UNIT-1

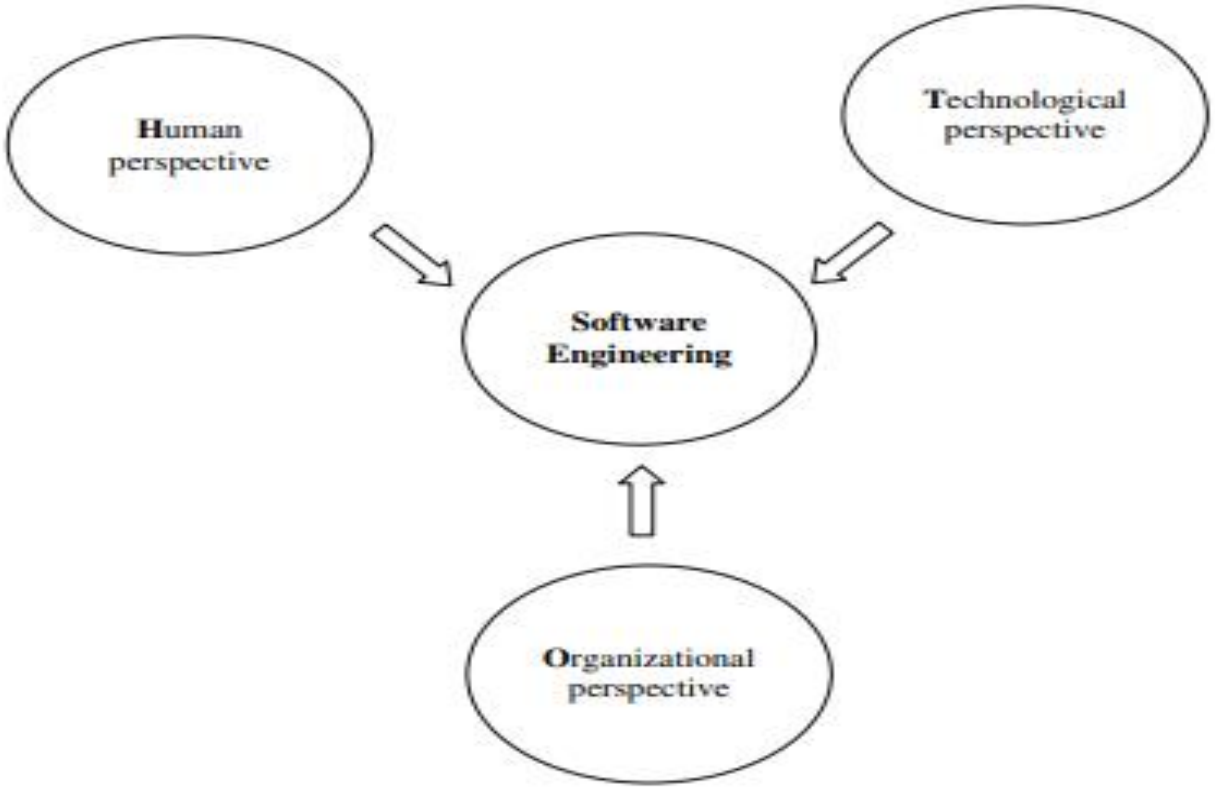
Agile and Scrum Principles

Three Perspectives of software engineering

Human perspective

Organizational perspective

Technological perspective



Agile Manifesto

The Agile Manifesto is a document that identifies four key values and 12 principles that its authors believe software developers should use to guide their work.

Agile Principles

1. Satisfy Customers Through Early & Continuous Delivery
2. Welcome Changing Requirements Even Late in the Project
3. Deliver Value Frequently
4. Business people and developers must work together daily throughout the project
5. Build Projects Around Motivated Individuals
6. Communicate Face-to-face
7. Working Software is the Primary Measure of Progress
8. Maintain a Sustainable Working Pace
9. Continuous attention to technical excellence
10. Simplicity
11. Self-organizing Teams
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Elaborate your view…..

Scrum

Scrum is a framework that helps teams work together. Much like a rugby team training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

Scrum Team

A scrum team is a group of collaborators, typically between five and nine individuals, who work toward completing projects and delivering products.

- One scrum master,
- One product owner and
- A group of developers.

Basic Definitions

- A Sprint is a short, time-boxed period when a scrum team works to complete a set amount of work.

- A Product backlog is a prioritized list of work for the development team that is derived from the roadmap and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first

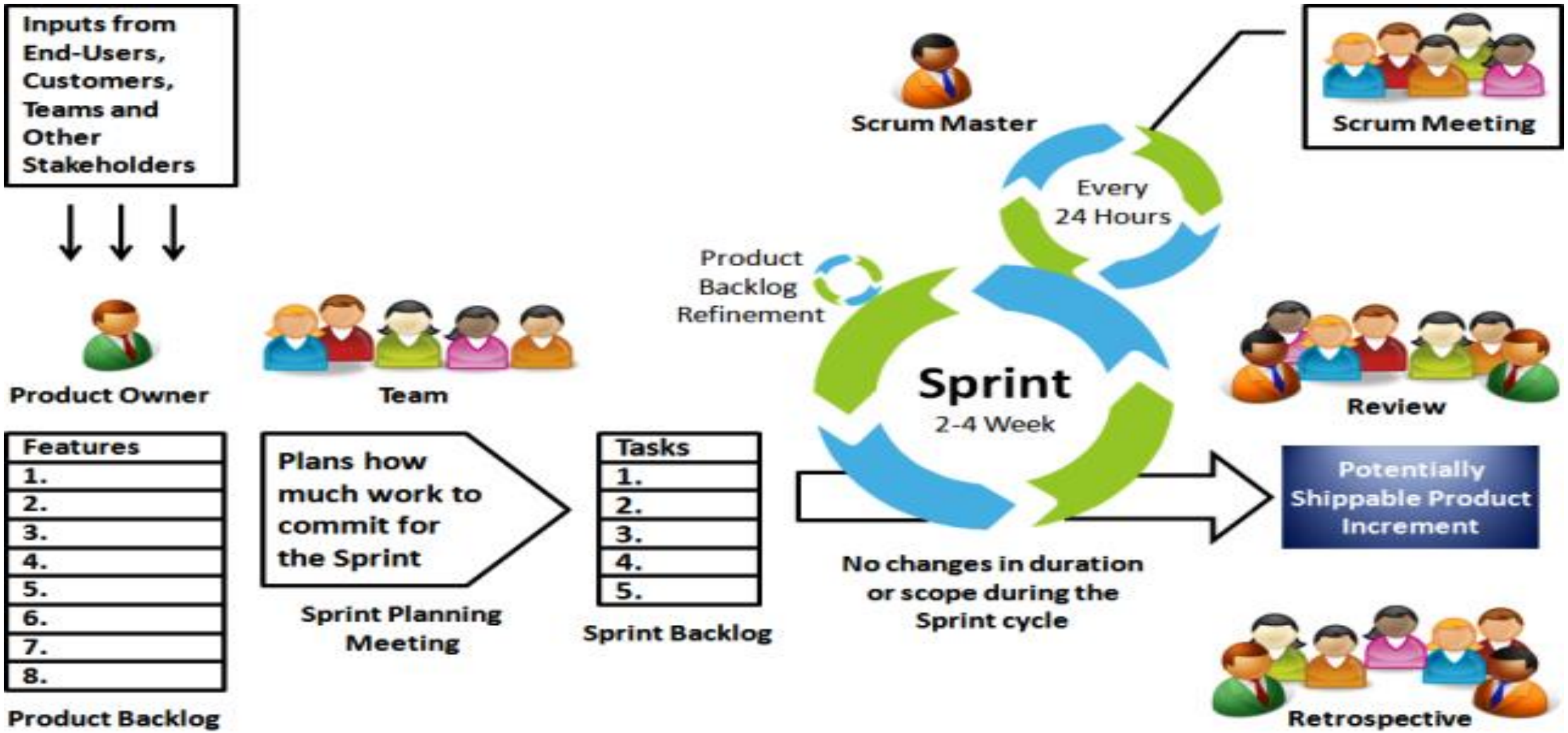
- A Sprint backlog is a subset of the product backlog and lists the work items to complete in one specific sprint. The purpose of the sprint backlog is to identify items from the product backlog that the team will work on during the sprint.

Scrum Events

1. Sprint Planning
2. Daily Scrum
3. Sprint Review
4. Sprint Retrospective
5. The Sprint

THE 3 SCRUM ARTIFACTS

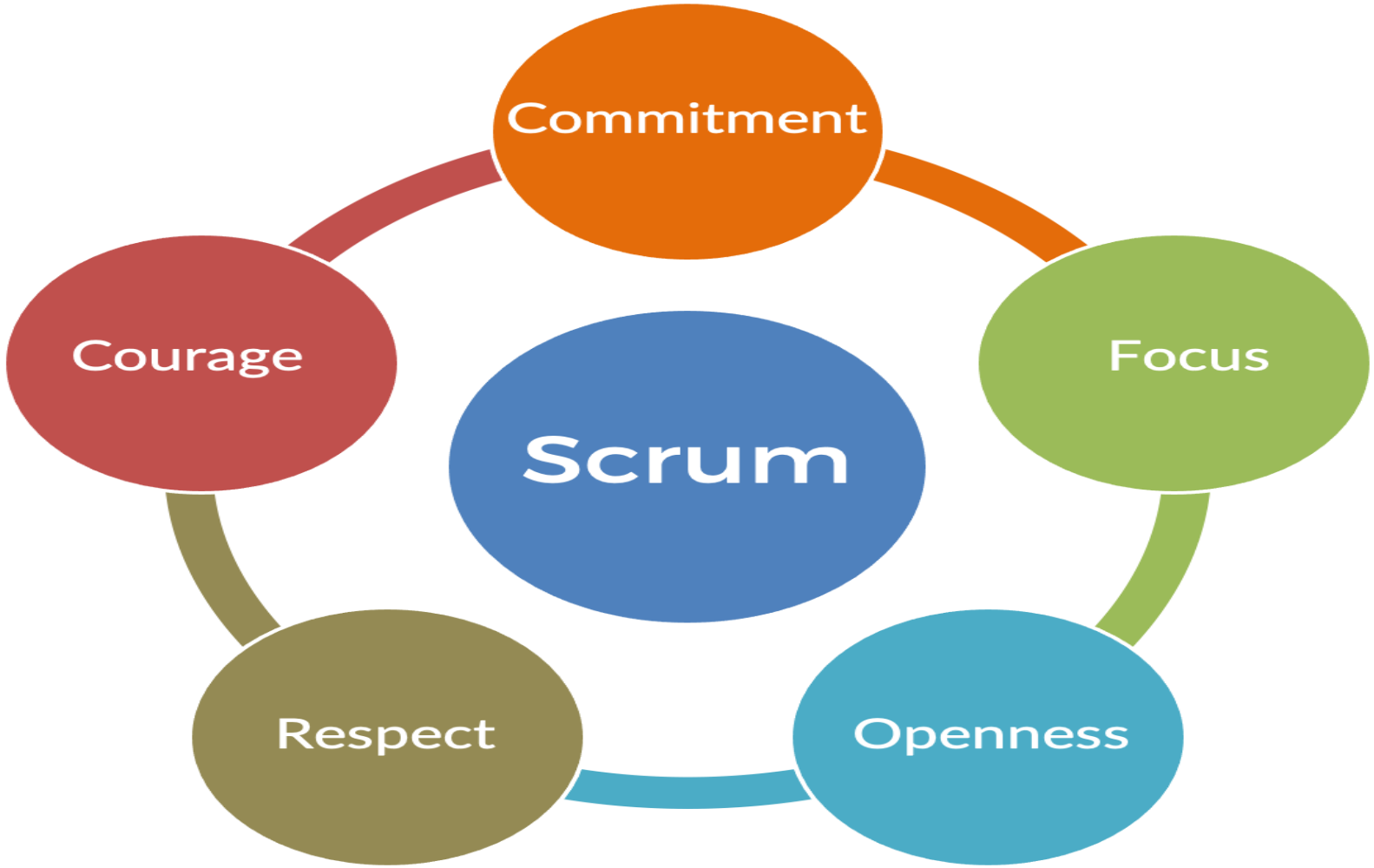
Scrum Process



Uses of scrum

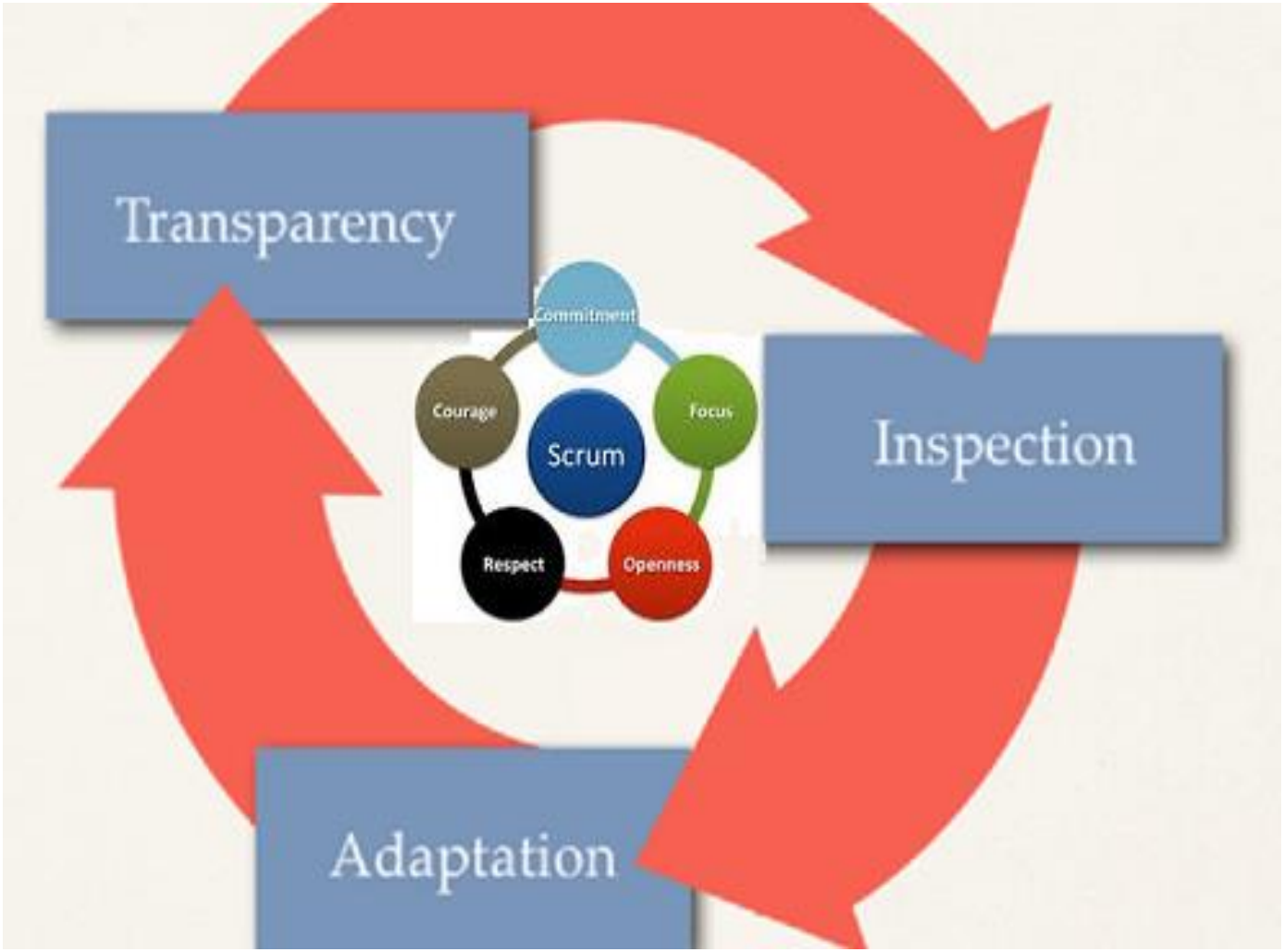
- Scrum can help teams complete project deliverables quickly and efficiently
- Scrum ensures effective use of time and money
- Large projects are divided into easily manageable sprints
- Developments are coded and tested during the sprint review
- Works well for fast-moving development projects
- The team gets clear visibility through scrum meetings
- Scrum, being agile, adopts feedback from customers and stakeholders
- Short sprints enable changes based on feedback a lot more easily
- The individual effort of each team member is visible during daily scrum meetings

The Scrum Values



(Gunther Verheyen – Ullizee-Inc)

Scrum Theory



Kanban

Kanban is a popular framework used to implement agile and DevOps software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.

Team Kanban Board

QUICK FILTERS: Critical partners Only my partners Recently updated

Release

1 To do

- TIS-28 
  Research options to travel to Pluto

4 In progress

- TIS-25 
  Engage Jupiter Express for travel
- TIS-25 
  Add Deimos Tours as a travel partner
- TIS-20 
  Engage Saturn Lines for group tours
- TIS-24 
  Sign Contract for SunSpot Tours

3 Code review Max 2

- TIS-27 
  Engage Saturn Resort as PTP
- TIS-27 
  Engage Speedy SpaceCraft
- TIS-26 
  Reach out to the Red Titan Hotel

1 Done

- ~~TIS-23~~ 
  Engage JetShuttle SpaceWays for travel

Scrum Vs Kanban

	Scrum	Kanban
Origin	Software development	Lean manufacturing
Ideology	Learn through experiences, self-organize and prioritize, and reflect on wins and losses to continuously improve.	Use visuals to improve work-in-progress
Cadence	Regular, fixed-length sprints (i.e. two weeks)	Continuous flow
Practices	Sprint planning, sprint, daily scrum, sprint review, sprint retrospective	Visualize the flow of work, limit work-in-progress, manage flow, incorporate feedback loops
Roles	Product owner, scrum master, development team	No required roles

Teamwork

Teamwork is the collaborative effort of a group to achieve a common goal or to complete a task in the most effective and efficient way.

Role Scheme

Groups	Roles
Leading Group	Coach Tracker Methodologist
Customer Group	User Evaluator Customer Acceptance Tester
Code Group	Designer Unit Tester Continuous Integrator Code Reviewer
Maintenance Group	Presenter Documenter Installer

Dilemmas in Teamwork

How to allocate incentives, rewards, and bonuses among team member?

	Personal Bonus (% of the total bonus)	Team Bonus (% of the total bonus)	How this option may influence teammates' cooperation
A	100	0	
B	80	20	
C	50	50	
D	20	80	
E	0	100	

Teamwork in Learning Environments

1. Assign Roles to Team Members.
2. Role Activities
    - Role Assignment Activities
    - Role Maintenance Activities
    - Role Improvement Activity

UNIT-2

Agile estimation techniques, Customers And Users, Time

Agile estimation techniques

- T-Shirt Sizing
- Sprint Poker
- Three-Point Method
- Affinity Estimation
- Relative Mass Evaluation
- Dot voting
- Maximum allowable size (MAS)
- Big, Uncertain, Small.

T-Shirt Sizing

T-Shirt Size	XS	S	M	L	XL
Estimate	1	2	3	4	5

Smaller Than XS = a Task.

Extra Small = 1

Small = 2

Medium = 3

Large = 5

Extra Large = 8

Larger than XL = an Epic

T-Shirt Sizing Procedure

1. Product Owner will explain the story to be estimated and the development team will ask questions if they have any issues
2. Each developer gives each story, a t-shirt size.
3. All in members in the development team will raise their cards simultaneously.
4. The development team will discuss the differences.
5. The product owner explains the story further or clarifies misunderstanding if any.
6. The team will Go back to Step 2-Step4 until all are agree with one size.
7. Complete or place the stories in size buckets.
8. Estimate the time to complete all stories in S, M, L, XL buckets.

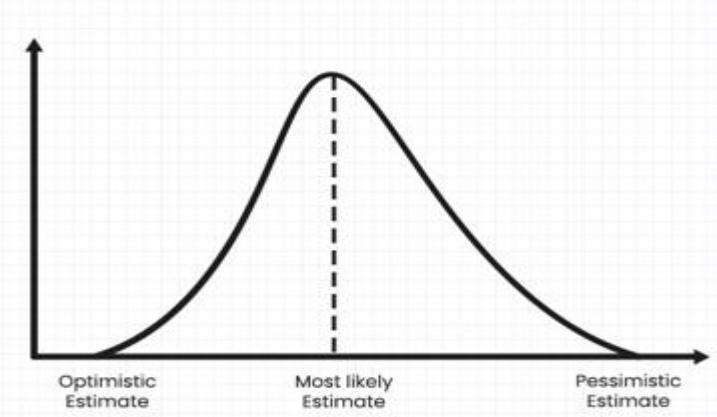
Sprint Poker

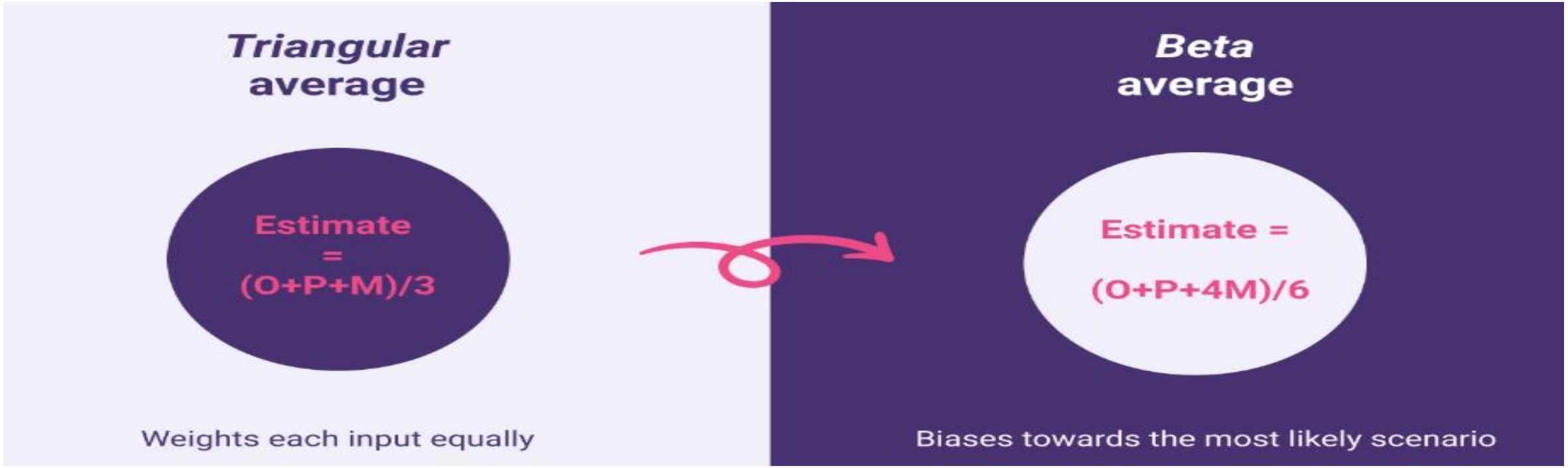
1. The product owner or customer reads an agile user story or describes a feature to the estimators.
2. Each estimator is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
3. The estimators discuss the feature, asking questions of the product owner as needed. 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.
4. 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.
5. 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.

Three-Point Method

Three-point Estimation looks at three values –

- the most optimistic estimate (O),
- a most likely estimate (M), and
- a pessimistic estimate (least likely estimate (L)).





Affinity Estimation

Three steps of Affinity Estimation are:

1. Silent Relative Sizing
2. Editing the Wall
3. Placing items in correct bucket

Affinity Estimation Procedure

Step 1: Silent Relative Sizing

- First a horizontal scale is chosen. One end of the scale is marked with “Smaller” and the other end is marked with “Larger”
- The product owner provides the user stories to the team.
- Each participant estimates their story solely without any discussion with other participants and after estimation, places it at the correct location on the scale, anywhere between “Smaller” and “Larger”.

Step 2: Editing the Wall

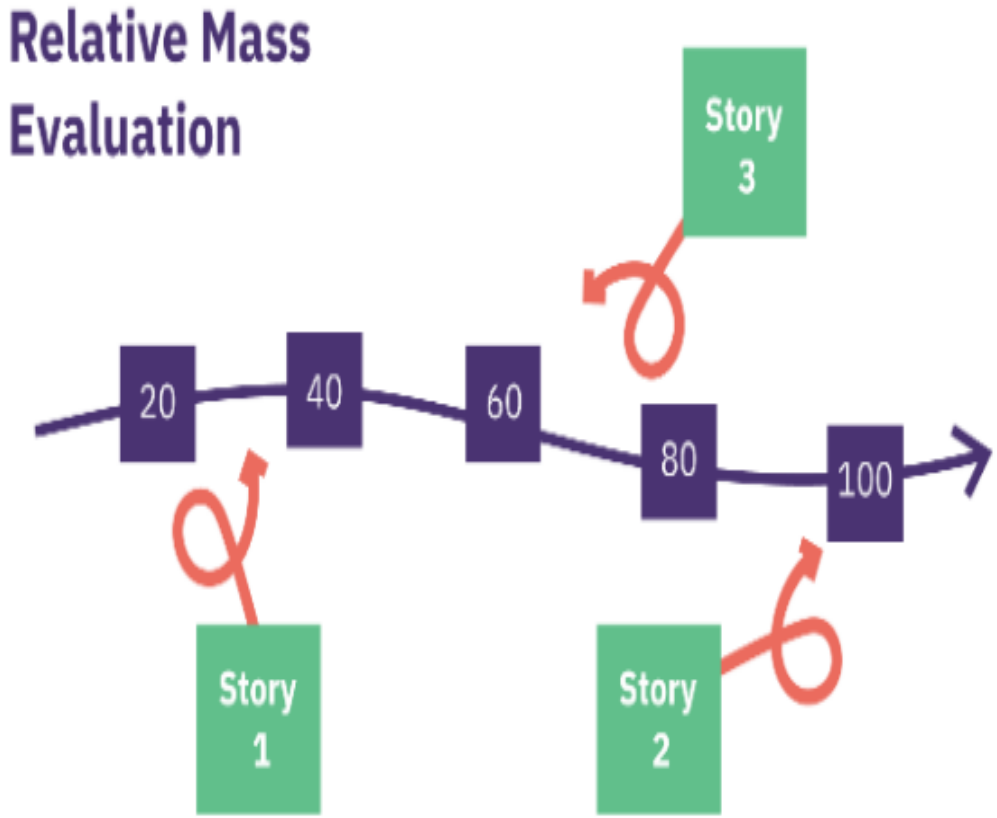
- Once each story is placed on the scale team members edit the relative sizes on the wall. For this team discussed about the story with each other about its design, implementation or any challenges. Team gets all the doubts clarified from Product Owner as well.
- Based on the understanding made during discussion, team re-arranges its story on the scale if required.

Step 3: Placing items in correct bucket

- The scale “Smaller” to “Larger” is divided and marked appropriately with the markers of XS, S, M, L, and XL if we use t-shirt sizing technique or with 0, 1, 2, 3, 5, 8, 13, and so on, if we use Fibonacci series of planning poker estimating technique.
- Team member now places their stories, which were on scale “Smaller” to “Larger”, to appropriate bucket of the converted scale.

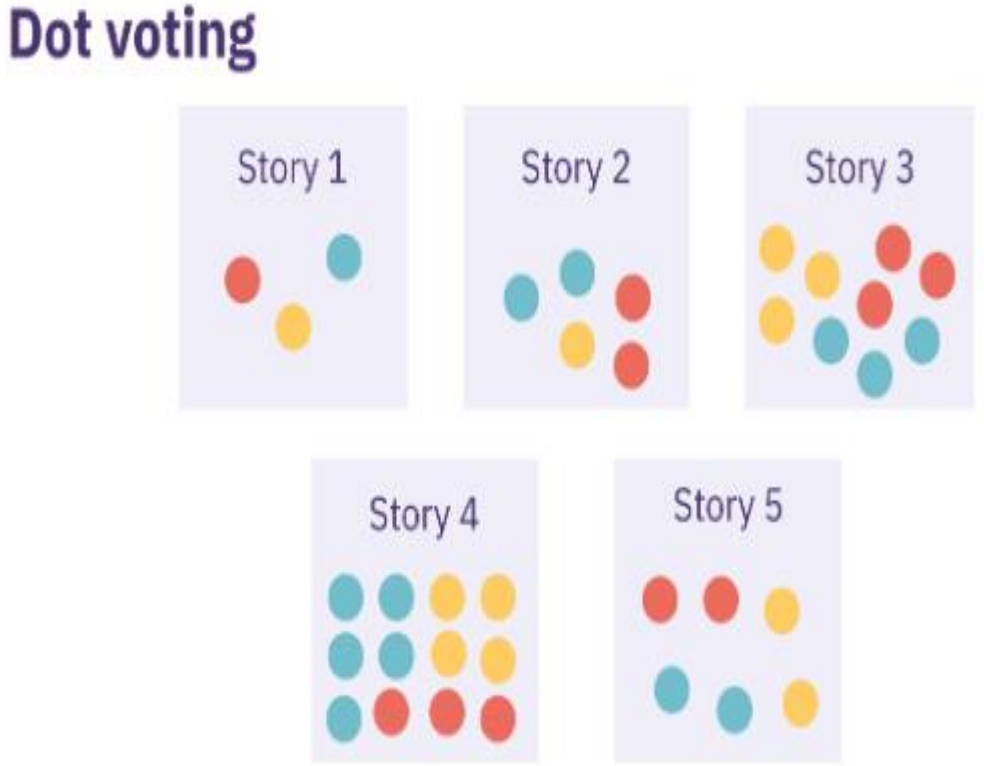
Relative Mass Evaluation

1. Write up a card for each story.
2. Then set up a large table so the stories can be moved around easily relative to each other.
3. Pick any story to start, then get the team to estimate whether they think that it is relatively large, medium, or small.
4. If it's a large story, place it at one end of the table. If it's a small story, it goes at the other end of the table. A medium story goes in the middle. Now select the next story and ask the team to estimate if it's more or less effort than the story that you just put down. Position the story card on the table relative to the previous card, and go to the next card.



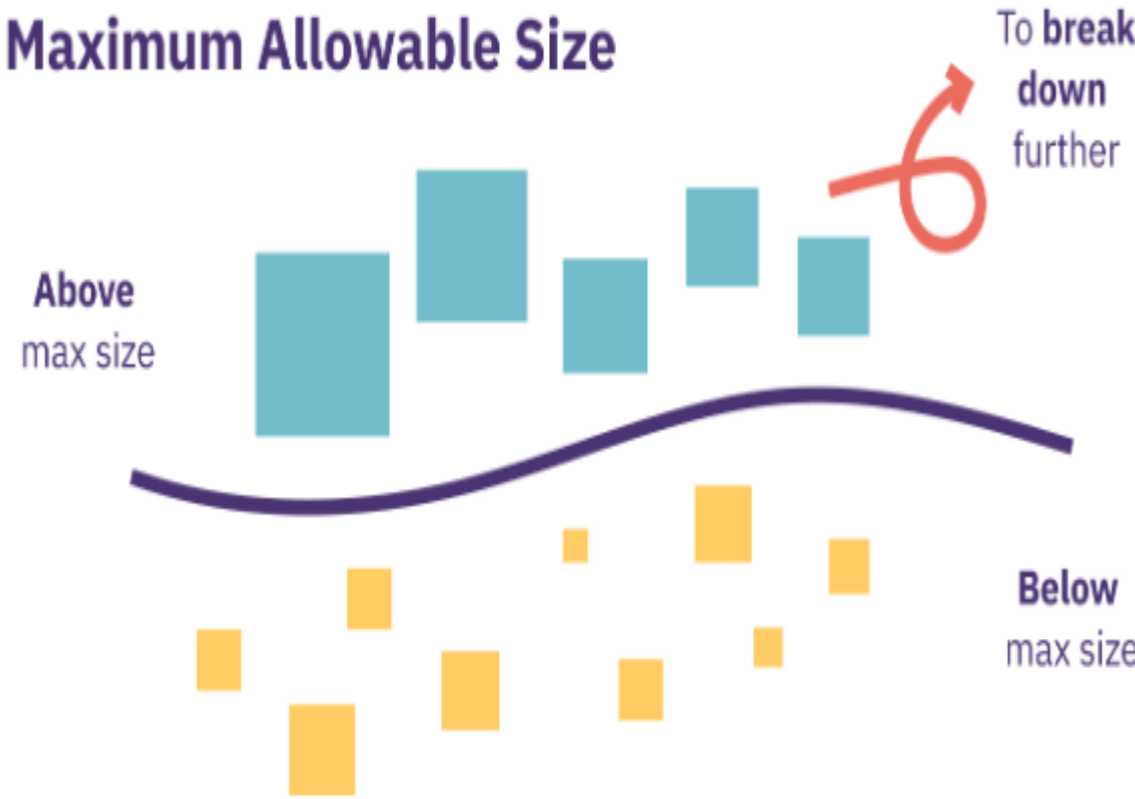
Dot Voting

1. All user stories are put on a wall, virtual or real by the Product Owner.
2. Every team member is given 4-5 votes; these can be small round sticky notes.
3. Every team member is asked to give their votes on the stories they think are bigger.
4. Team member puts/pastes the round red sticky notes on the stories they think are big in size.
5. Every team member performs this process until their all 4-5 votes are exhausted/used. At the end of this process, the story with higher votes is termed as biggest and that with low number of votes are smallest.
6. Product owner then orders the story from higher votes to lower votes.
7. One person can vote more than once for one story.
8. Same technique is used to decide priority as well. More votes means higher priority item.



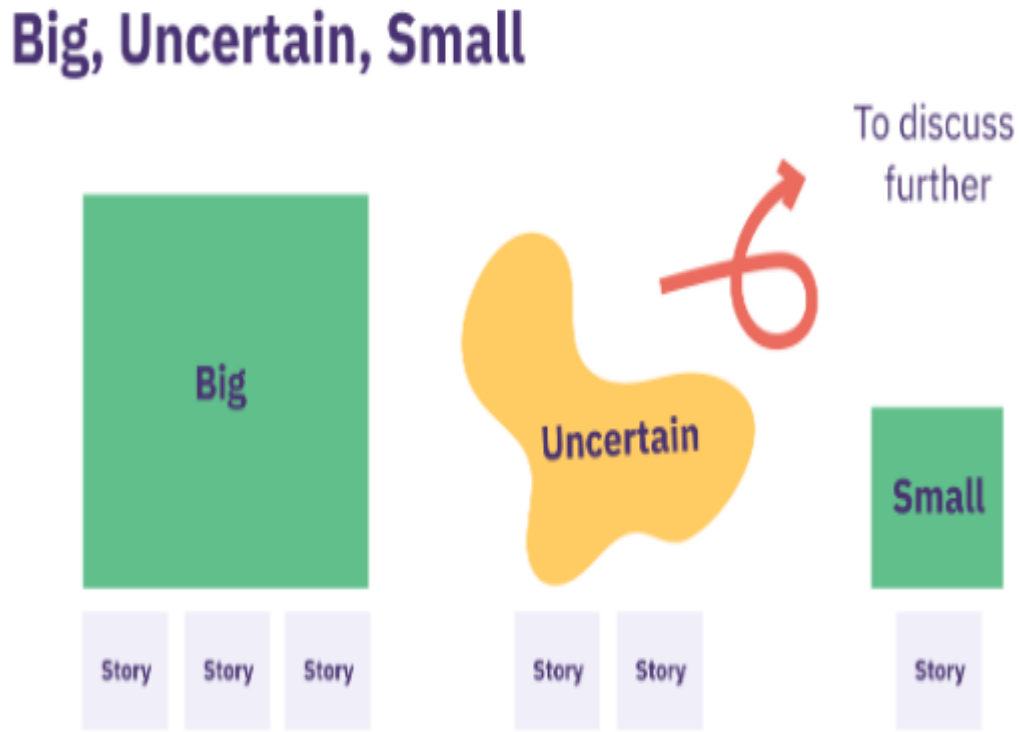
Maximum allowable size (MAS)

1. Set your estimation scale. In this case, it might be worth using a scale according to the number of hours something will take, so you have a concrete idea of the maximum amount of time you want your team to spend on a given piece of work. This will make your filtration more actionable.
2. As a general rule, user stories probably shouldn't take more than 16 hours of work, otherwise they can become unwieldy. So you might want to take 16 hours as your maximum allowable size. If you don't need the same level of precision, use the good old t-shirt sizes to guide you.
3. Now the task is to go through all the stories in your backlog and filter out the ones the team think are higher than 16 hours by voting on them.
4. You can then take those items and discuss how to break them down so they fit below the maximum allowable size.



Big, Uncertain, Small

The ‘Big/Uncertain/Small’ agile estimation method is similar to the bucket system. It involves placing the estimated items in one of the three categories. Teams discuss the items and then a ‘divide-and-conquer’ technique used to estimate the rest of the items. The items left are distributed among team members and can be quickly estimated in parallel.



Customer and User

- Who is a customer?
- Who is a User?
- Can customer be a user?
- Can a user be a customer?
- Who can be a user?
- Who can be a customer?

Customer Vs User

CUSTOMER	CONSUMER
A person who buys the products or services from a shop or business	Person who uses these products or services
Since it is the customer who spends the money and buys the products or goods, the main focus person in the business world is the customer	Since it is the consumer who actually consumes/uses the specific products, he or she may identify the genuineness of the marketing procedures