Agile and JIRA

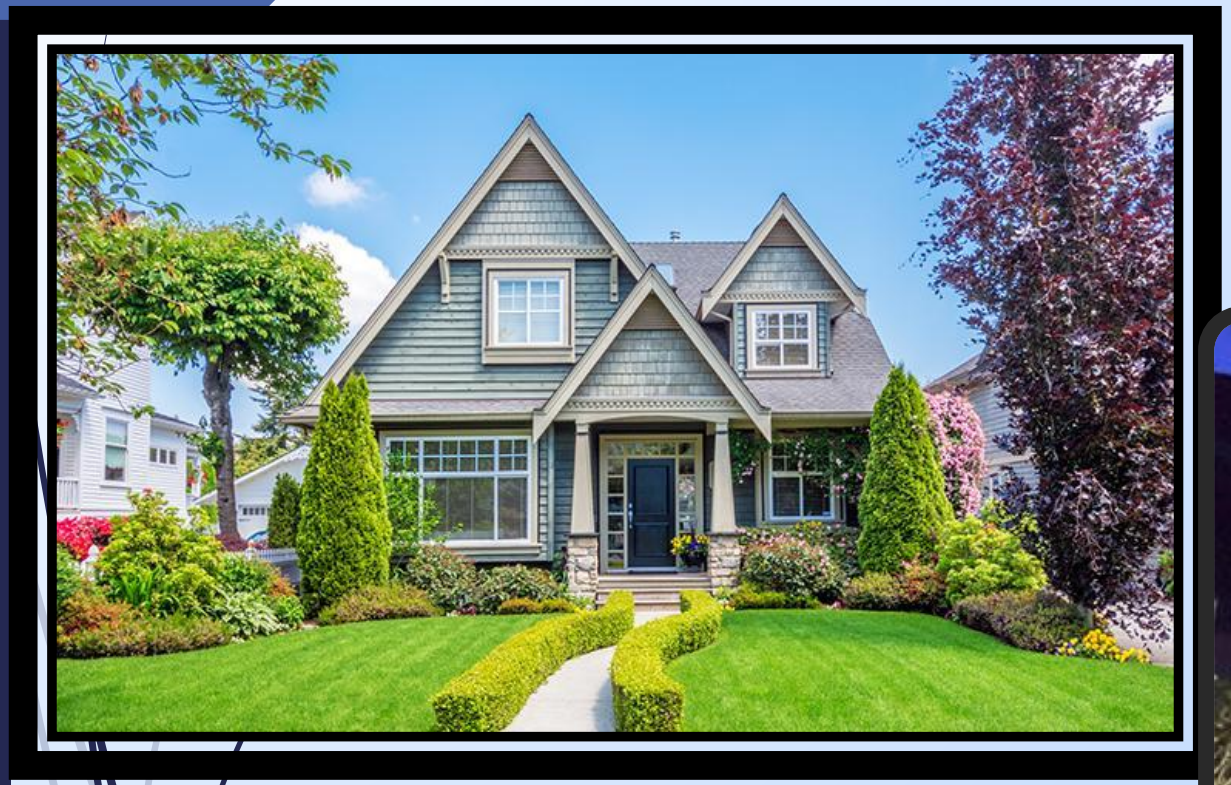
Sarower Ahmmed

Want To Buy A House?

My Need?

- To Buy a house
- Its called **concept or need** in IT world

Your dream house?



Dream house Criteria in detail- **Requirement**

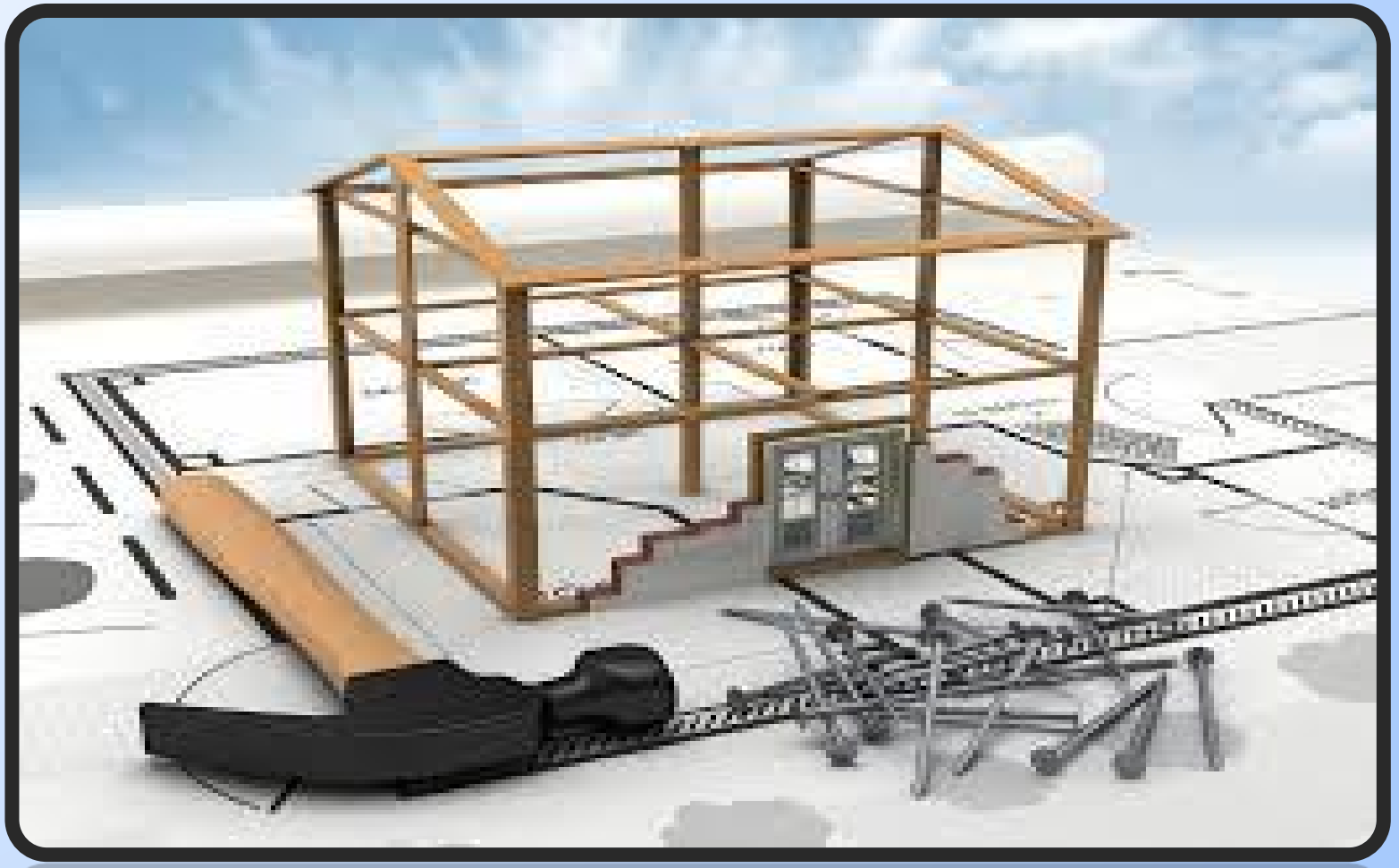
1. ……
2. ……..
3. ……
4. ……
5. ……
6. ……
7. ……
8. ……


What Do You Need To
BUY A HOUSE?
GCH
GreatColoradoHomes.com
Colorado Springs Real Estate

You should need some ability to buy dream house

- Financial ability
- ……
- ……
- ……
- ……

If you want custom build house: Need Design from architecture /design team



Need a team of construction worker to build Dream house

If you buy ready house :: like

- 3rd party application or software
  - No need own design team
  - No need development

When development done or before buy: need to check as it fulfill your requirement- **Testing**

Repaired and maintenance your dream house if required


Software
Software


SODA
MINECRAFT


Snapchat


Like us on:
facebook

SDLC

Stages:

- Concept
- Requirement
- System design
- Coding/ Development
- Testing
- Implementation & maintenance

PAPER BASED

HANDS ON

QA responsibility in SDLC

Responsibility	Stages
No active involvement, But can Participate on meetings,	Concept
Read requirement document try to understand as requirement : Clear, Consistence and Testable	Requirement
Read design document, Writing test cases based on design prototype of application	Design Stage
Black box tester not active but White box tester do unit testing.	Development Stage
All testing need to do in this stage with manage Defect Life cycle	Testing Stage
Do regression test if any update version or defect fix	Implementation & Maintenance

STLC stages: SDLC stages

- Req analysis
- Test process
  - Test plan
  - Test case
  - Test data
  - RTM
  - Executed
  - Defect manage
  - Test reporting

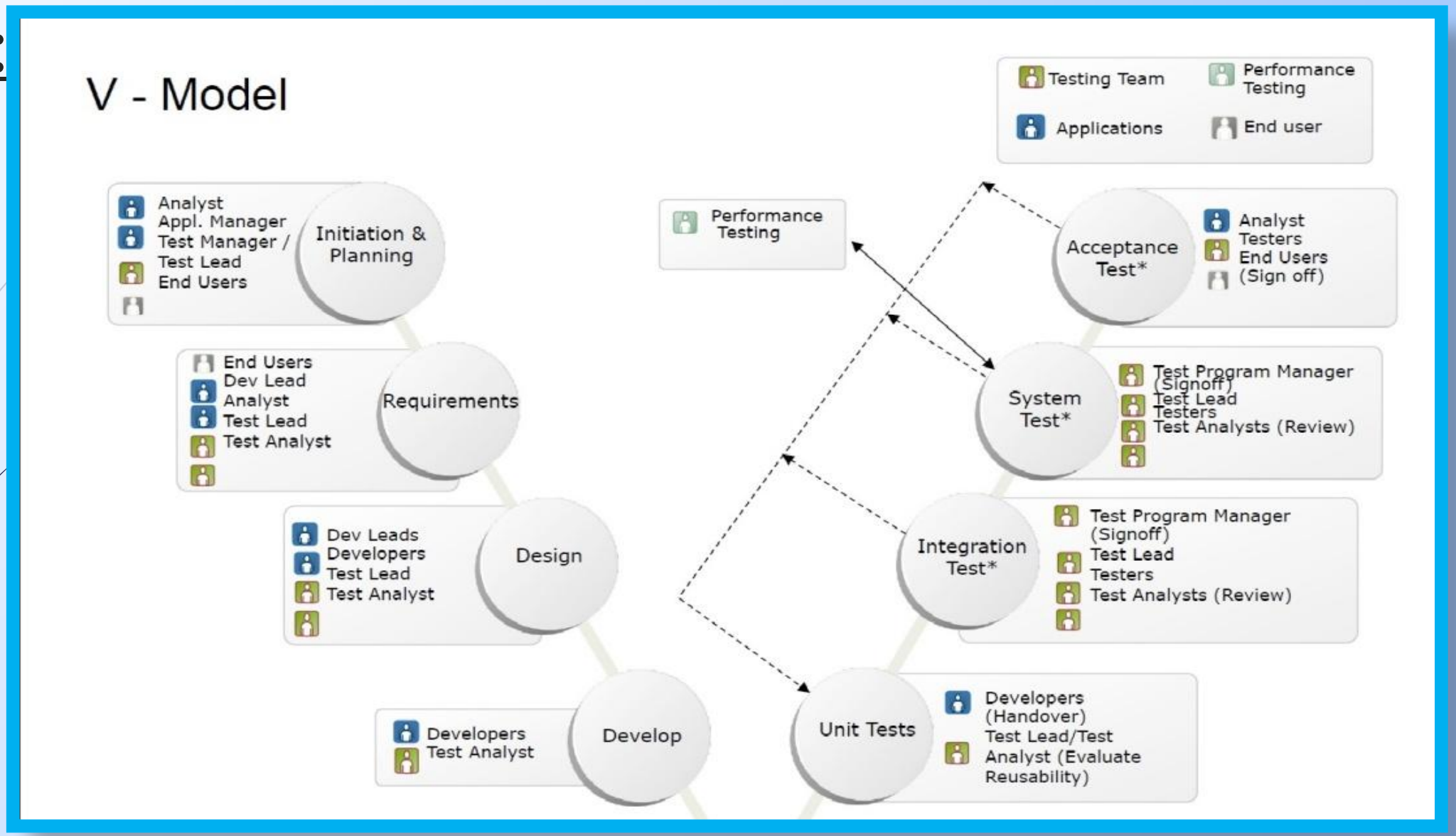
On PAPER

Pass

Fail

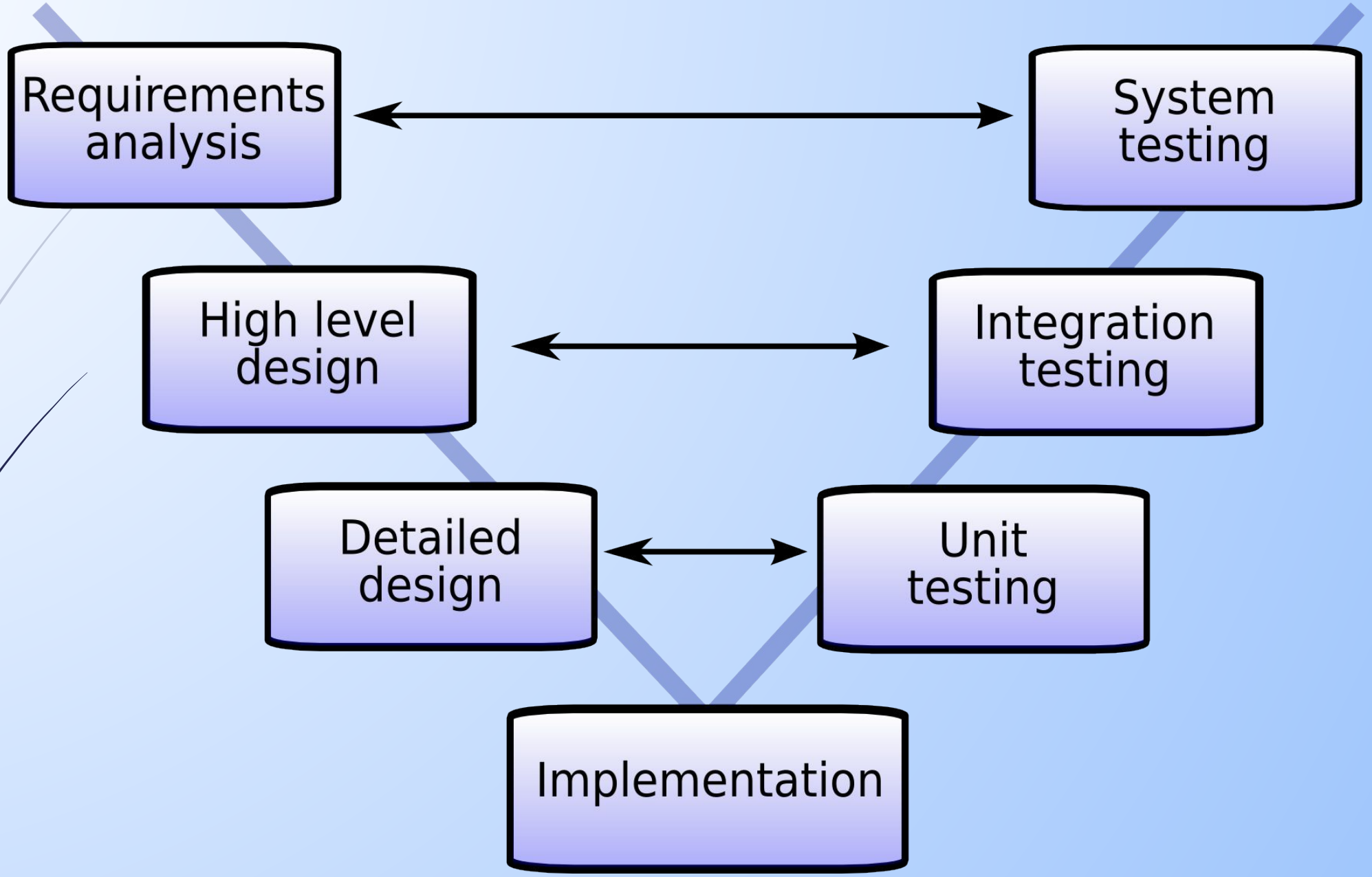
- Concept
- Req
- System design
- Code/ Development
- Testing
- Implementation & maintenance

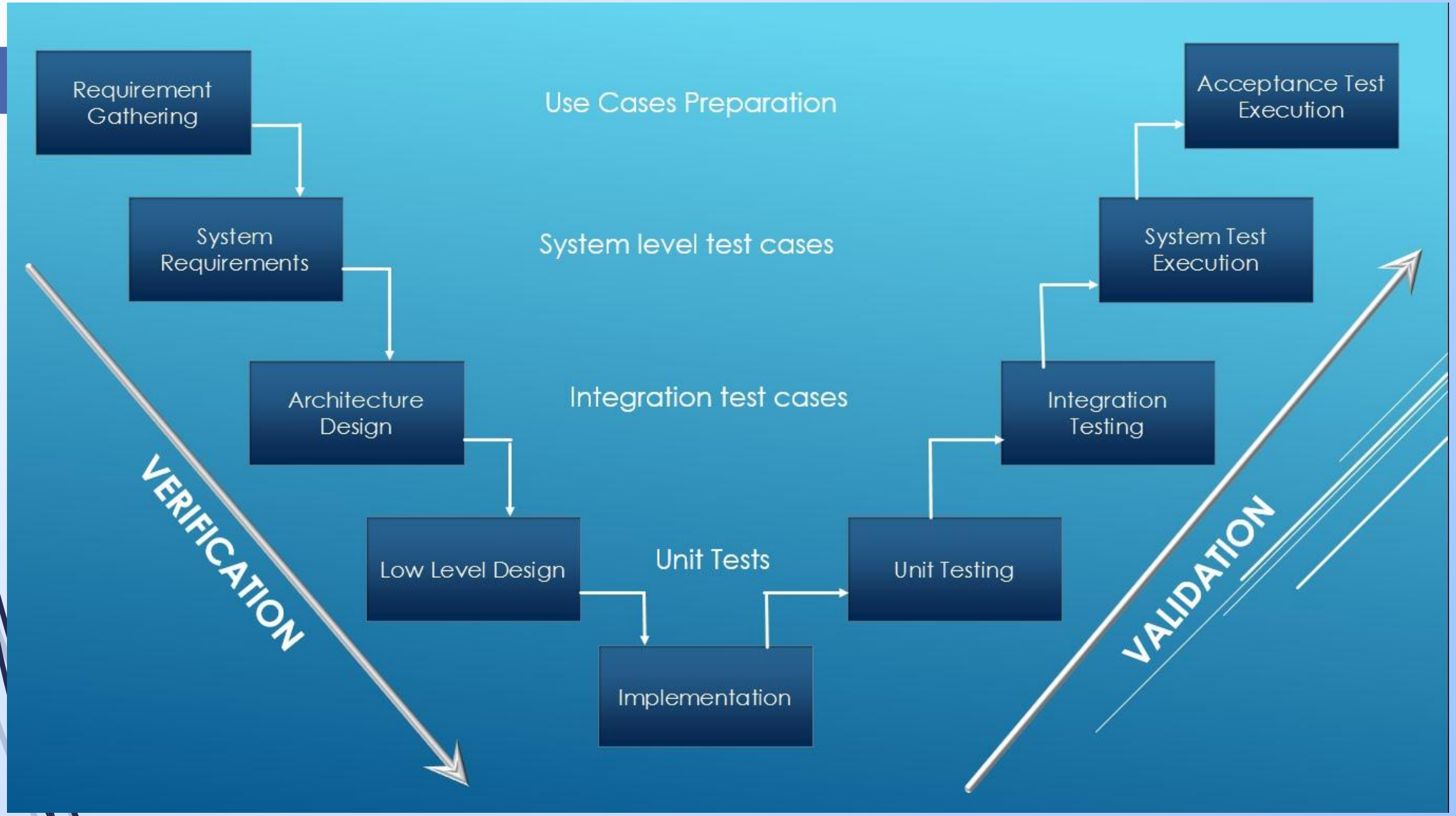
V model:

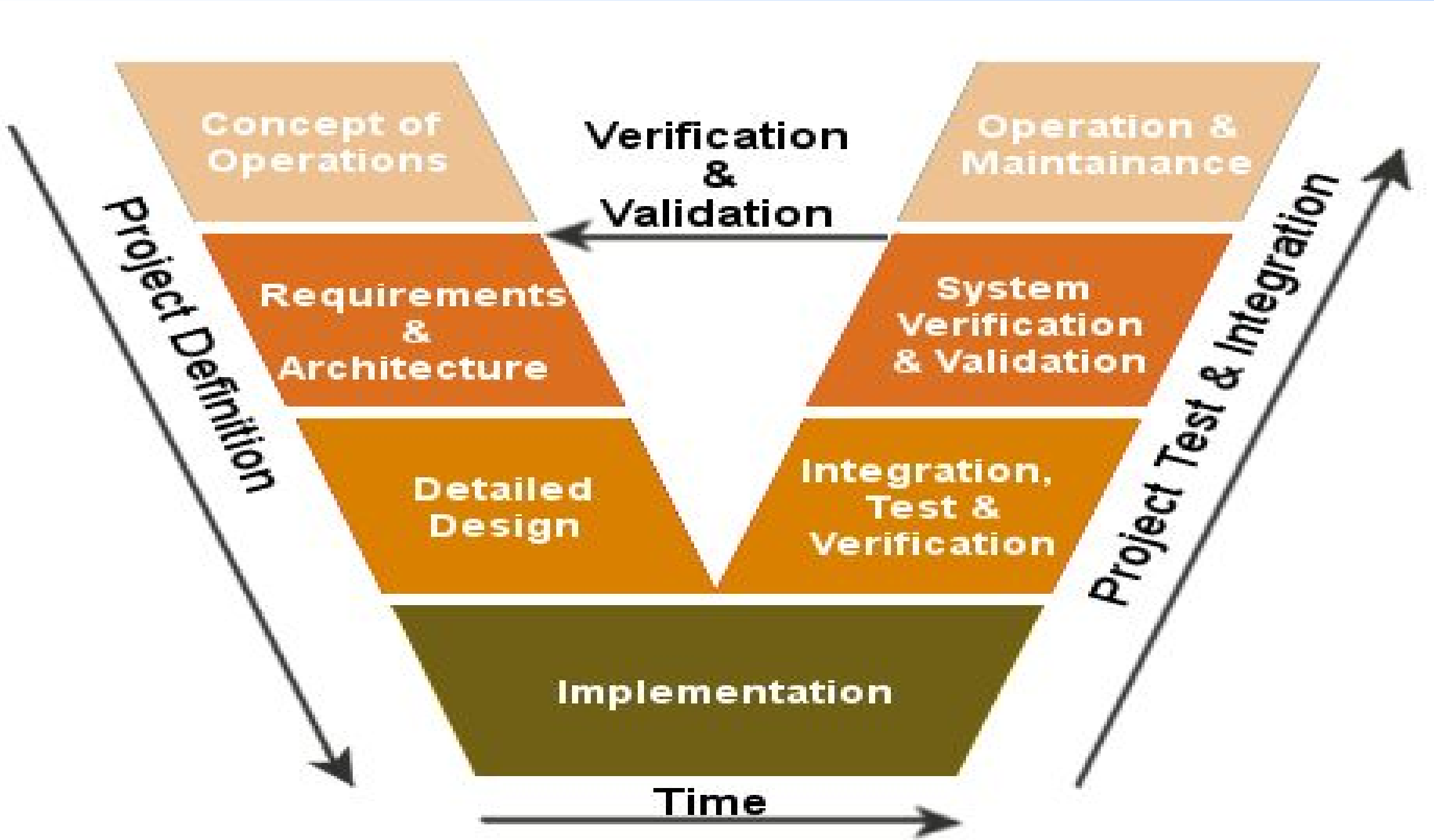


- **Two things:**

  Lt side > verification. Rt side > validation


Requirements analysis
System testing
High level design
Integration testing
Detailed design
Unit testing
Implementation


Requirement Gathering
Use Cases Preparation
Acceptance Test Execution
System Requirements
System level test cases
System Test Execution
Architecture Design
Integration test cases
Integration Testing
VERIFICATION
Low Level Design
Unit Tests
Unit Testing
VALIDATION
Implementation

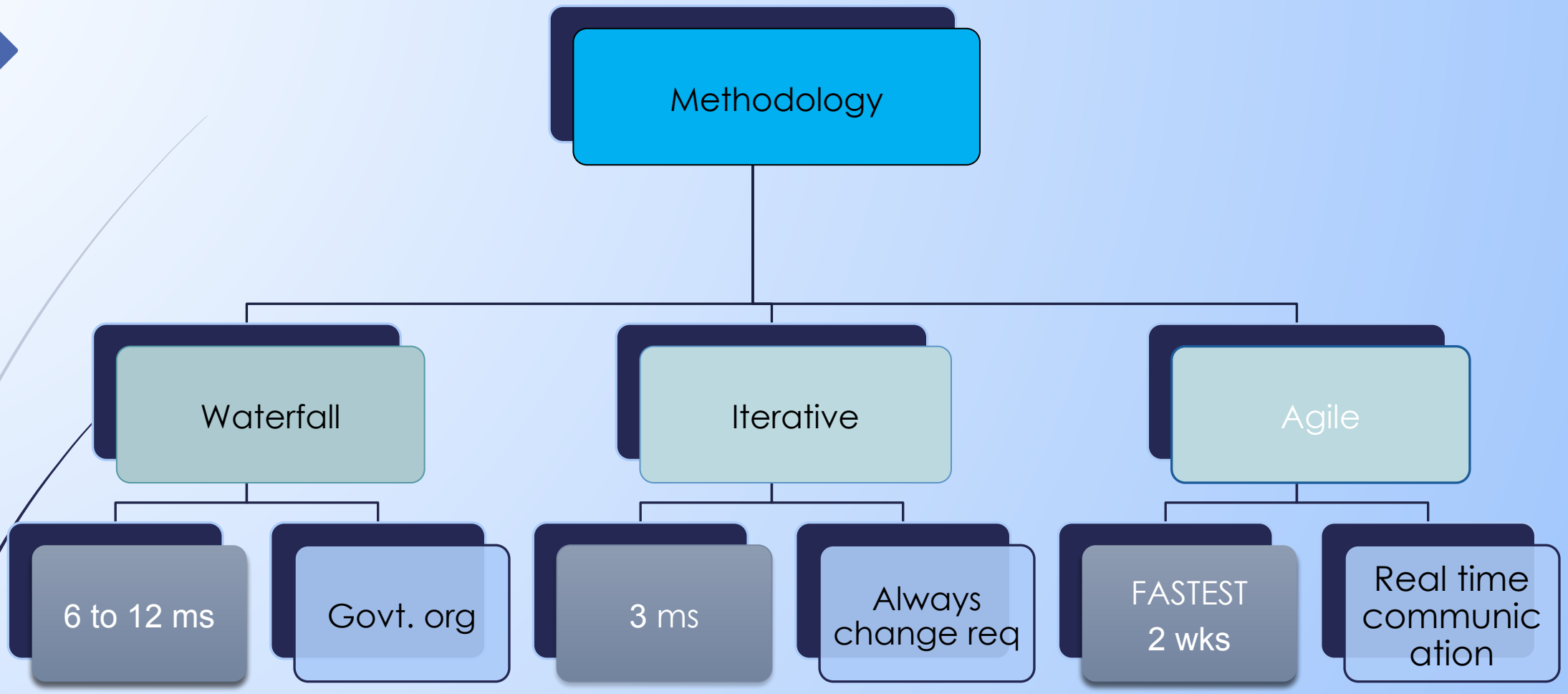

Concept of Operations
Verification & Validation
Operation & Maintainance
Requirements & Architecture
System Verification & Validation
Detailed Design
Integration, Test & Verification
Implementation
Project Definition
Project Test & Integration
Time

Methodology:

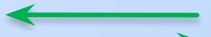
➢A standard/guideline how to follow the stages.

- **Types:**
  - Waterfall
  - Iterative
  - Agile

- Purpose:
  - Faster
  - Better
  - Cheaper

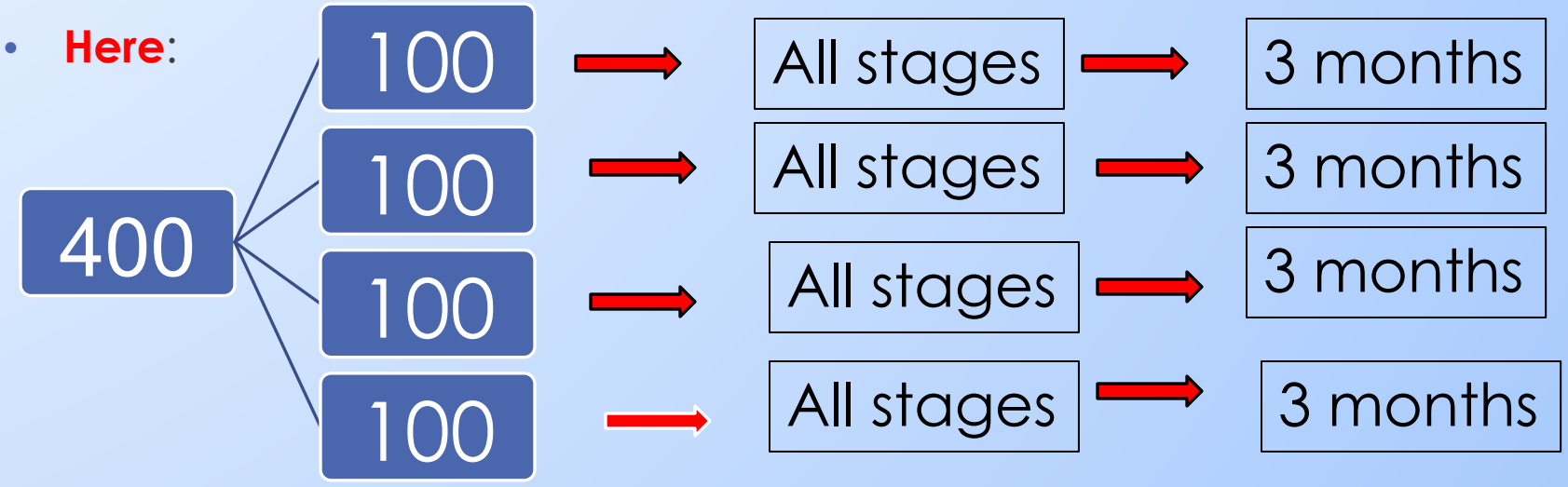

Methodology
Waterfall
Iterative
Agile
6 to 12 ms
Govt. org
3 ms
Always change req
FASTEST 2 wks
Real time communic ation

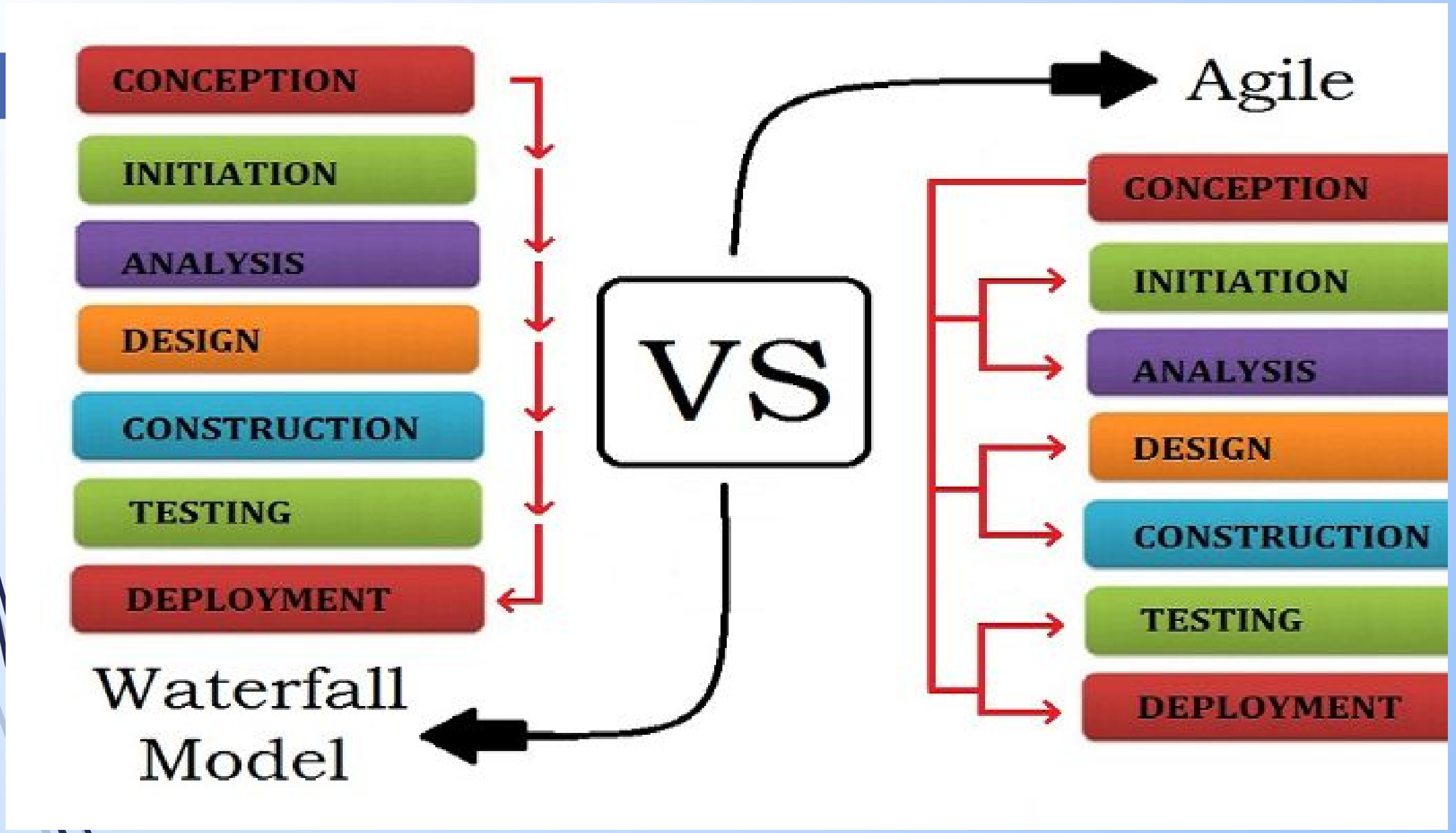
Waterfall

- The oldest.
- Must be sequential-Waterfall Model is a flow based model, in which we pass every phase once, and can not go back to that phase again.
- **Type**:
- **Flexible waterfall**: requirement  system design.
- **Stick waterfall-**
  - Can't go back and front,
  - Can't change the requirement.
- **Disadvantage:**
- Long wait time> 6 to 12 months=> private comp not use.
- **Advantage:** If we know what customer requirement>then its best to use.

Iterative

- Build a system with increment.
- We can changes the requirement-we can always come back to previous phases, and make the changes accordingly.
- **Example:**
- Supposed we have **400** requirements
  - Waterfall=>> all **400** will be done in **1** yr
- **Here**:




CONCEPTION
INITIATION
ANALYSIS
DESIGN
CONSTRUCTION
TESTING
DEPLOYMENT
Waterfall Model
VS
Agile
CONCEPTION
INITIATION
ANALYSIS
DESIGN
CONSTRUCTION
TESTING
DEPLOYMENT

Agile Methodology:

- Mostly used.
- More atomic but its a similar type of Iterative
- It can deliver within 2 wks to 4 wks =>> So its FASTER
- Handle less number of requirements.
- Real-time communication:
  - Less documents
  - 1 developer for 1 tester for 1 business analysis

Scrum calls for four ceremonies that bring structure to each sprint:

- **Sprint planning:** A team planning meeting that determines what to complete in the coming sprint.
- **Daily stand-up:** Also known as a daily scrum, a 15-minute mini-meeting for the software team to sync.
- **Pre Planning (Backlog/sprint Grooming): When:** Mid sprint
  - everyone helps prepare the scrum backlog for the sprint planning meeting. This usually includes adding new stories and epics, extracting stories from existing epics, and estimating effort(t-shirt sizing) for existing stories.
- **Sprint retrospective:** A review of what did and didn't go well with actions to make the next sprint better.

Daily Scrum Meeting:

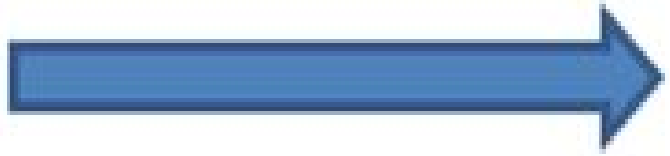
- Called Daily Standup meeting
- Scrum meeting held every day and always 3 Qs and time 10 to 15 mints:

  - What did u do yesterday?
  - What will u do today?
  - Any major issue that requires escalation?

Scrum Master

Scrum Meeting

Product Owner



Prepare product backlog

Team organizes tasks

Three essential roles for scrum success

- **The product owner:**
  - Build and manage the product backlog
  - Closely partner with the business and the team to ensure everyone understands the work items in the product backlog
  - Give the team clear guidance on which features to deliver next
  - Decide when to ship the product with the predisposition towards more frequent delivery

The scrum master

- Head of Sprint
- Deeply understands the work being done by the team and can help the team optimize their delivery flow.
- As the facilitator-in-chief, they schedule the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and the sprint retrospective.

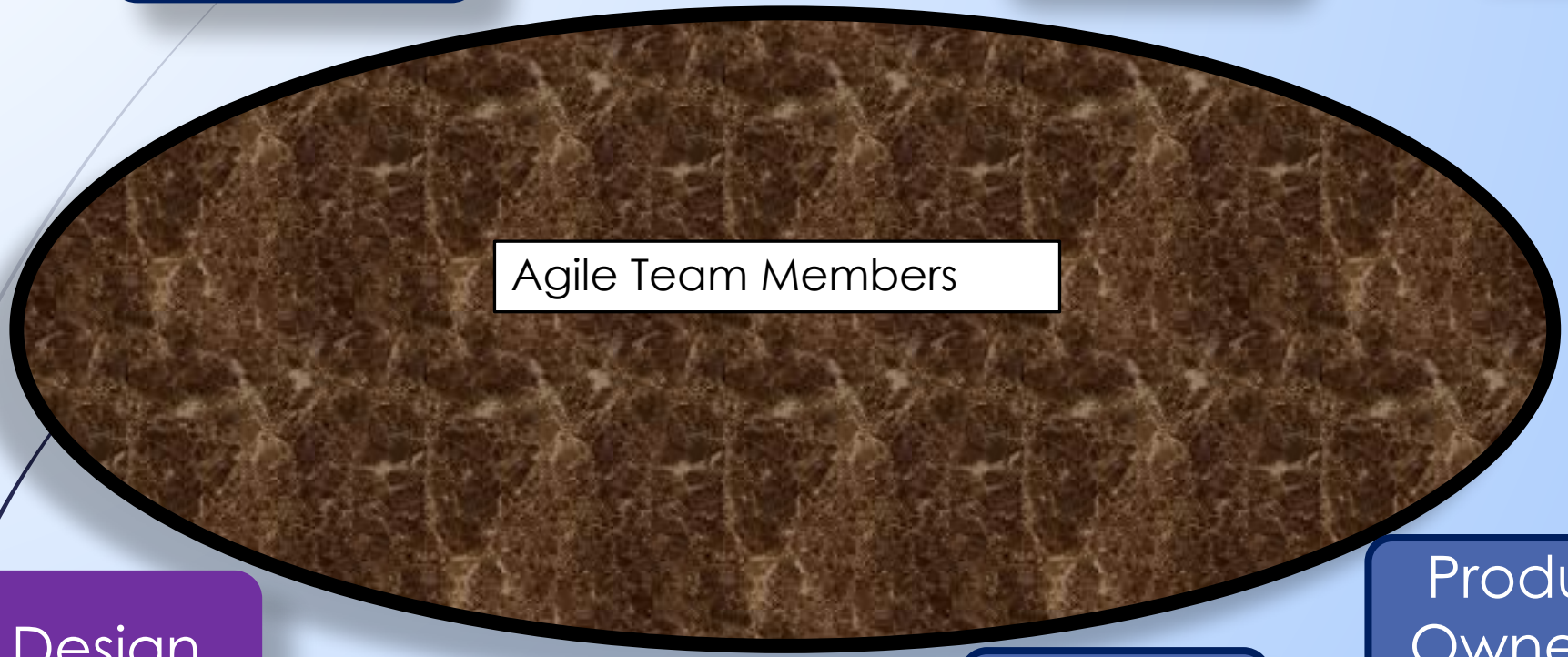
The scrum team

Dev: Tester= 2:1

Developer

Tester

- **Manager**
- **Lead**
- **Automation Tester**
- **Manual Tester**



Scrum Master=1

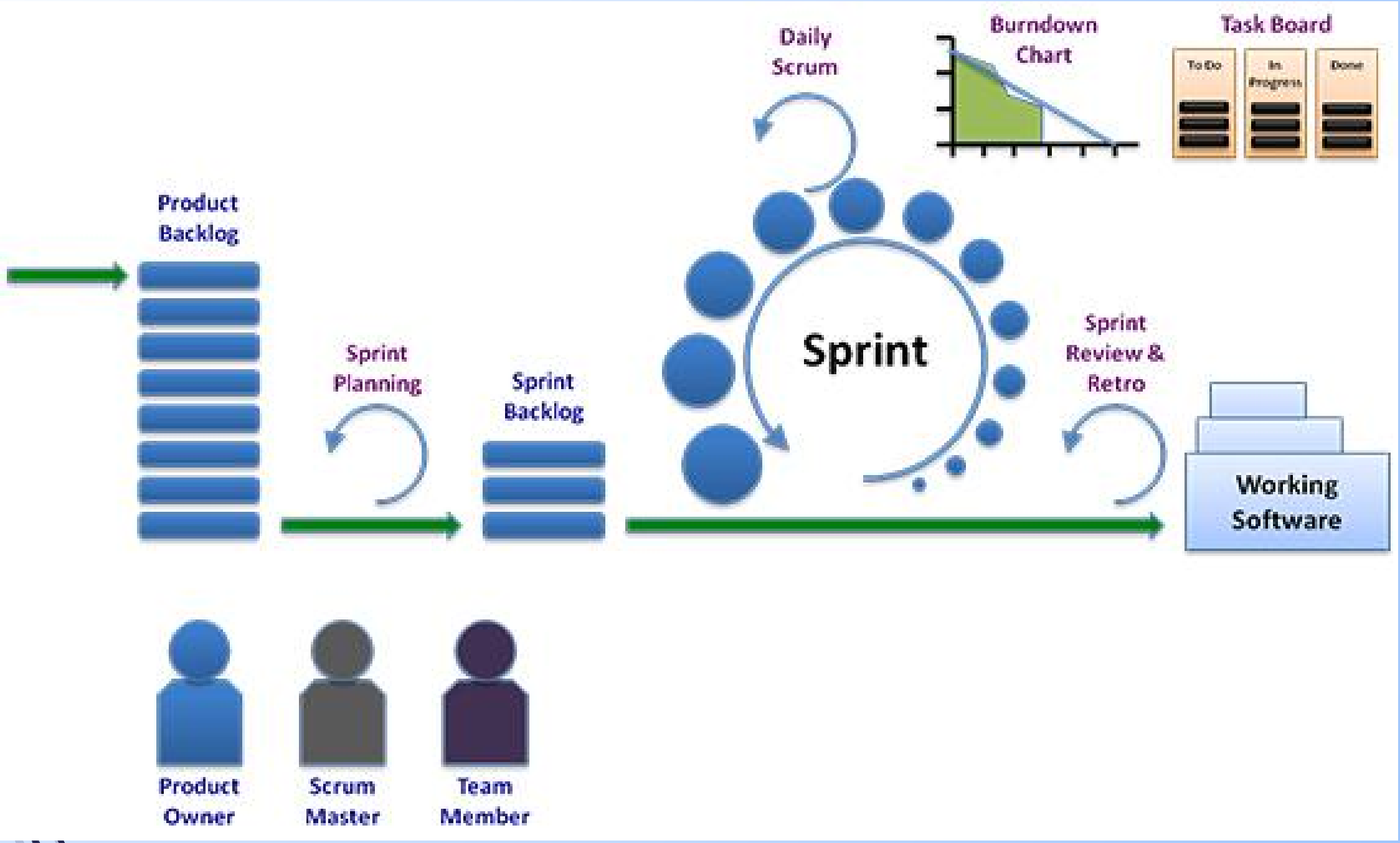
Product Owner=1

Design

ETL/DBA

BA=1

Project manager ⟶ Head of Project


Daily
Scrum
Burndown
Chart
Task Board
To Do
In
Progress
Done
Product
Backlog
Sprint
Planning
Sprint
Backlog
Sprint
Sprint
Review &
Retro
Working
Software
Product
Owner
Scrum
Master
Team
Member

PBI=Product back lock Items

- All requirement
  - High
  - Medium
  - Low
- PO=Product owner responsible to tell PBI with ranking as High>medium>low

- Break each PBI into Tasks / Items:
- Task 1: Design related------Assign to One in deign team
  - Task 2: Development related-------Assign to one Dev
  - Task 3: Data analysis----------Assign to ETL
  - Task 4: Testing----------Assign to tester

Sprint= Time frame to complete certain PBI or requirement or Epic

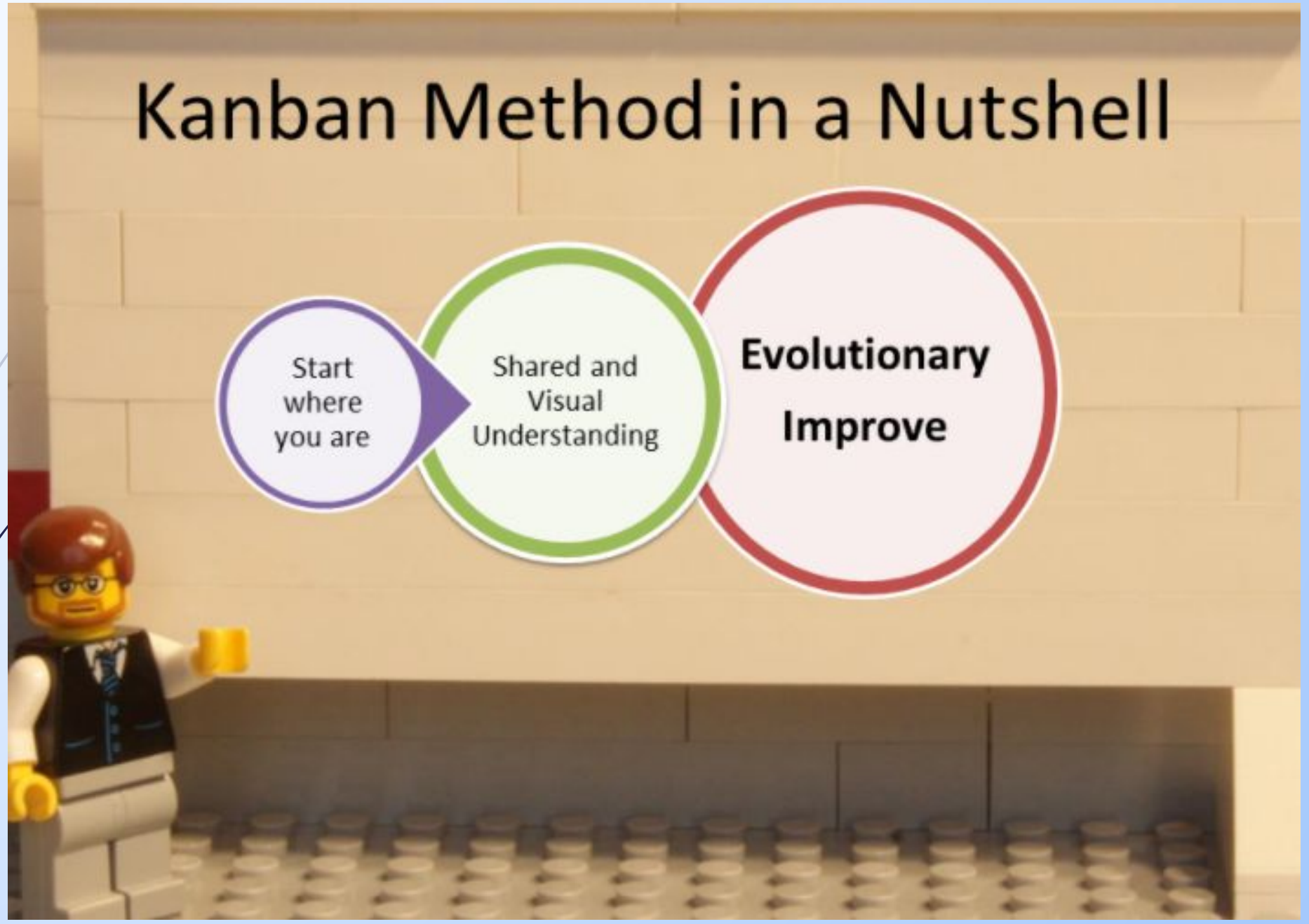
- Standard sprint 2 to 4 wks
- Usually take 2/3 PBI in one sprint
- How to select number/Story point of PBI=>
  - By poker play by Agile team member
- Which PBI to be taken in sprint=> usually take High level PBI
  - By a meeting with
    - SM
    - PO/BA
    - All manager & Lead

Meeting in Sprint

- **Sprint planning-** sprint start with this meeting
- **Sprint review/Retrospective meeting-** sprint end with this meeting
  - **If 2 wks sprint ::**
    - **Sprint planning- On Monday**
    - **Sprint review/Retrospective meeting- On Friday**
- **Daily Scrum – every day 15 mints**
- **Mid-sprint meeting – may or may not => middle of two wks**

Agile framework

- Kanban
- Scrum, or
- your own unique flavor

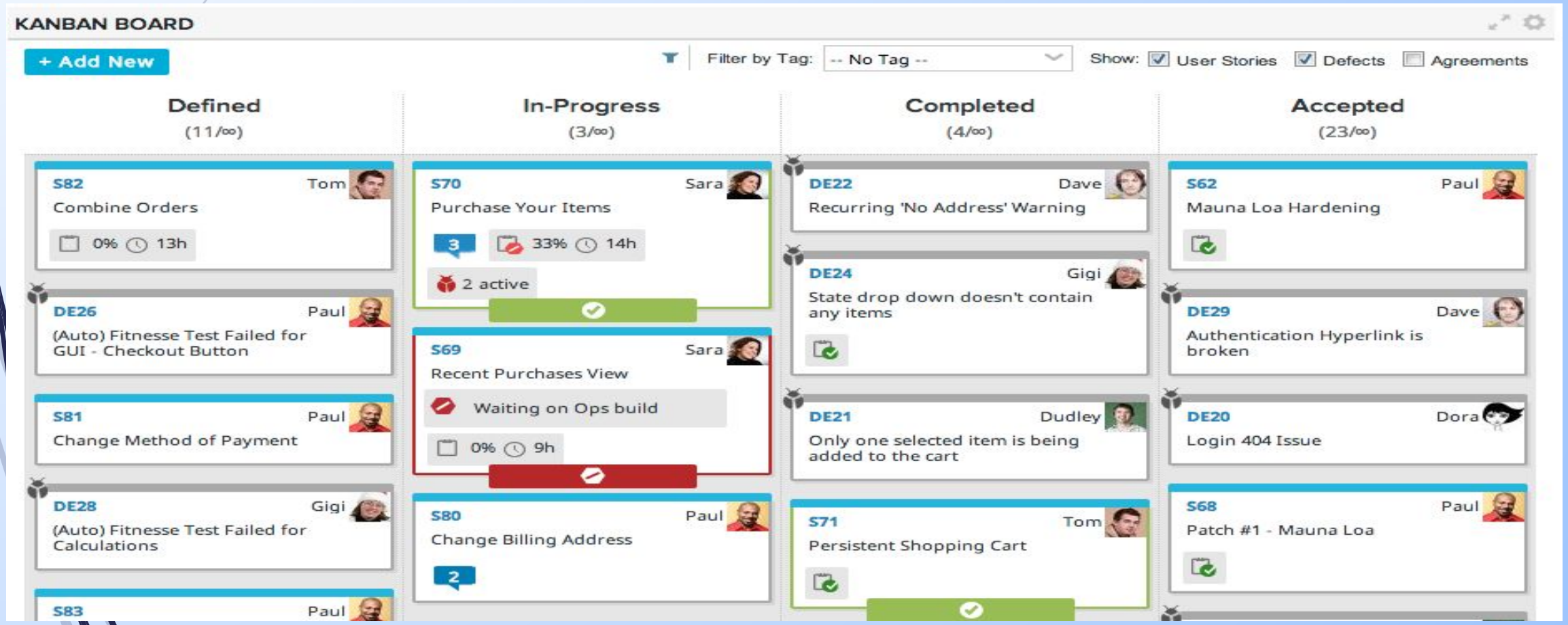

Kanban Method in a Nutshell
Start where you are
Shared and Visual Understanding
Evolutionary
Improve

Kanban framework in Agile

- **Kanban :** Japanese manufacturing system –**TOYOTA**: regulated through the use of an instruction card sent along the production line(1940).
- Agile software development
  - same "just in time" (or JIT) principles by matching
  - the amount of work in progress (WIP) to the team's capacity.
- Why Kanban?? gives teams more
  - flexible planning options,
  - faster output,
  - clearer focus, and
  - transparency throughout the development cycle.

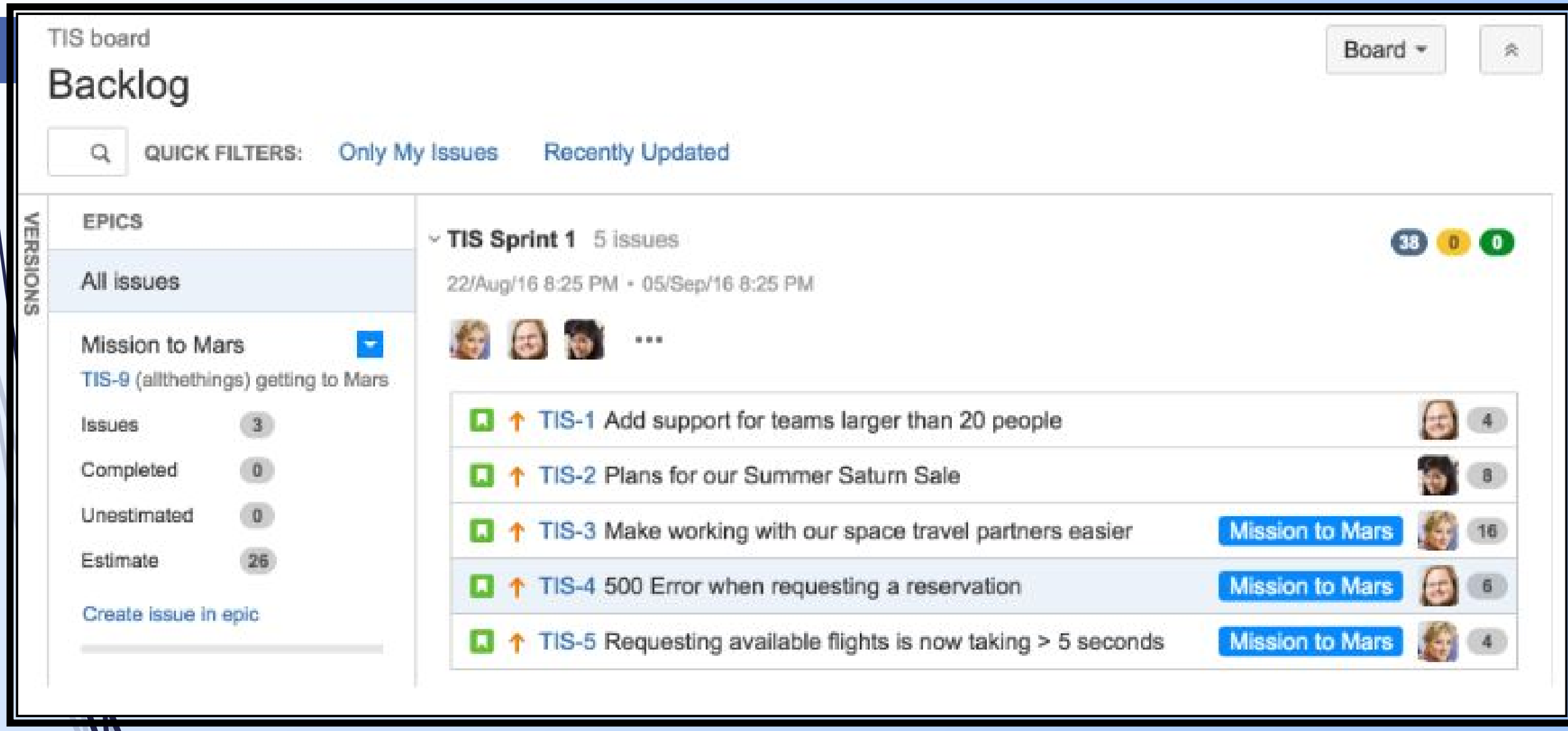
Kanban cards

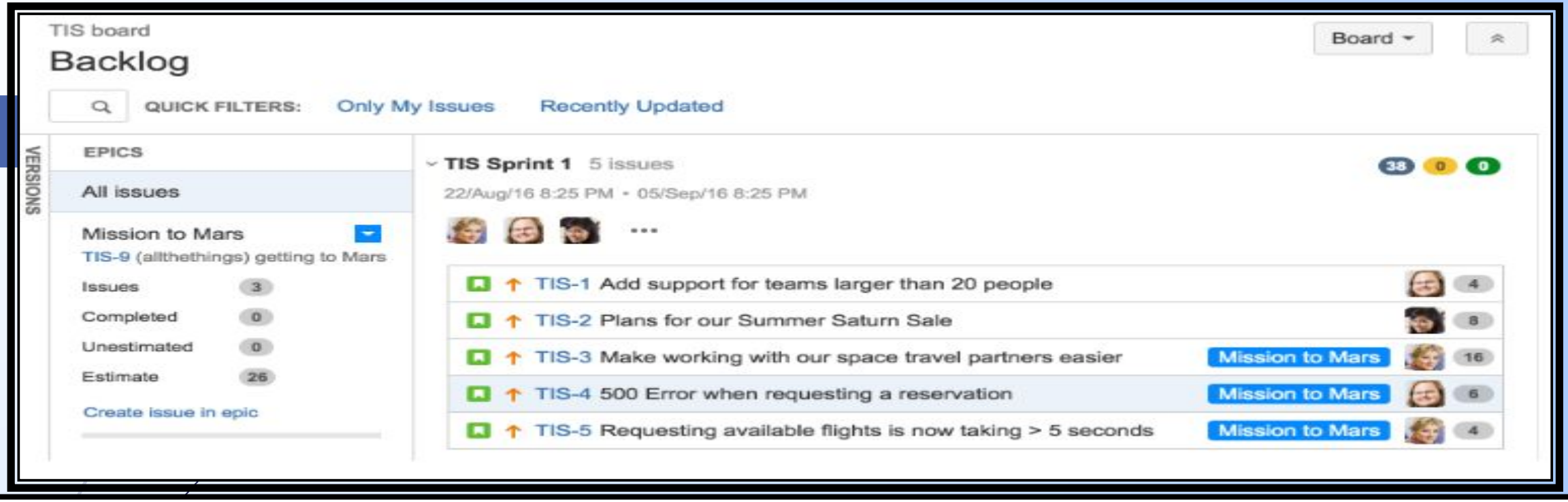
- In Japanese, kanban literally translates to "visual signal."
- For kanban teams, every work item is represented as a separate card on the board.

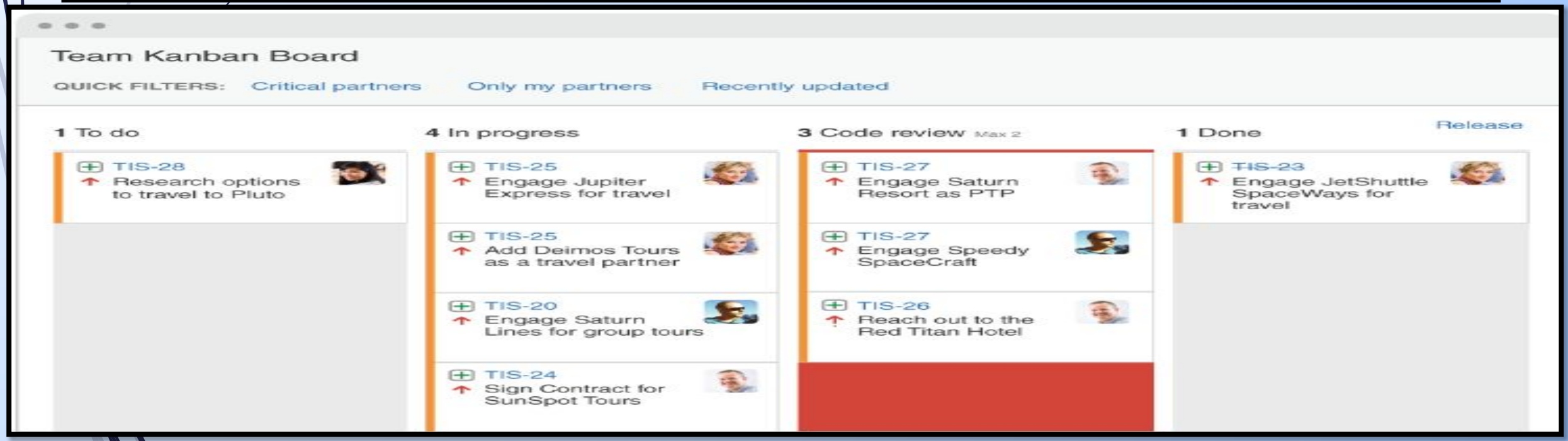


Scrum framework in Agile

- Scrum is the most popular frameworks for agile.
- So popular, in fact, that many people think scrum and agile are the same thing. (They're not.)
- With scrum, the product is built in a series of **fixed-length iterations** called **sprints** that give teams a framework for shipping software on a regular cadence.

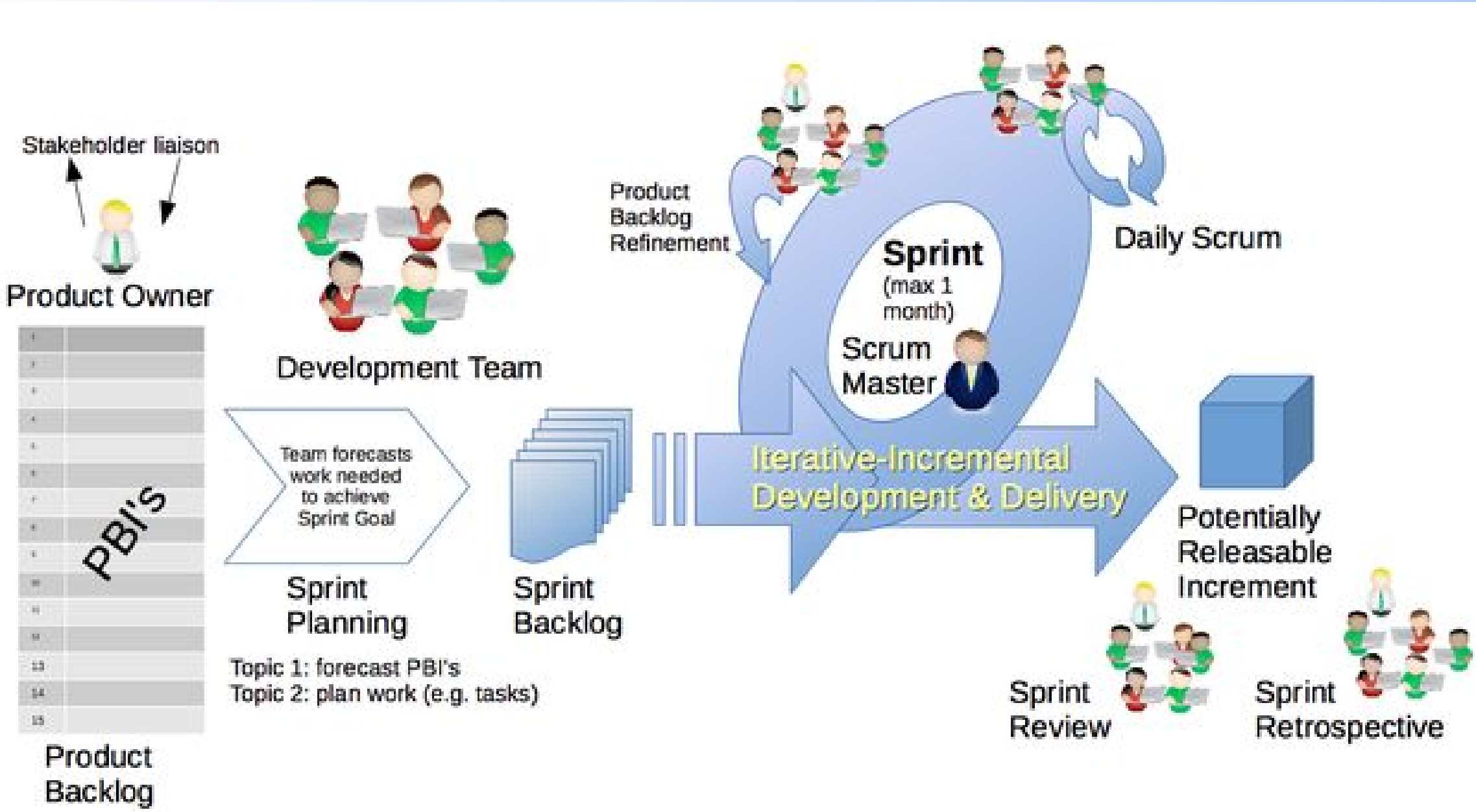

TIS board
Backlog
Board
QUICK FILTERS:
Only My Issues
Recently Updated
VERSIONS
EPICS
All issues
Mission to Mars
TIS-9 (allthethings) getting to Mars
Issues 3
Completed 0
Unestimated 0
Estimate 26
Create issue in epic
TIS Sprint 1 5 issues
38 0 0
22/Aug/16 8:25 PM • 05/Sep/16 8:25 PM
TIS-1 Add support for teams larger than 20 people 4
TIS-2 Plans for our Summer Saturn Sale 8
TIS-3 Make working with our space travel partners easier Mission to Mars 16
TIS-4 500 Error when requesting a reservation Mission to Mars 6
TIS-5 Requesting available flights is now taking > 5 seconds Mission to Mars 4


TIS board
Backlog
Board
QUICK FILTERS:
Only My Issues
Recently Updated
VERSIONS
EPICS
All issues
Mission to Mars
TIS-9 (allthethings) getting to Mars
Issues 3
Completed 0
Unestimated 0
Estimate 26
Create issue in epic
TIS Sprint 1 5 issues
22/Aug/16 8:25 PM - 05/Sep/16 8:25 PM
38 0 0
TIS-1 Add support for teams larger than 20 people 4
TIS-2 Plans for our Summer Saturn Sale 8
TIS-3 Make working with our space travel partners easier Mission to Mars 16
TIS-4 500 Error when requesting a reservation Mission to Mars 6
TIS-5 Requesting available flights is now taking > 5 seconds Mission to Mars 4

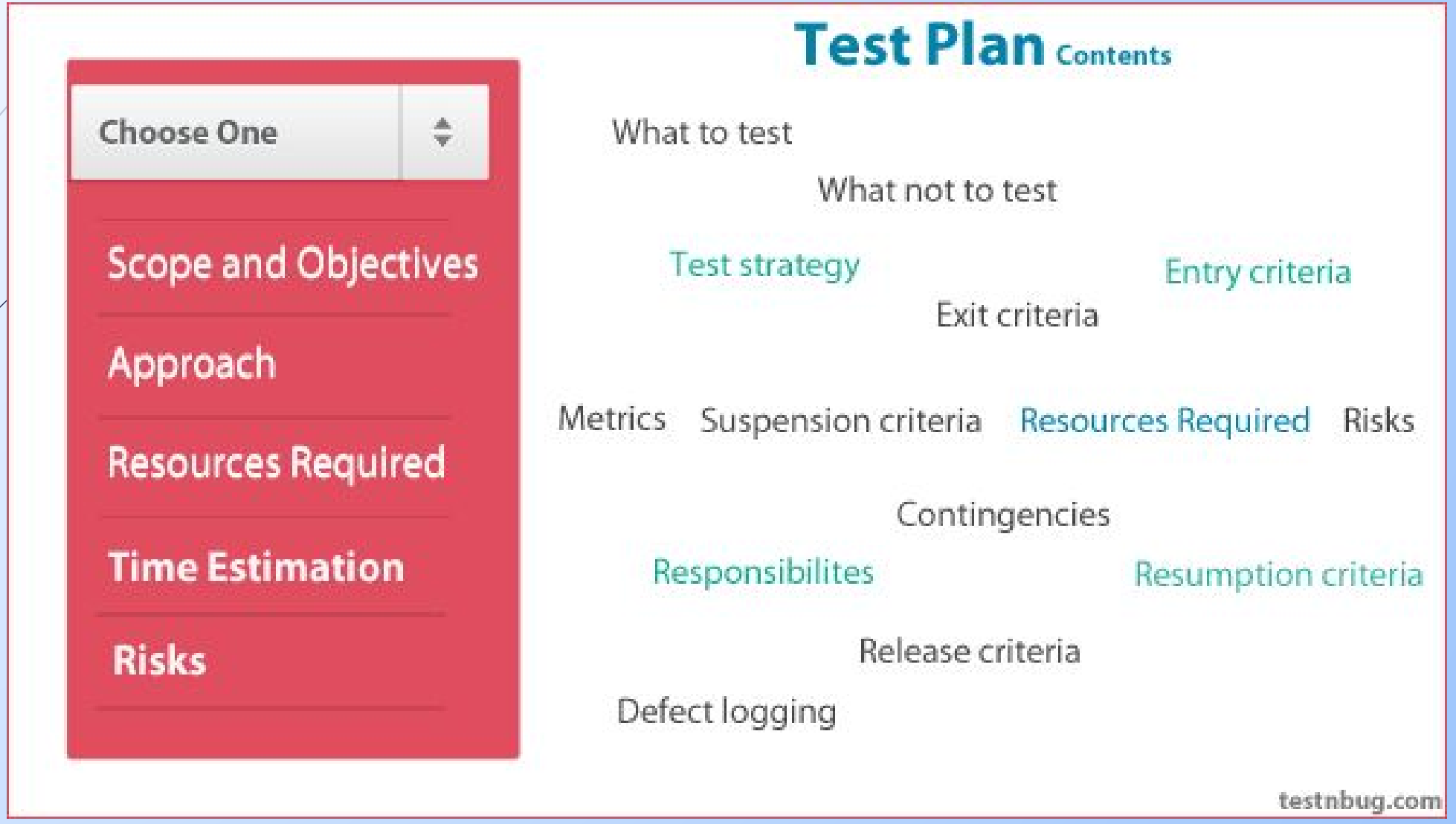

Team Kanban Board
QUICK FILTERS:
Critical partners
Only my partners
Recently updated
1 To do
TIS-28
Research options to travel to Pluto
4 In progress
TIS-25
Engage Jupiter Express for travel
TIS-25
Add Deimos Tours as a travel partner
TIS-20
Engage Saturn Lines for group tours
TIS-24
Sign Contract for 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
Release
TIS-23
Engage JetShuttle SpaceWays for travel

Scrum vs. Kanban

	SCRUM	KANBAN
Cadence/sequence/time frame	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint if approved by the product owner	Continuous delivery or at the team's discretion
Roles	Product owner, scrum master, development team	No existing roles. Some teams enlist the help of an agile coach.
Key metrics	Velocity	Cycle time
Change philosophy	Teams should strive to not make changes to the sprint forecast during the sprint. Doing so compromises learnings around estimation.	Change can happen at any time


Stakeholder liaison
Product Owner
Development Team
Product Backlog Refinement
Sprint
(max 1 month)
Scrum Master
Daily Scrum
PBI's
Team forecasts work needed to achieve Sprint Goal
Sprint Planning
Sprint Backlog
Iterative-Incremental Development & Delivery
Potentially Releasable Increment
Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)
Product Backlog
Sprint Review
Sprint Retrospective

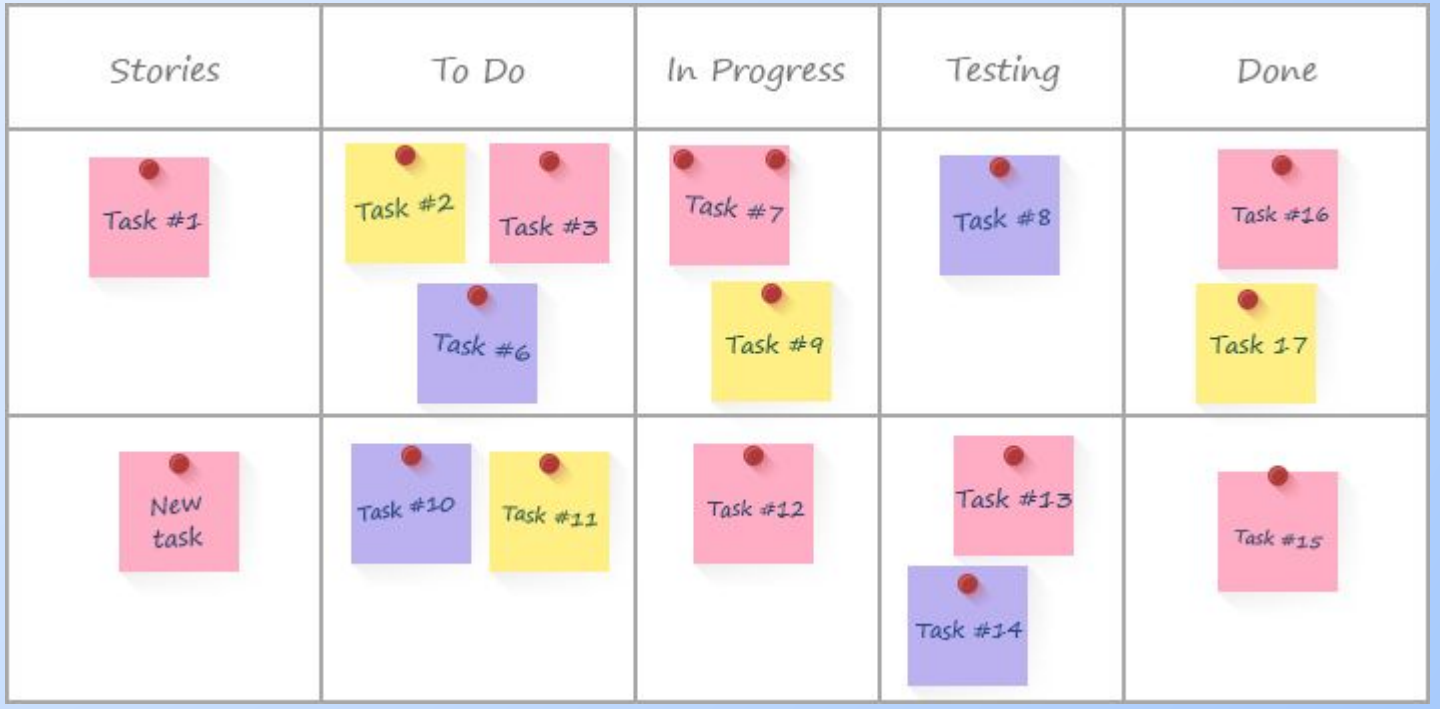
Test Plan


Choose One
Scope and Objectives
Approach
Resources Required
Time Estimation
Risks
Test Plan Contents
What to test
What not to test
Test strategy
Entry criteria
Exit criteria
Metrics
Suspension criteria
Resources Required
Risks
Contingencies
Responsibilites
Resumption criteria
Release criteria
Defect logging
testnbug.com

PBI	Sprint backlog	Roll over
All requirements in a project	Requirements/PBI: those taken in a sprint	Leftover work from one sprint to next sprint

Time Estimation in Sprint

- If 2 wk sprint:
  - Total 10 days:: 7 hrsx10=70 hrs
  - Estimate time for each PBI task or item by your self



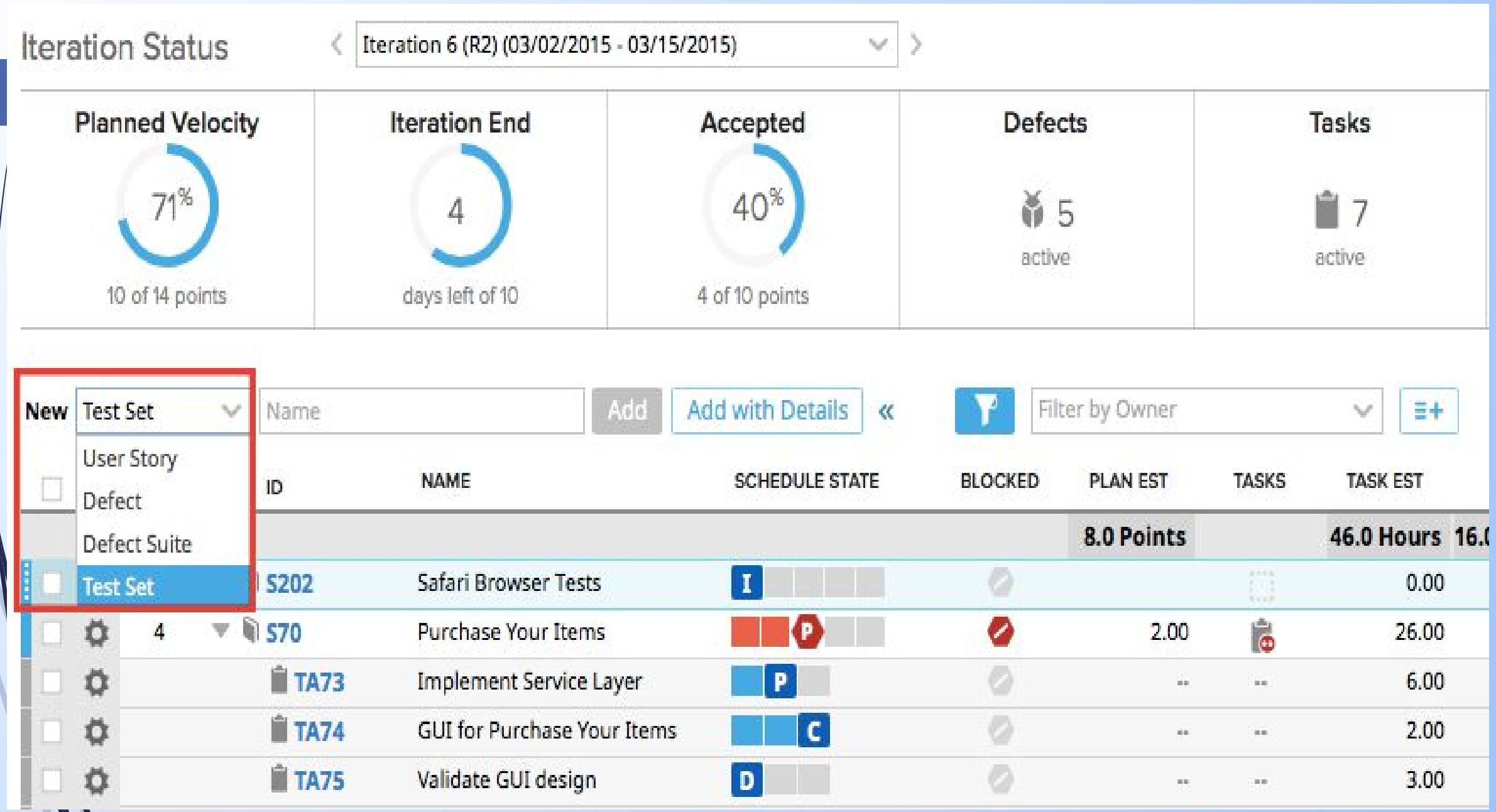
How to measure productivity of team members working in Agile Scrum?

- Burn down chart
- Velocity matrix

Agile velocity

- Velocity is a metric that predicts how much work an Agile software development team can successfully complete within a two-week sprint (or similar time-boxed period).

- estimating how fast work can be completed and how long it will take to complete a project


Iteration Status
Iteration 6 (R2) (03/02/2015 - 03/15/2015)
Planned Velocity
71%
10 of 14 points
Iteration End
4
days left of 10
Accepted
40%
4 of 10 points
Defects
5
active
Tasks
7
active
New
Test Set
User Story
Defect
Defect Suite
Test Set
Name
Add
Add with Details
Filter by Owner
ID
NAME
SCHEDULE STATE
BLOCKED
PLAN EST
TASKS
TASK EST
8.0 Points
46.0 Hours
S202
Safari Browser Tests
0.00
4
S70
Purchase Your Items
2.00
26.00
TA73
Implement Service Layer
6.00
TA74
GUI for Purchase Your Items
2.00
TA75
Validate GUI design
3.00

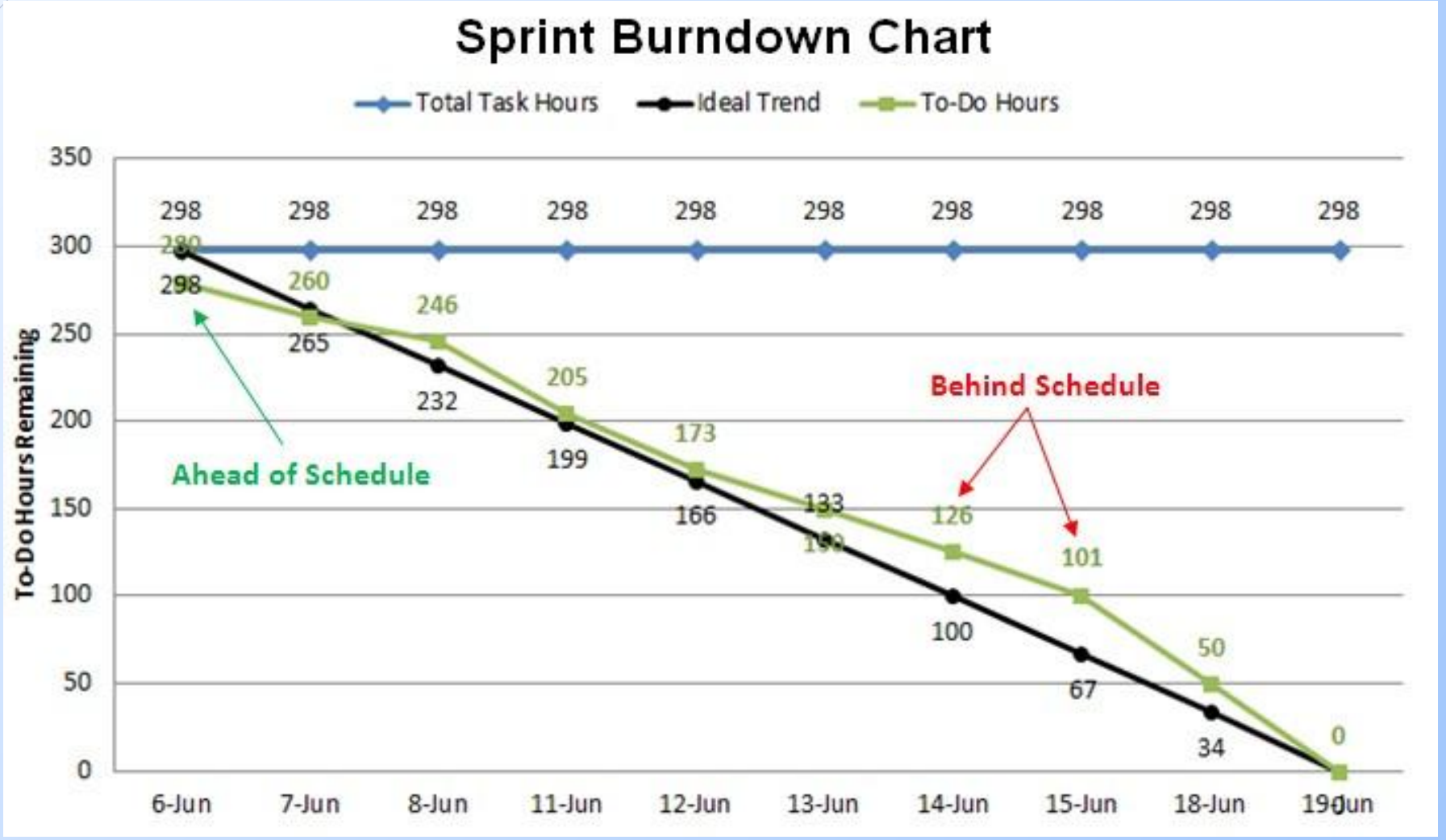
How to calculate Agile Velocity

- The metric is calculated by reviewing work the team successfully completed during previous sprints;
- for example, if the team completed 10 stories during a two-week sprint and each story was worth 3 story points,
- then the team's velocity is 30 story points per sprint.

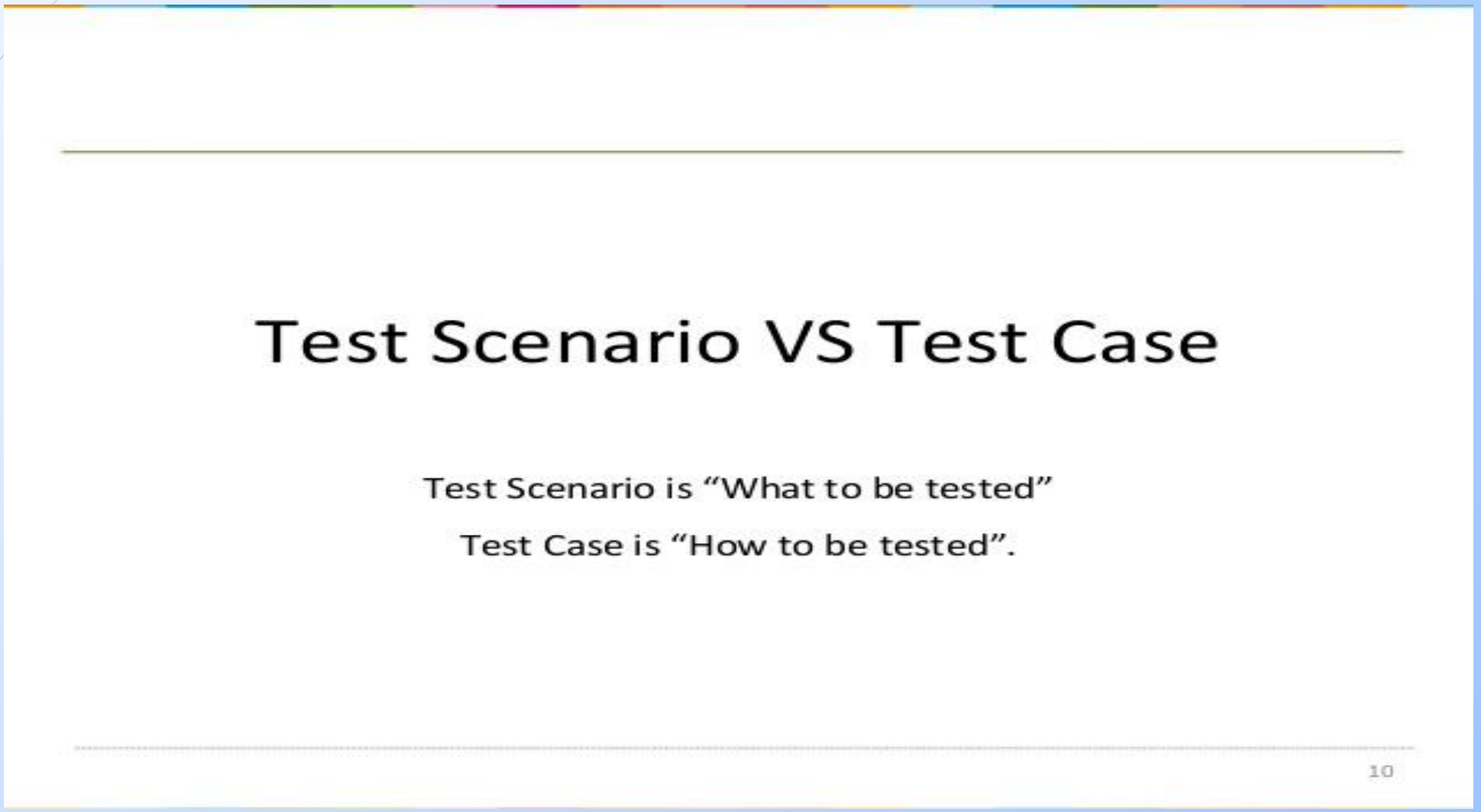
Velocity Tell when the project will finish

- Generally, velocity remains somewhat constant during a development project, which makes it a useful metric for estimating how long it will take a team to complete a software development project.
- If the product backlog has 300 story points, and
- the team is averaging 30 story points per sprint,
- it can be estimated that team members will require 10 more sprints to complete work.
- If each sprint lasts two weeks, then the project will last 20 more weeks.

How to check Team status in Sprint:



What is what??


Test Scenario VS Test Case
Test Scenario is “What to be tested”
Test Case is “How to be tested”.
10

Scenario ID	Test case ID	Test case	Test case Description	Test Step	Test step Description	Expected Result	Actual Result	Status	Comment
TS001	TC001	Validate Login Credentials	Test the login functionality of the e-commerce site to make sure that registered user is allowed to login into site using valid credentials.	Pre-Condition	1) Make sure that site under test is available and testable. 2) Make sure that required data for login is available.				
				Step 1	Launch the ecommerce application with the givn URL: <Test Site URL>	The e-commerce site launched properly.	Site launched successfully.	Pass	
				Step 2	Navigate to Login page	Login page is displayed to user with Username and Password fields are displayed on the page.	Login page loaded successfully.	Pass	
				Step 3	Enter valid Username in username field.	Username field should be editable and accept the username.	Usrename input accepted	Pass	
				Step 4	Enter valid Password in Password field.	Password field should be editable and accept the password and display as star or dot.	Password input displayed in dot and accepted	Pass	
				Step 5	Click on login button.	User should login into site and navigated to Home page.	User navigated Checkout page.	Fail	

Test Suite/Regression Suite

- Suite: Collection of test cases or test scenarios
- Regression Suite: Collection of regression Test cases/regression scenarios
    - We run test suite or reg suite –
        - in every release
        - If any feature update
        - If any major defect or bug fixed

Test Scenario vs. Test Case

- Scenario – it is a general description of a particular requirement/functionality
  - (Login, Sales order creation, Use Case).

Test Scenarios lead to the creation of test cases.

- **Test Case: (Granular)** Set of actions a tester will take to determine if the software is working the way it is supposed too.
  - Login with invalid ID
  - Login with invalid PW
  - Multiple login – same id –same machine
  - Multiple login – same id – different machine

Test Case ID	Description	Expected Result	Actual Result	Status
TC001	Open Browser	Browser open		
	Go https://www.facebook.com/	Facebook login page appear		
	Put valid name in username field	name field should editable and accept name		
	Put valid password in password field	password field should editable and accept password		
	Click OB button	OK button should be Enable		
	Validate login status	Home page appear		

How to write test case in BDD

- BD**D** is behavior driven development
- TD**D** is Test driven development
- ATD**D** is acceptance test driven development

  - **BDD focused on behavior or business logic or requirement**

Test case in BDD known as Feature file:

- A simple feature file consists of the following keywords/parts −
- **Feature** − Name of the feature under test.
- **Description** (optional) − Describe about feature under test/ requirement
- **Background**: Any common Steps
- **Scenario** − What is the test scenario.
- **Given** − Pre-Action(open browser or put URL)
- **When** − All Action(put user name & pass, OK button)
- **Then** − After action or validation(Successfully enter home page)
- **And /or-** Extra steps

Cucumber with BBD- Gherkin Language

- **Feature**: Login Functionality Feature
- **Description:**
  - In order to ensure Login Functionality works- want to run the cucumber test to verify it is working.
  - Password must be 8 char with one number and one symbol
- **Background:** usually not use in normal scenario

- **Scenario:** Validate Login Functionality
- **Given** user able to open browser
- And Navigate to application
- **When** user logs in using <Username>
- | Username |
- |sarower|
- And put <Password>
- | Password |
- |PASSWORD2$|
- And Click OK Button
- **Then** login should be successful

JIRA

Download from = https://www.atlassian.com/software/jira/try

ATLASSIAN

Try our products in the Cloud, free for 7 days

72% of our customers combine our products

Jira Software

From $10 /month

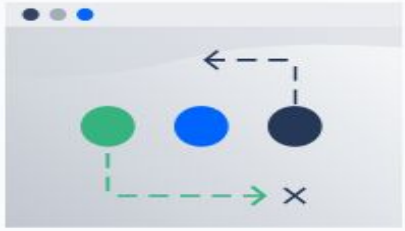
Plan, track, and release world-class software

Try it free

MOST POPULAR

Jira Software + Documentation

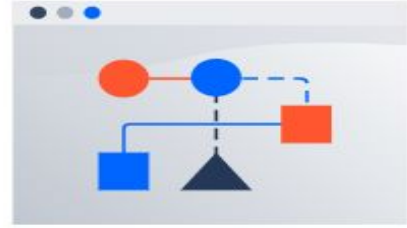
From $20 /month



Centralize and share ideas to supercharge development

Jira Software + Help Desk

From $30 /month



Create, build, and support software, from end to end

Try it free

Get started with JIRA Software

You're signing up for our Cloud option, hosted by us.

Looking to host it on your own server?

Claim your site

sarower .atlassian.net

Full name

Sarower Ahmmed

Email

studentttech@gmail.com

Password Show password

••••••••

Password strength: Good

By clicking here you accept the Atlassian Customer Agreement.

Start now

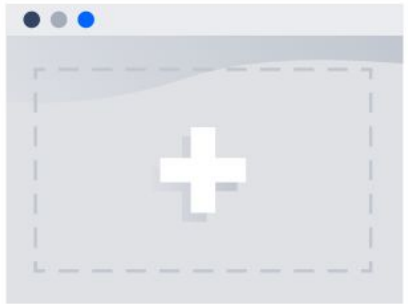
Welcome!

Choose an option below to get started.

See a project in action

Create and explore a project that's pre-loaded with sample data.

Create sample project



Create a new project

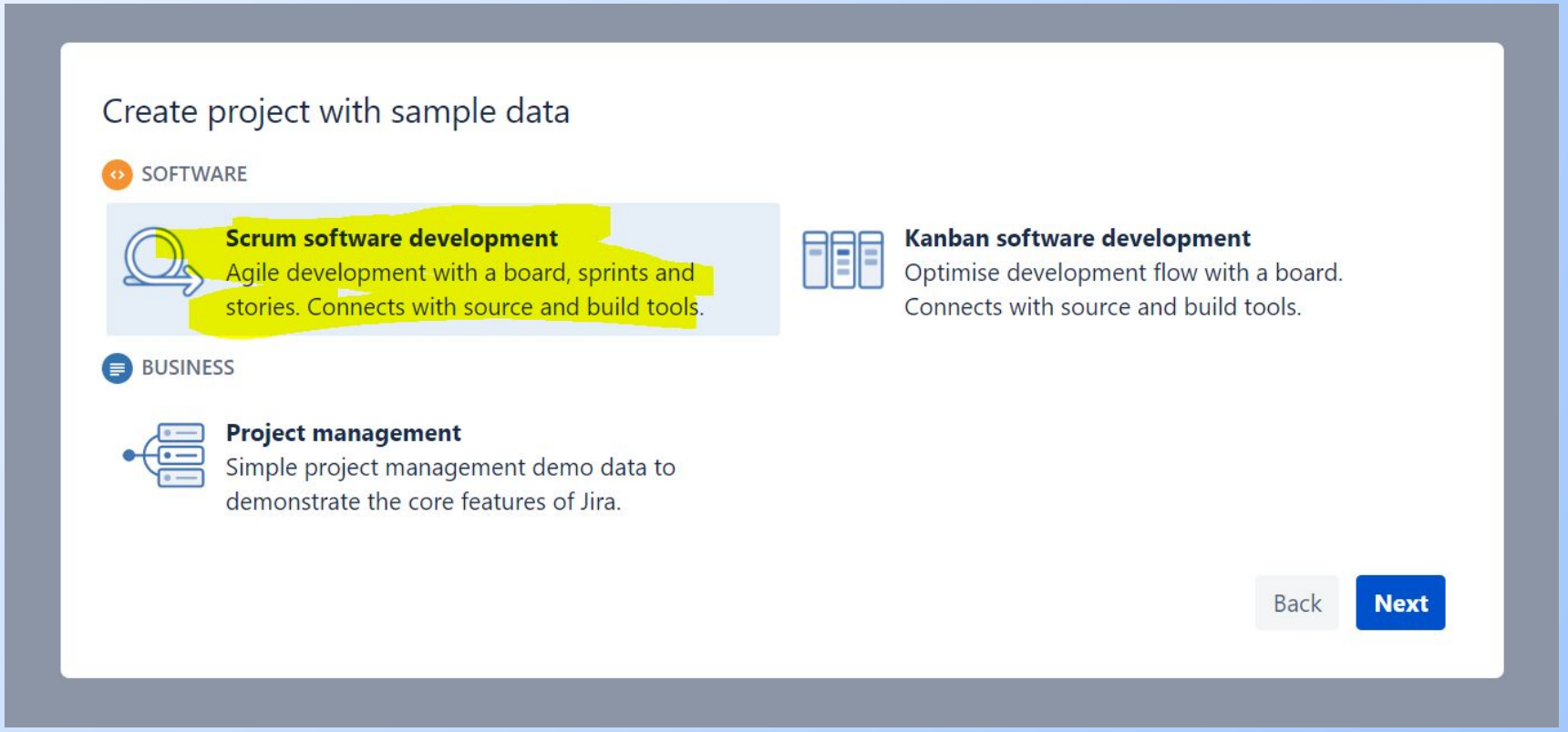
No time to waste! Get stuck in and create your first project in Jira.

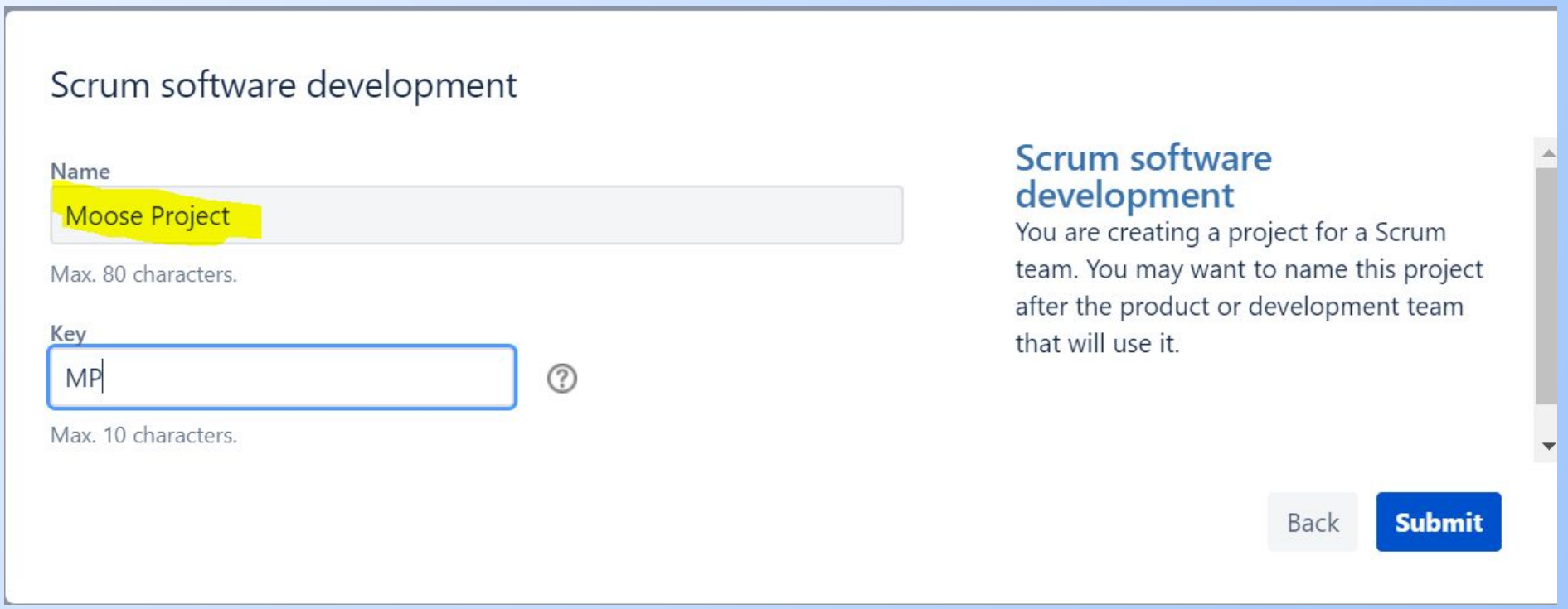
Create new project

Import from another tool

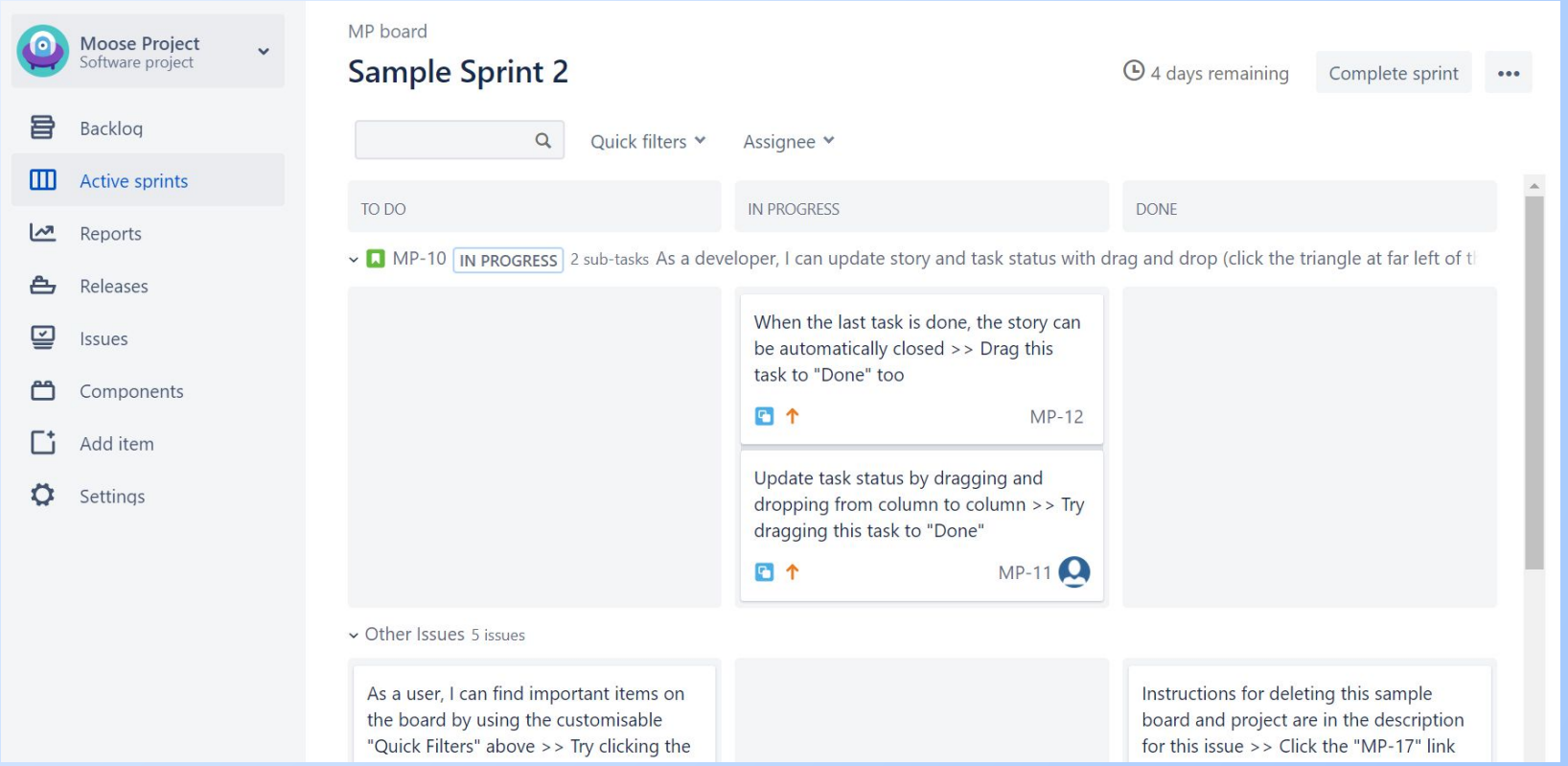
Switching to Jira from another issue tracker? Quickly import your issues.

Import issues

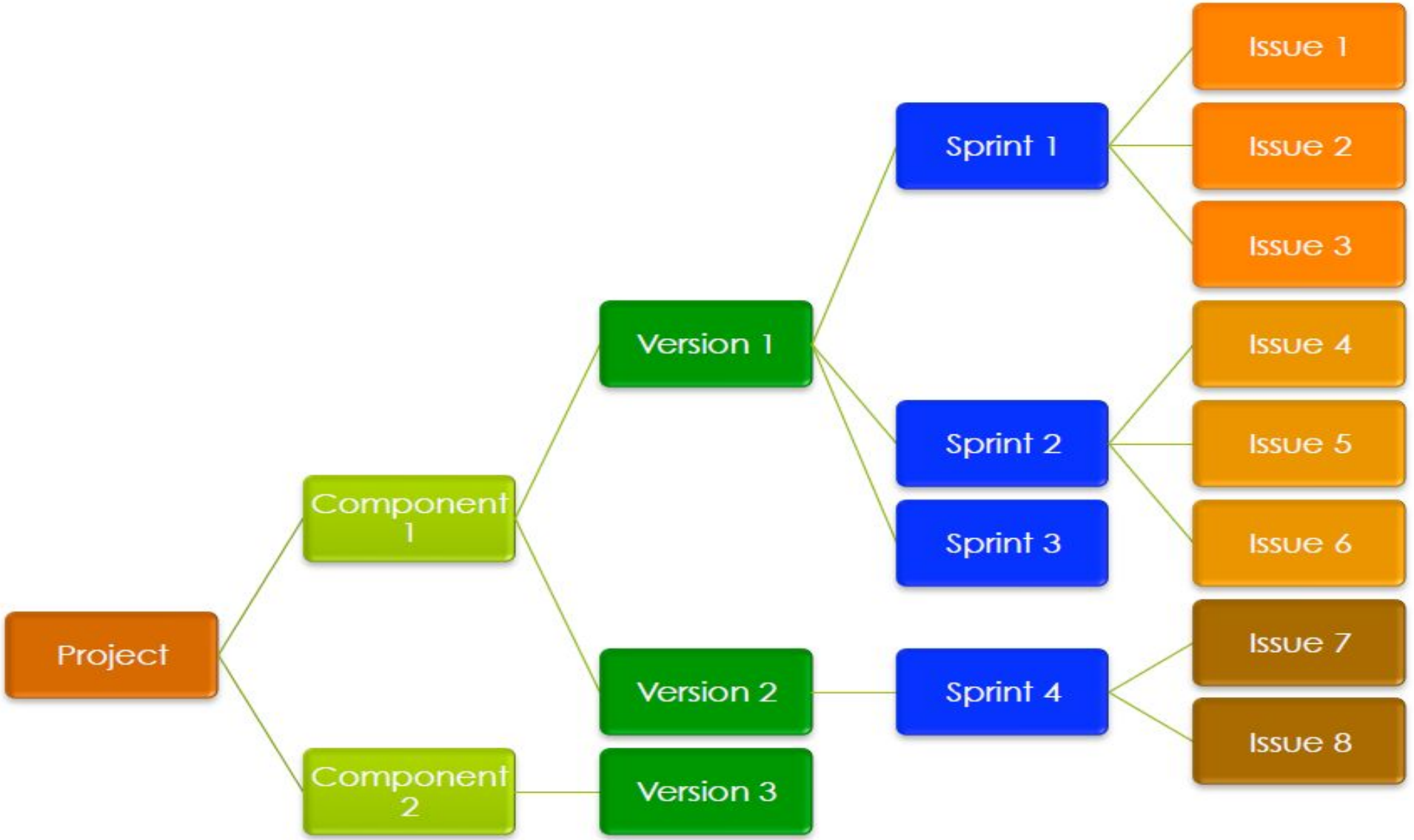

Create project with sample data
SOFTWARE
Scrum software development
Agile development with a board, sprints and stories. Connects with source and build tools.
Kanban software development
Optimise development flow with a board. Connects with source and build tools.
BUSINESS
Project management
Simple project management demo data to demonstrate the core features of Jira.
Back
Next


Scrum software development
Name
Moose Project
Max. 80 characters.
Key
MP
Max. 10 characters.
Scrum software development
You are creating a project for a Scrum team. You may want to name this project after the product or development team that will use it.
Back
Submit

https://sarower.atlassian.net/secure/RapidBoard.jspa?rapidView=1&projectKey=MP

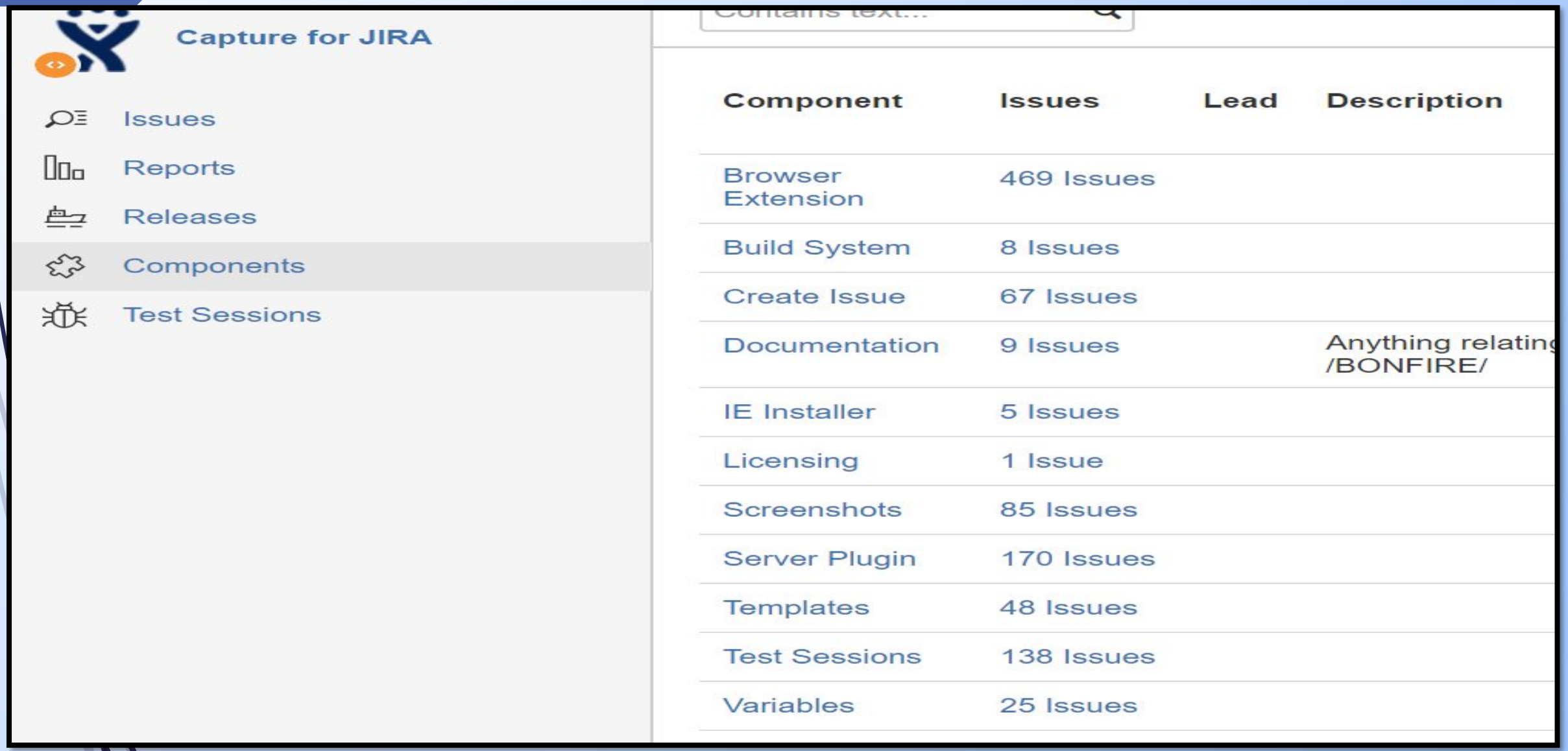


JIRA Structure



talentech

Component: divided project into small part



Issue Types

- **Issues**
  - Epic
  - Story
  - Bug
  - Improvement
  - Task
  - New Feature
  - Risk
- **Sub-Issues**
  - Technical Task
  - Sub-Bug

talentech

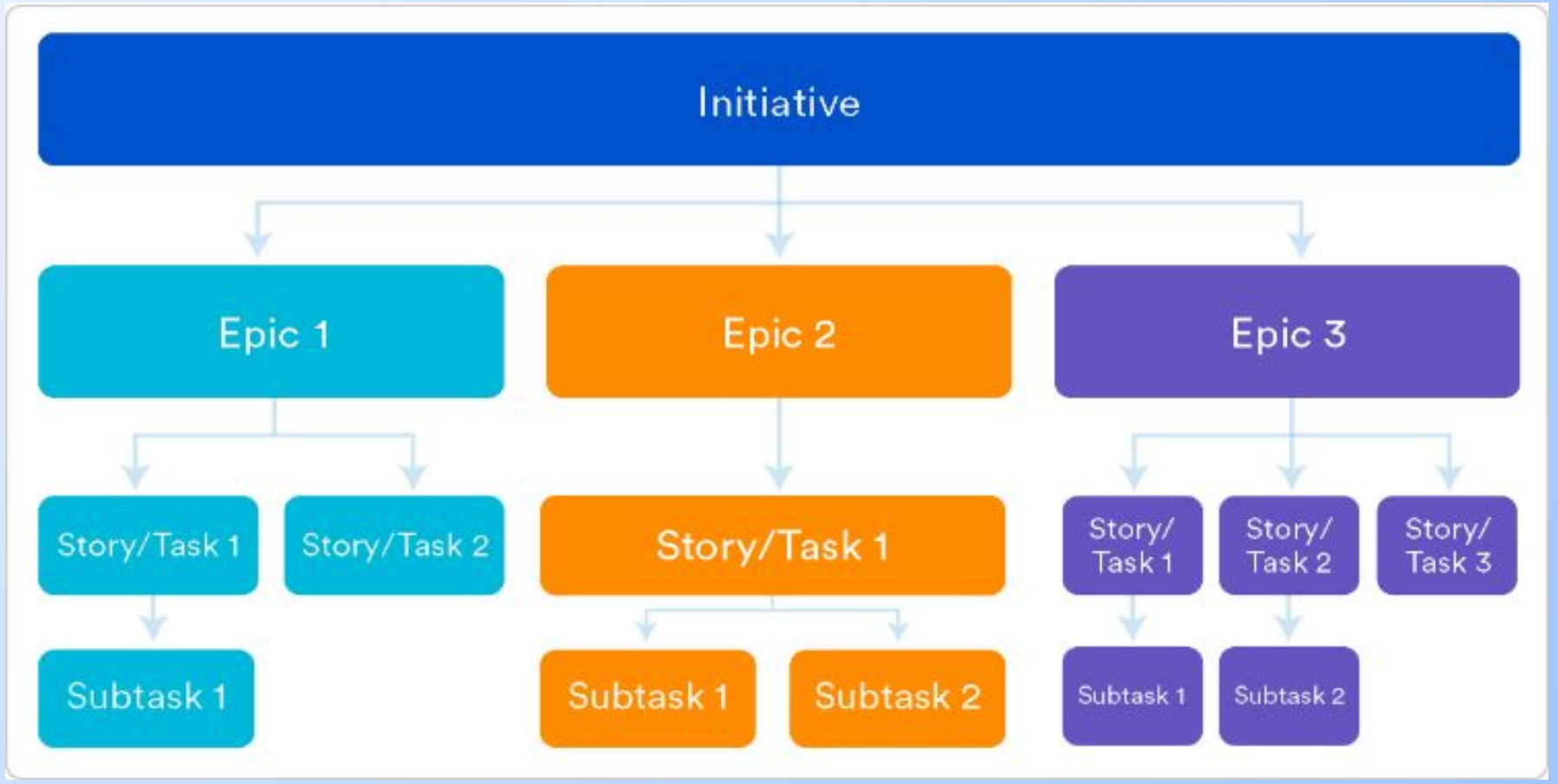
Epic>Story/task/item

Epic	Story	Version	Sprint
Large body of work, contains stories	Smallest unit of work, also known as a task	The release of software to the customer	Iteration where team does the work

Epic>Story/task/item…….version

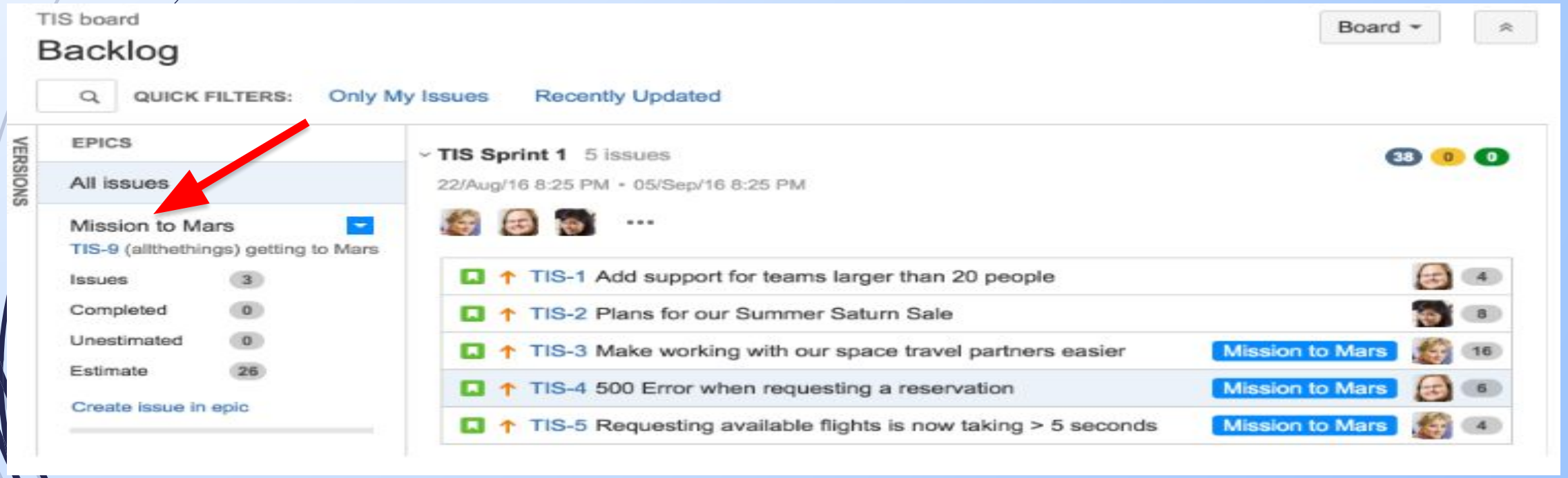
- **Epics** are larger bodies of work that stories roll up into more strategic goals or initiatives.
- Can span across multiple sprints and versions.

Epic vs Story



Example of an epic

- **Kanban:**
  - epics can be used as swim lanes to segment different streams of work.
- **Scrum,** epics can help label the work in your sprints,
  - like in the example below. **Mission to Mars is an epic** in this sprint. TIS 1, TIS 2, etc. are all user stories within the sprint (TIS Sprint 1).



What is a user story/story/task?

- The **user story** describes the type of **user**,
- what they want and
- why.

- A **user story** helps to create a simplified description of a requirement.

What is a version?-it's a piece of product or functions

- Versions are the actual releases of software out to customers.
- Versions are often developed over a set of sprints, much like epics.

- **Example of a version**
- A product owner may structure the release strategy as follows:
- Version 1: login, logout, password management
- Version 2: purchase history
- Version 3: saving preferences
- etc.

How to use agile metrics to optimize your delivery in JIRA?

- By Different JIRA Reports
- Sprint Burndown
- Epic and Release Burndown
- Velocity
- Control Chart
- Cumulative Flow Diagram
- Even more metrics!