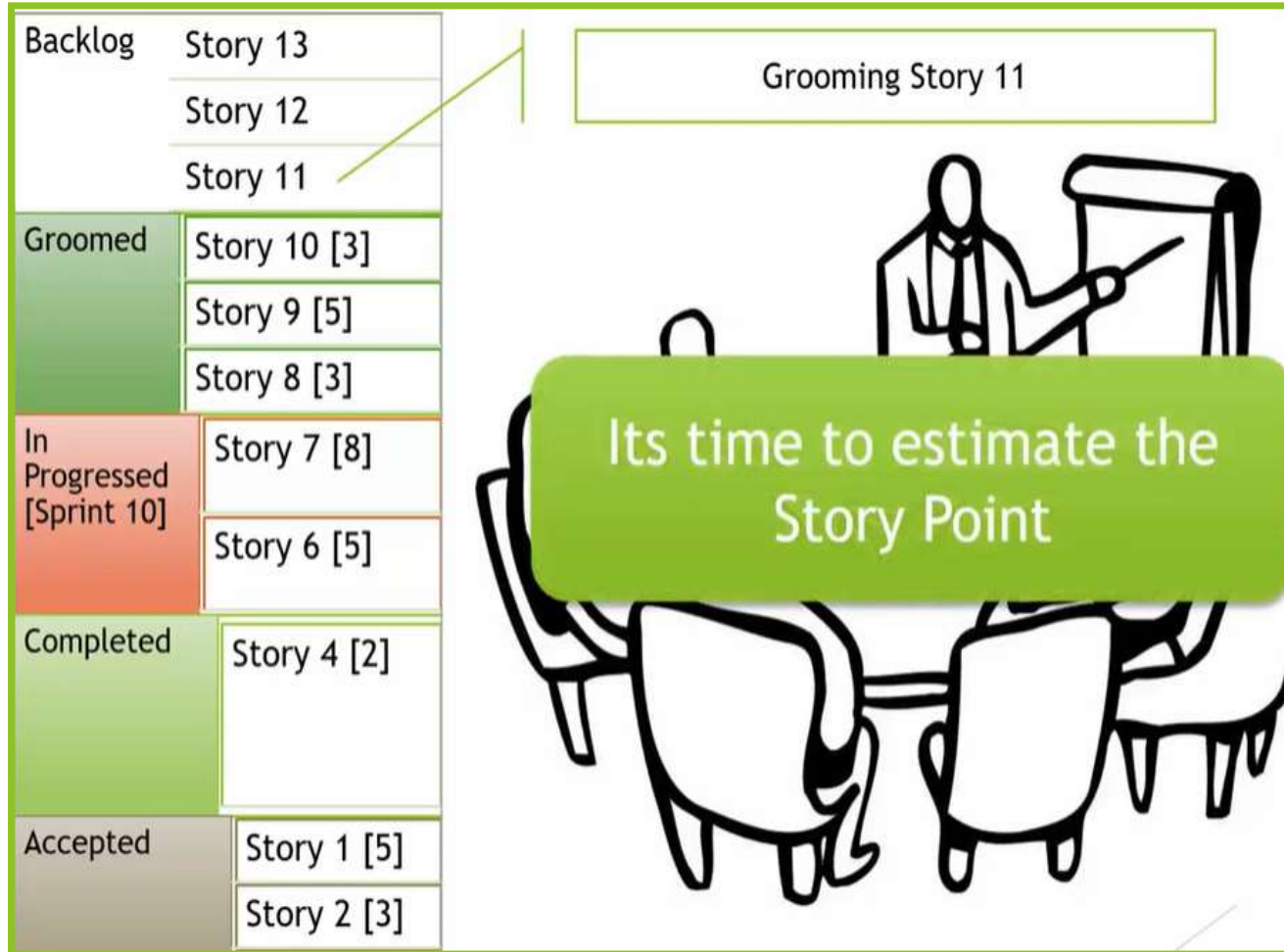
1. Amount of Work
2. Associated Risks
3. Technical changes
4. Dependencies
5. Complexity
6. and the value

Product Owner is explaining all the aspects of the story requirement, dependencies, acceptance criteria and business value

Scrum Master facilitating the grooming

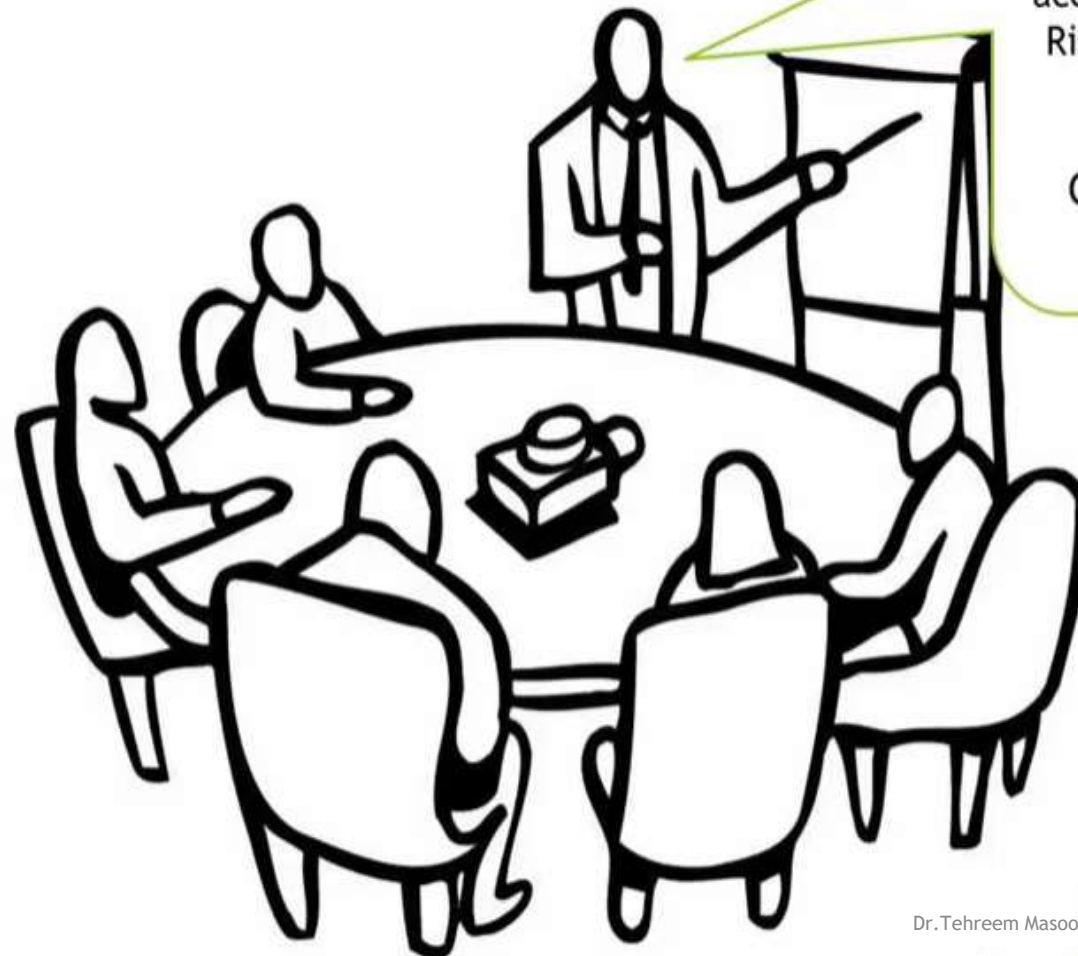






Backlog	Story 13
	Story 12
	Story 11
Groomed	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]
	Story 2 [3]

Grooming Story 11



Guys, Are you clear about the requirement and acceptance criteria. And its Risks dependencies or any other Impact

Can We estimate the Story

Backlog	Story 13
	Story 12
	Story 11
Groomed	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]
	Story 2 [3]

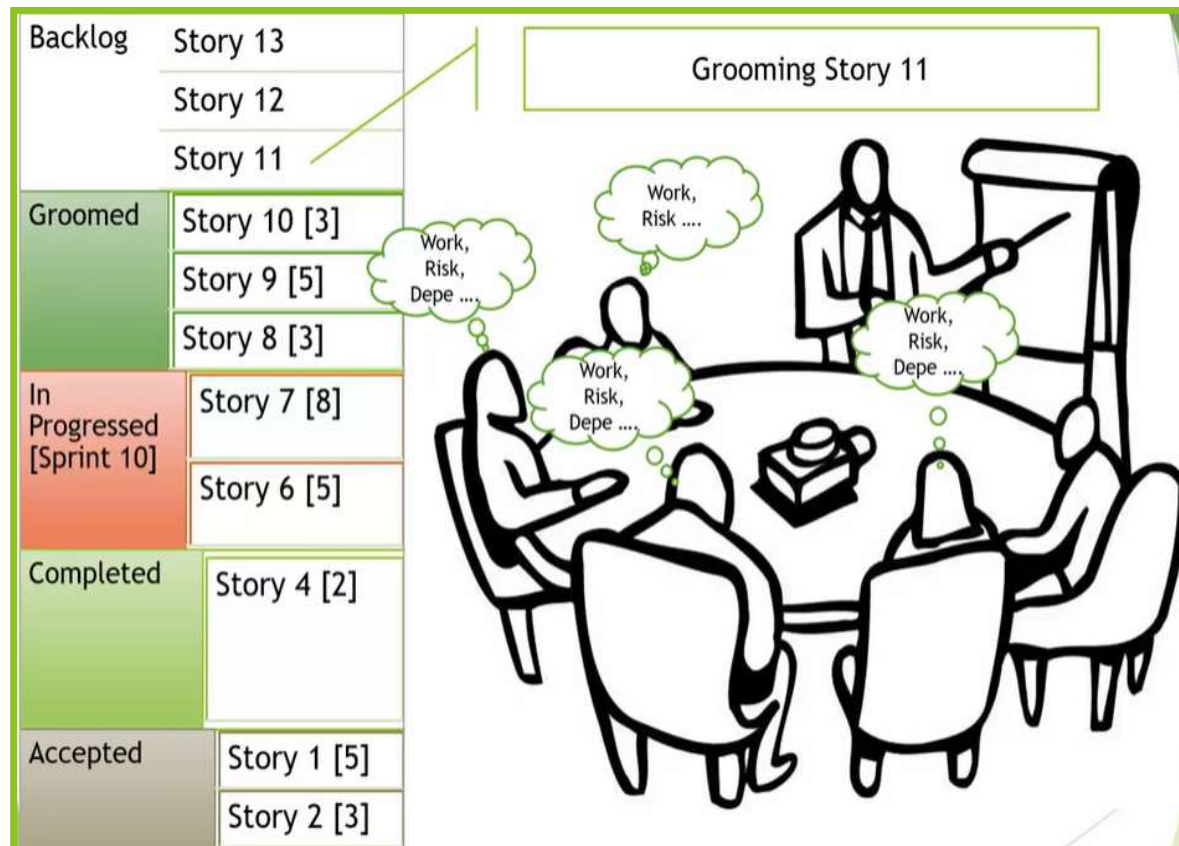
Grooming Story 11

Guys, Are you clear about the requirement and acceptance criteria. And its Risks dependencies or any other Impact

Can We estimate the Story

Remember, Only Development team can estimate by expressing their view points with a Number in Fibonacci Series

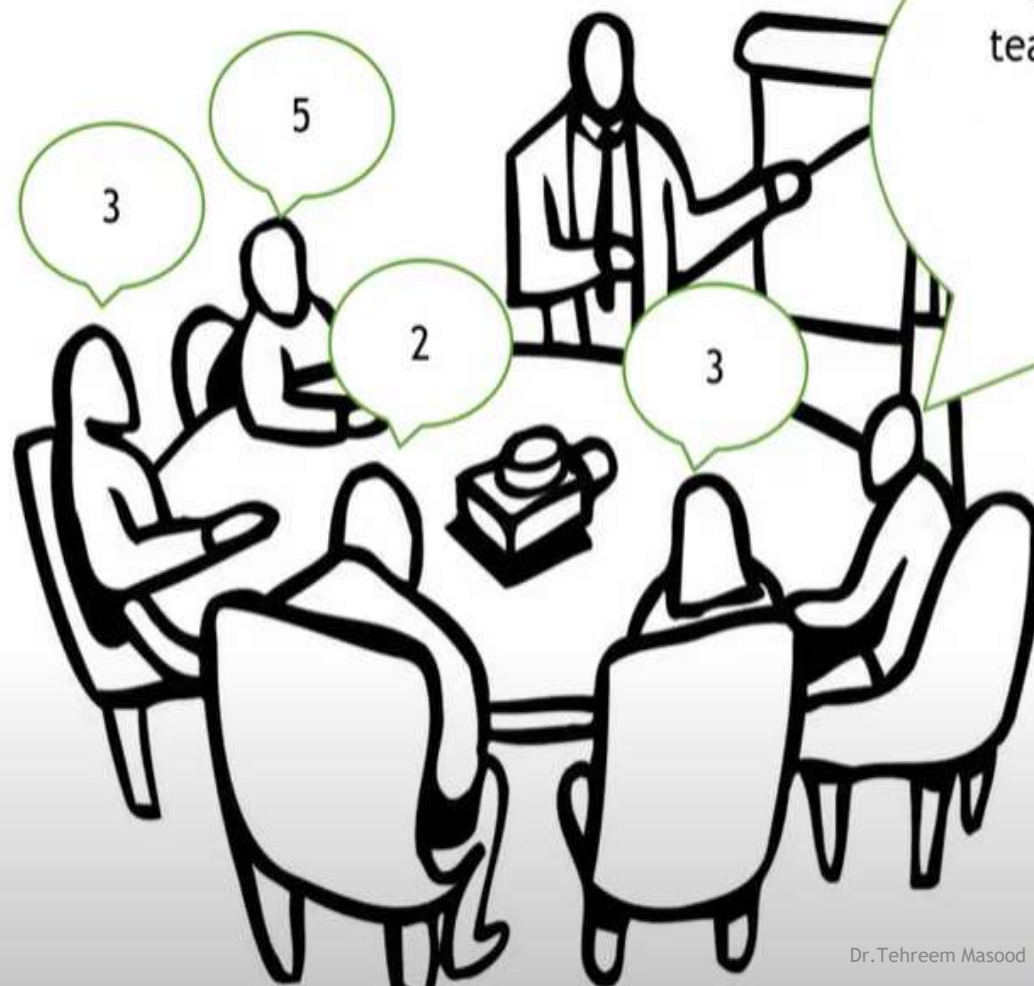
They can discuss details with PO, BA or external team member to gather more information to gauge the size better





Backlog	Story 13
	Story 12
	Story 11
Groomed	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]

Grooming Story 11



Great, as we can see the majority of the team member voted for Story Point as 3  
Let's Mark this story estimation as

3

Backlog	Story 13
	Story 12
Groomed	Story 11 [3]
	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]
	Story 2 [3]
	Story 3 [2]

Now We have one story Groomed and estimated as Story Point **3**, and its Ready for Planning



Backlog	Story 13
Groomed	Story 12 [2]
	Story 11 [3]
	Story 10 [3]
	Story 9 [5]
	Story 8 [3]
In Progressed [Sprint 10]	Story 7 [8]
	Story 6 [5]
Completed	Story 4 [2]
Accepted	Story 1 [5]
	Story 2 [3]
	Story 3 [2]

Now We have one story Groomed and estimated as Story Point **3**, and its Ready for Planning

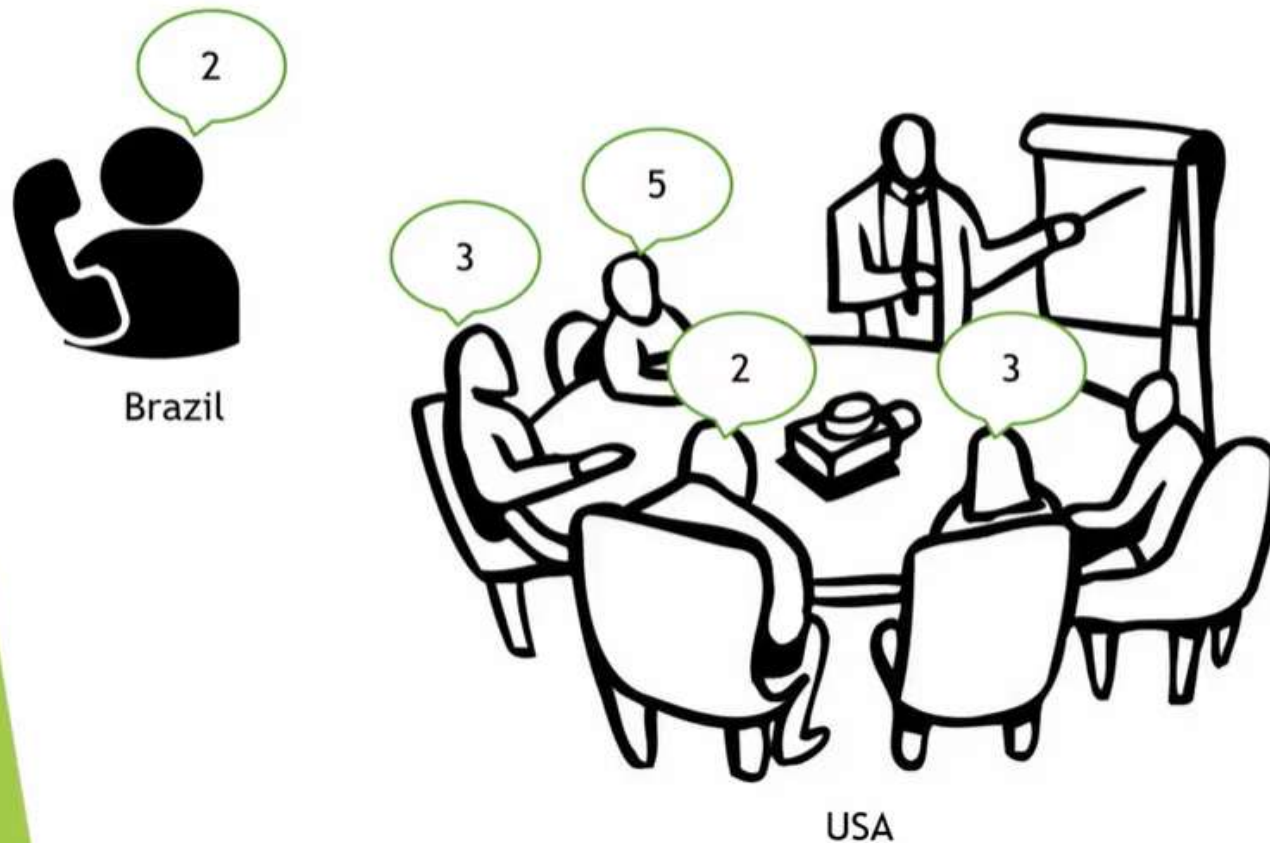
With the same technique we groomed the story 12 and estimated **2** Story Points for it.

The Team was unable to estimate the User Story 13, Because there are some technical doubts, team wants to clarify with External Team, And will estimate on next grooming session.

Scrum Master will organize a meeting with required persons to clarify all doubts

# Planning poker with distributed Agile

We can estimate user story in a same technique, even when our team is distributed in different geographic location. Only difference is they are connected on telephone and live screen sharing like WebEx, blue jeans etc.

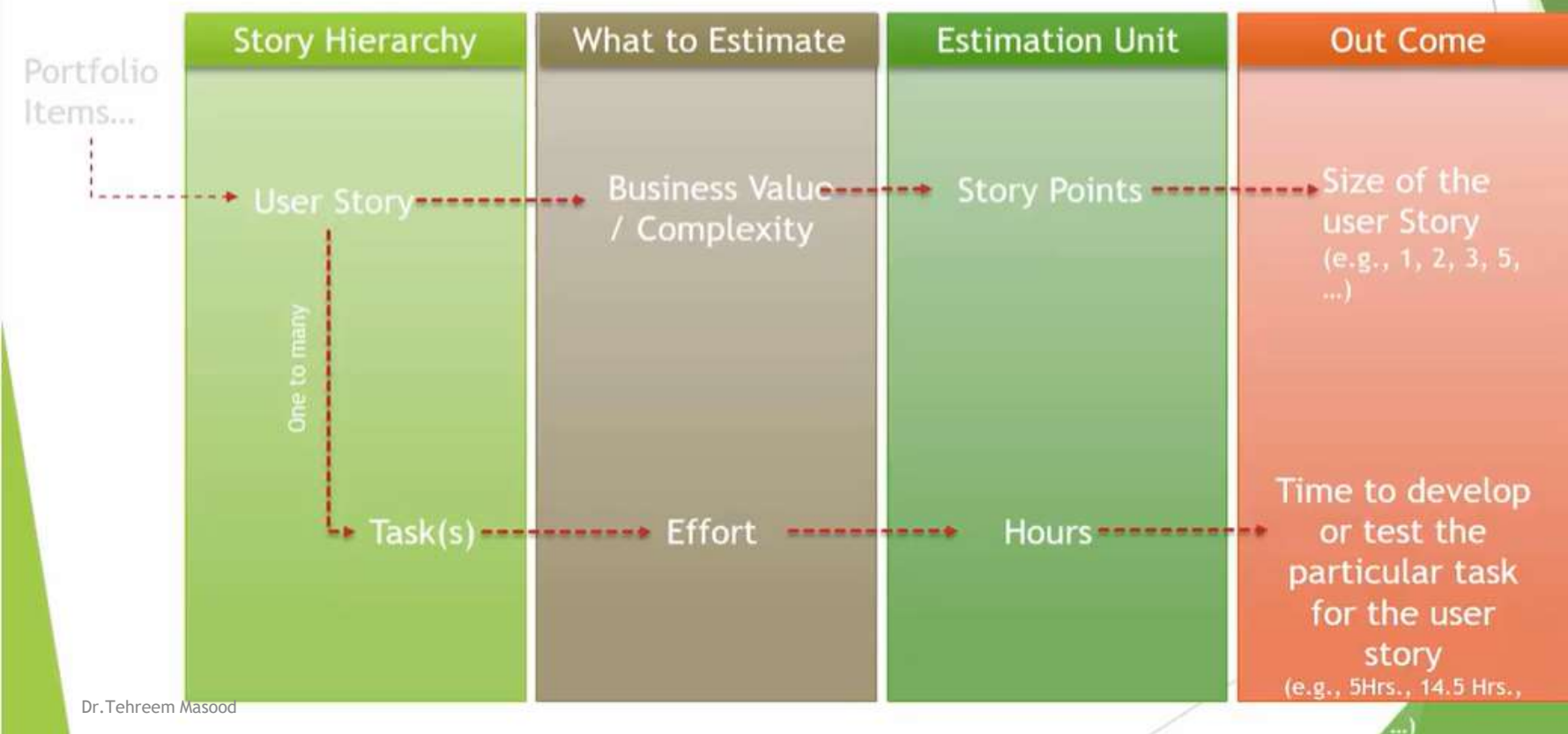


# Story point estimation vs effort estimation

Remember We don't and should not relate Story Points and Efforts.

Story Point is to estimate the Story Size, Value, Complexity, Risk, Dependencies etc

Where Tasks are the child's of user story is to estimate its efforts in Hour, for better planning and capacity Mapping



# Agile Reports - Burn Down Chart

- Burndown chart maps the remaining amount of work left against a timeline. At one glance, an agile burndown chart shows you how much work you've completed, how much work you have left, the ideal rate of completion, and a forecast of how much you need to complete each day if you are to complete your goals, given your current progress.



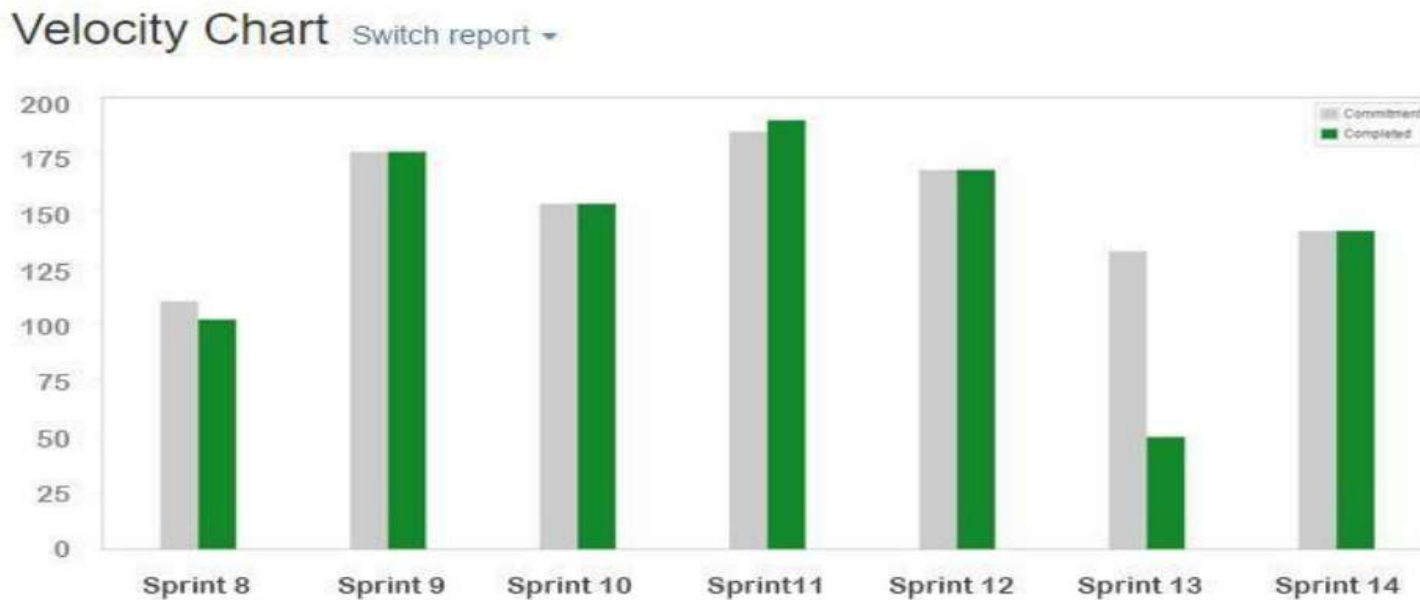
# Agile Reports - Cumulative Flow Diagram

- ▶ A cumulative flow diagram maps the number of work items in each status over a period of time. In a typical example, the number of work items under "To do" goes down over the period of the sprint, and the number of work items under "Done" should plausibly start to increase, at least from the middle of the sprint. A cumulative flow diagram helps you identify bottlenecks in your workflow. By comparing different statuses in your workflow, you can see at which stage they stagnate the most and, consequently, which part of the workflow needs to be more efficient.



# Agile Reports - Velocity Graph

- For every sprint, a velocity graph compares the amount of committed estimation points and completed estimation points. Using this data, the amount of average velocity is calculated. The velocity of a team is a measure of their capacity and gives the team an idea of how much they can take on in the next sprint. A team's velocity is bound to fluctuate initially while the team finds their stride and refines their estimation technique.

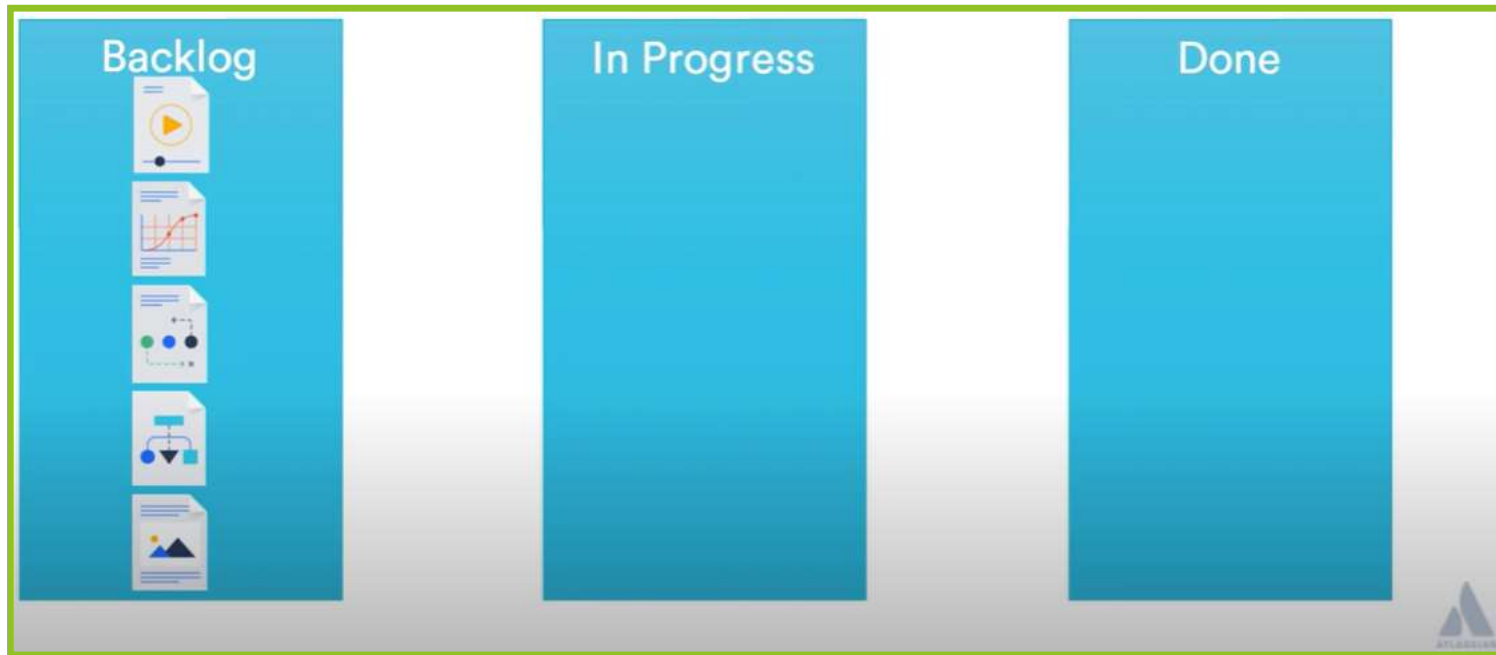


# Sprint Velocity





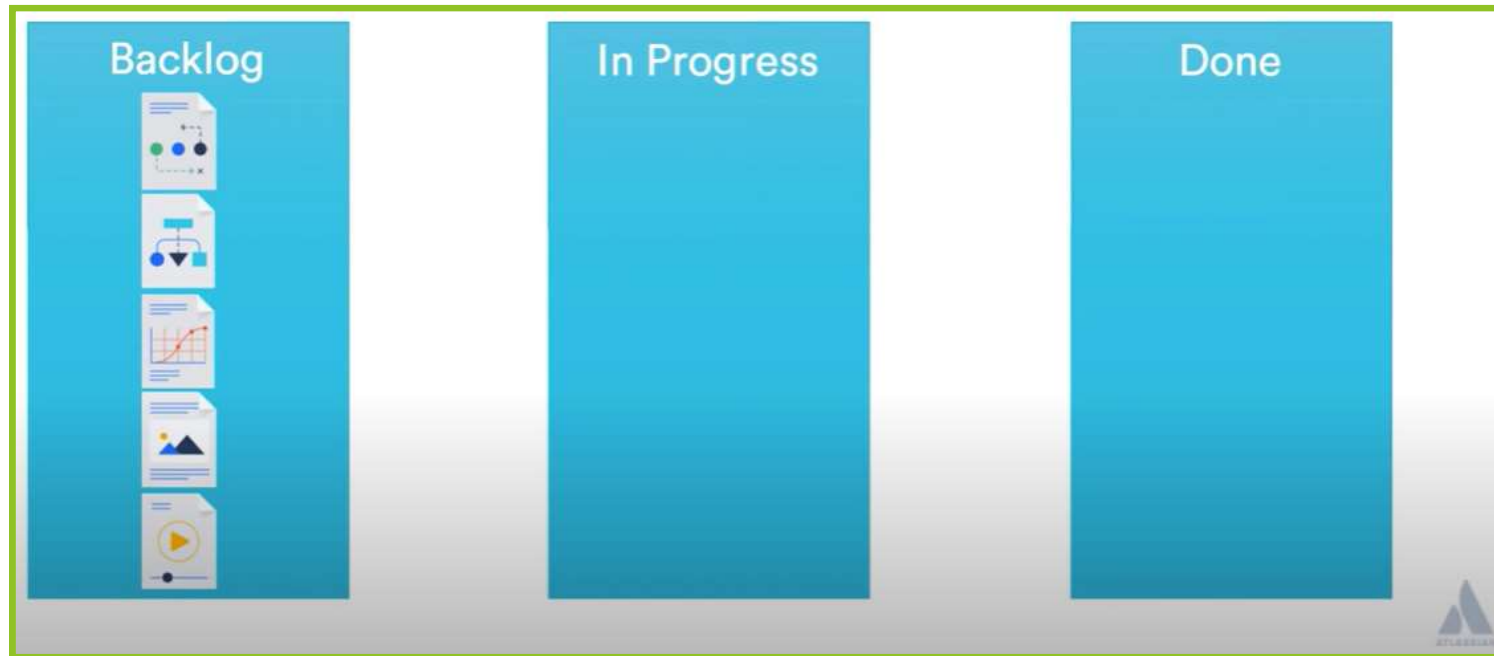
# KANBAN BOARD



STEP 1: Adding work items by support team manager

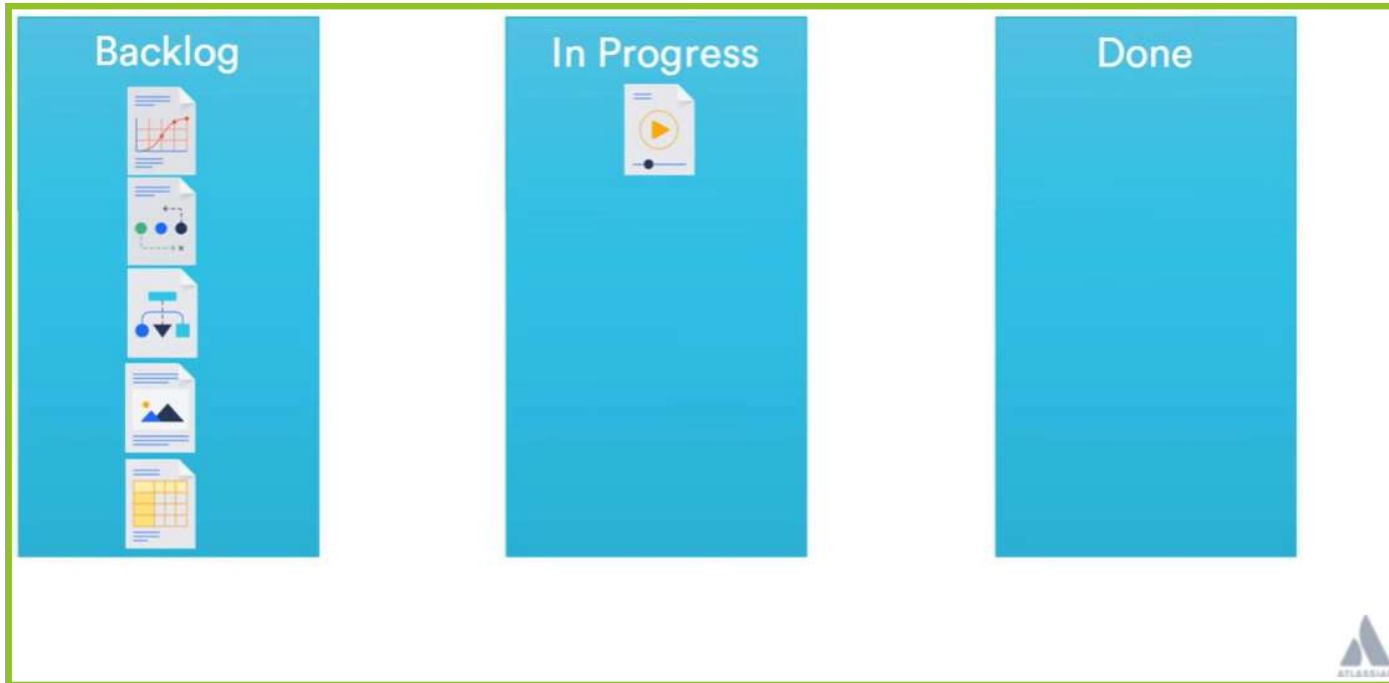


# KANBAN BOARD



Step 2: Manager arranges the items based on priority  
Making it efficient for team members to pick it up

# KANBAN BOARD

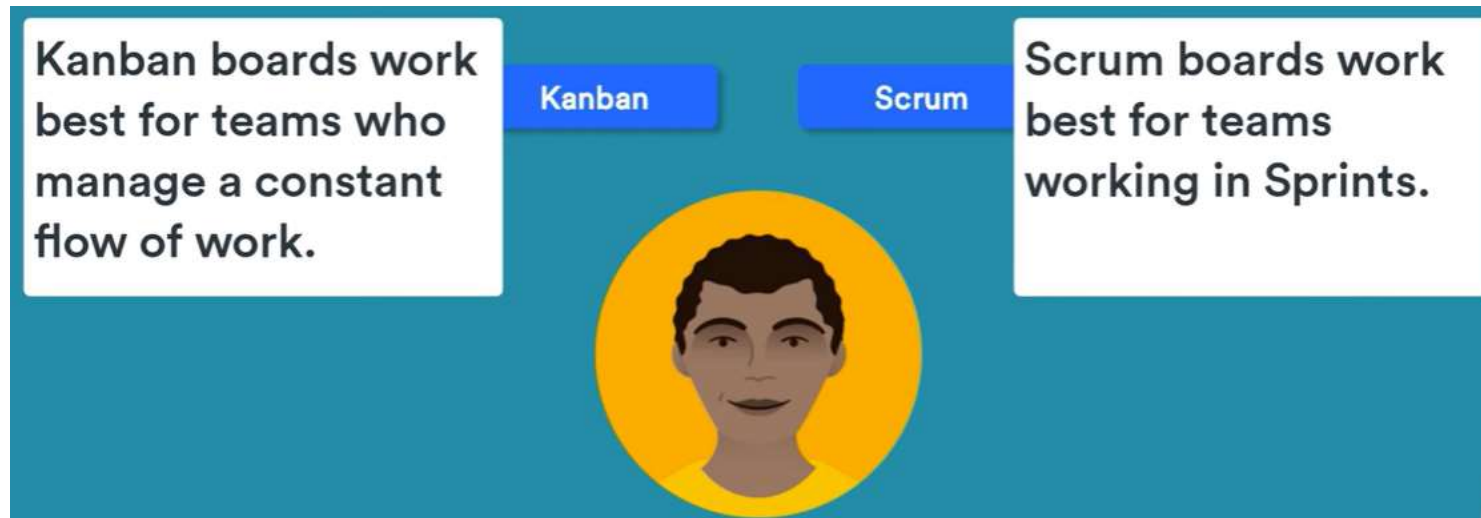


STEP 3: Team selects the items  
KANBAN board manages the continuous flow of work

# KANBAN BOARD



# KANBAN VS SCRUM



# Risk Management

## ► What is Risk

Project risk is an uncertain event or condition that, if it occurs, has a positive or negative effect on one or more project objectives such as scope, schedule, cost, and quality.

A risk may have one or more causes and, if it occurs, it may have one or more impacts.

## ► Risk Probability & Risk Impact:

The probability is the likelihood of an event occurring.

The impact is the consequence to which extent the project is affected by an event.

By combining the probability and impact, the Level of Risk can be determined.

## ► Common Risk Types in Software Projects:

- Continuously Evolving Technologies
- Training Required
- Unproven Technologies
- Employees inclined to new and fancy technologies
- Short stays of employees within one organization
- Employees feel monotonous and lack of career growth by working in same technology
- Non-functional requirements
- Scope change
- Dependency on client's availability and organization of client
- Dependency on 3rd party tools
- Shared Resources

# Risk Management - Identification & Handling

- ▶ Risks Identification Techniques:
  - ▶ Brainstorming
  - ▶ Interviewing
  - ▶ Checklist Analysis
  - ▶ Expert Judgement
- ▶ Risks Handling Techniques:
  - ▶ Avoid - Eliminate the threat or protect the project from its impact.
  - ▶ Transfer - Shift the impact of a threat to a third party.
  - ▶ Mitigate - Reduce the probability of occurrence or impact of a risk.
  - ▶ Accept - Acknowledge the risk and not take any action unless the risk occurs. The most common active acceptance strategy is to establish a contingency reserve, including amounts of time, money, or resources to handle the risks

# Risk Management Plan - Standard Format

ID	Risk Description	Probability	Impact (1-5) (5 is highest)	Rating	Risk Handling Strategy	Mitigation/Comments
1	Client may ask to add more features after initial demos	0.9	4	3.6	Accept	Move new requirements to future sprints
2	Some of the team members may resign during project development	0.7	5	3.5	Mitigate	Create backups
3	Team may face ReactJS specific challenges	0.8	4	3.2	Avoid	Hire an experienced ReactJS engineer
4	Client might not be satisfied with performance of the product	0.5	3	1.5	Mitigate	Consider performance during design & Get performance benchmark/acceptance from client

## Notes:

- Some experts recommend more columns like Risk Category, Risk Assignee etc.
- This can be extended to RMMM (Risk Mitigation, Monitoring & Management) as well by mentioning how each risk will be monitored and managed.
- PMI recommend planning of positive risks (opportunities) as well.

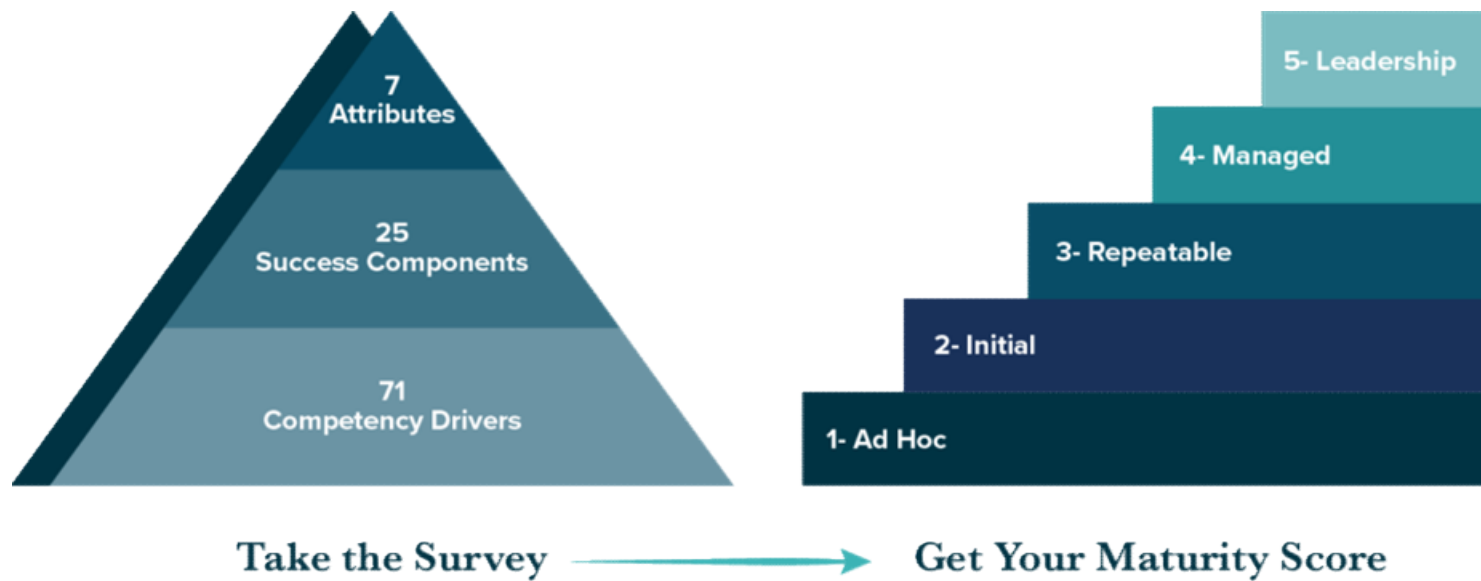
# Risk Management Plan - Simplest Form

## Project Risks:

1. Client may ask to add more features after initial demos
  - Mitigation: Move new requirements to future sprints
2. Some of the team members may resign during project development
  - Mitigation: Create Backups
3. Team may face ReactJS specific challenges
  - Mitigation: Hire an experienced ReactJS engineer
4. Client might not be satisfied with performance of the product
  - Mitigation: Consider performance during design. Also, get performance benchmark or acceptance from client



# Risk Maturity Model (RMM)



Seven attributes, or components of a successful enterprise risk management program, which breaks into 25 success factors

# RMM assessment

- ▶ During the RMM assessment, organizations score how well they embody each of the drivers using three dimensions:
  - (1) *Effectiveness* - Measures the frequency and effectiveness of key risk management activities (i.e. Are assessments ad-hoc or completed annually? Are high risks reviewed at least quarterly?)
  - (2) *Proactivity* - Measures forward-looking risk management versus reactive after events occur (i.e. Is risk management part of employee training? Are emerging risks regularly considered?)
  - (3) *Coverage* - Measures the breadth and depth of risk management within the organization (i.e. Does responsibility span across all departments and all vertical levels of the organization?)
- ▶ On a 5-level scale, receiving a score of (1) ad hoc or (2) initial, indicates a very basic or non-existent ERM program, whereas achieving the upper levels, (3) repeatable, (4) managed, and (5) leadership, indicates ERM programs that are established and increasingly aligned with industry standards.

# Risk Management - How Agile Helps

1. Risk Management is an iterative process and it is monitored and updated in each sprint
  - Daily stand-ups
  - Sprint Review Meetings
  - Retrospective Meetings
2. Collaborative Planning
3. Working software over comprehensive documentation
4. Customer collaboration over contract negotiation

# Contract Management - Contract Types

- ▶ Fixed-price contracts The payment does not depend on the resources or the time spent. It involves setting fixed price for the product, service or result defined in the contract.
- ▶ Cost-reimbursable contracts The contractor is paid its actual costs for the work completed. This places significant financial risk on the client and so these approaches tend to be used where the nature or scope of the work can't be adequately defined.
- ▶ Time and Material Contracts (T&M) Time and material contracts are a hybrid type of contractual arrangement that contain aspects of both cost-reimbursable and fixed-price contracts. They are often used for staff augmentation, acquisition of experts, and any outside support when a precise statement of work cannot be quickly prescribed.

# Contract Management - Influencing Factors

- Higher Level Estimates
- Expertise of the team
- Experience of the team
- Organizational Culture
- Documentation & Domain Backups
- Change Requests & Bugs
- Non-functional Requirements
- Planned & unplanned leaves of Team Members