Introduction to Agile and Scrum Framework

Agenda

Agile & Scrum Framework

- Introduction to Agile
- Agile Manifesto
- 12 principles of agile
- Introduction to Scrum Framework
- Scrum Ceremonies
- Scrum Artifacts
- Scrum Team

What is Agile?

Understanding Agile

History Of Agile

- The ability to create and respond to change in order to succeed in an uncertain and turbulent environment.
- Agile is a mindset describe by 4 Values and twelve Principle and manifested by many practices.

History Of Agile

- 1930s-1940s

  Walter Shewhart a quality expert at Bell Labs promoted “plan-do-study-act” (PDSA) . Later in 1940s Edward Deming another quality guru promoted this oppressively.

  Later in 1982 Thomas Glib applied the same in S/w Development

- 1950s-1960s

  Concept of Incremental and Iterative Development were used aggressively by Department of Defense and NASA on multiple projects. For example Project X-15, Mercury , Gemini and Apollo and many more

History Of Agile

1970s-1980s

- TRW used IID Missile Software, IBM FSD S/w for the Command Center for Submarine Barry Bohme promoted spiral model did formalize and make prominent the risk-driven-iterations concept

- Thomas Glib discussed his IID practice—evolutionary project management—and introduced the terms “evolution” and “evolutionary”. He suggested to implement complex project implemented with series of small steps and each step has clear measure of success condition

IID : **Iterative and incremental development (IID)**

History Of Agile

1970s-89s

- “The New New Product Development Game,” a whitepaper published by Takeuchi and Nonaka
- Takeuchi and Nonaka discuss the “rugby approach” of dedicated, self-organizing teams

1990s

- January 1994, a group of 16 rapid application development (RAD) practitioners met in the UK to discuss the definition of a standard iterative process to support RAD development. This methodology finally known as Dynamic System Development Method

History Of Agile

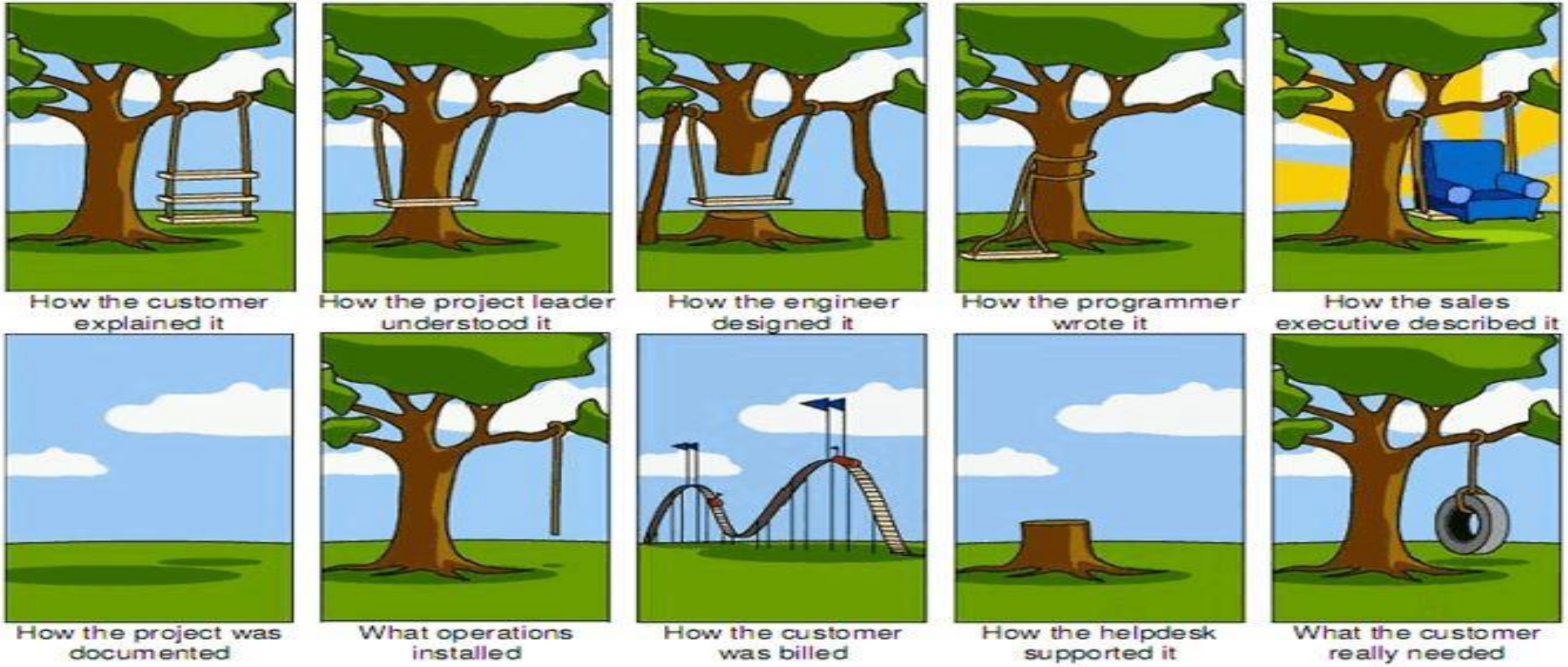
2001 Onward

- In February 2001, a group of 17 process experts—representing DSDM, XP, Scrum, FDD, and others—interested in promoting modern, simple IID methods and principles met in Utah to discuss common ground.
- Team come up with 4 agile Values and 12 Principles. These Agile Values and Principle that form the basis for Agile Software Development
- Scaled Agile Frame Work, Large Scale Scrum (LeSS), Scrum@Scale (SaS), Scrum of Scrums (SoS), Nexus, Disciplined Agile (DA), Spotify,

Why Agile?

Understanding Agile

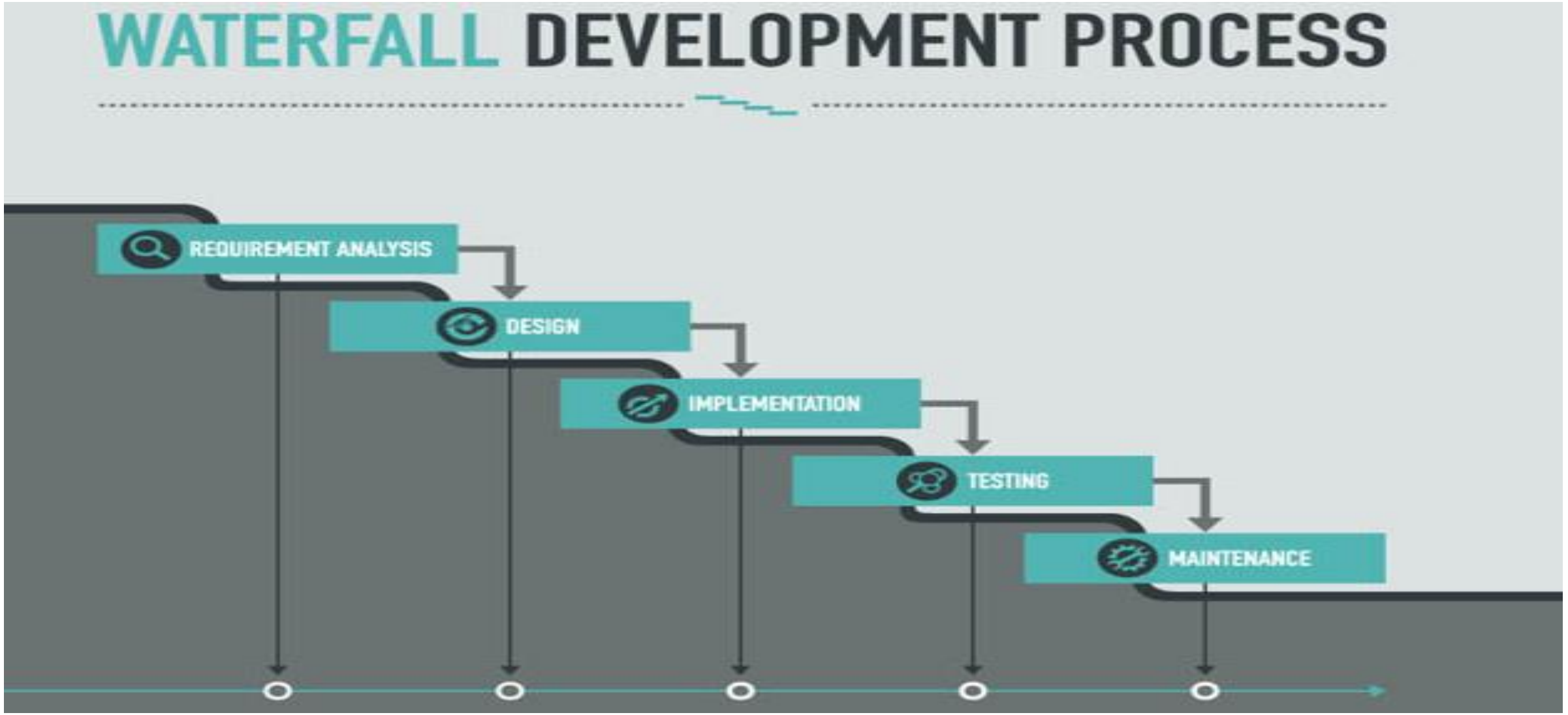
Why Agile


How the customer explained it
How the project leader understood it
How the engineer designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

Traditional Software Development Process

- Unable to get along with frequent changing market demand.
- Unable to cope up with race with competitor to launch their product 1st.
- Delayed time to market the Product
- Hinderance to respond to the change in the product requirement

Waterfall Model



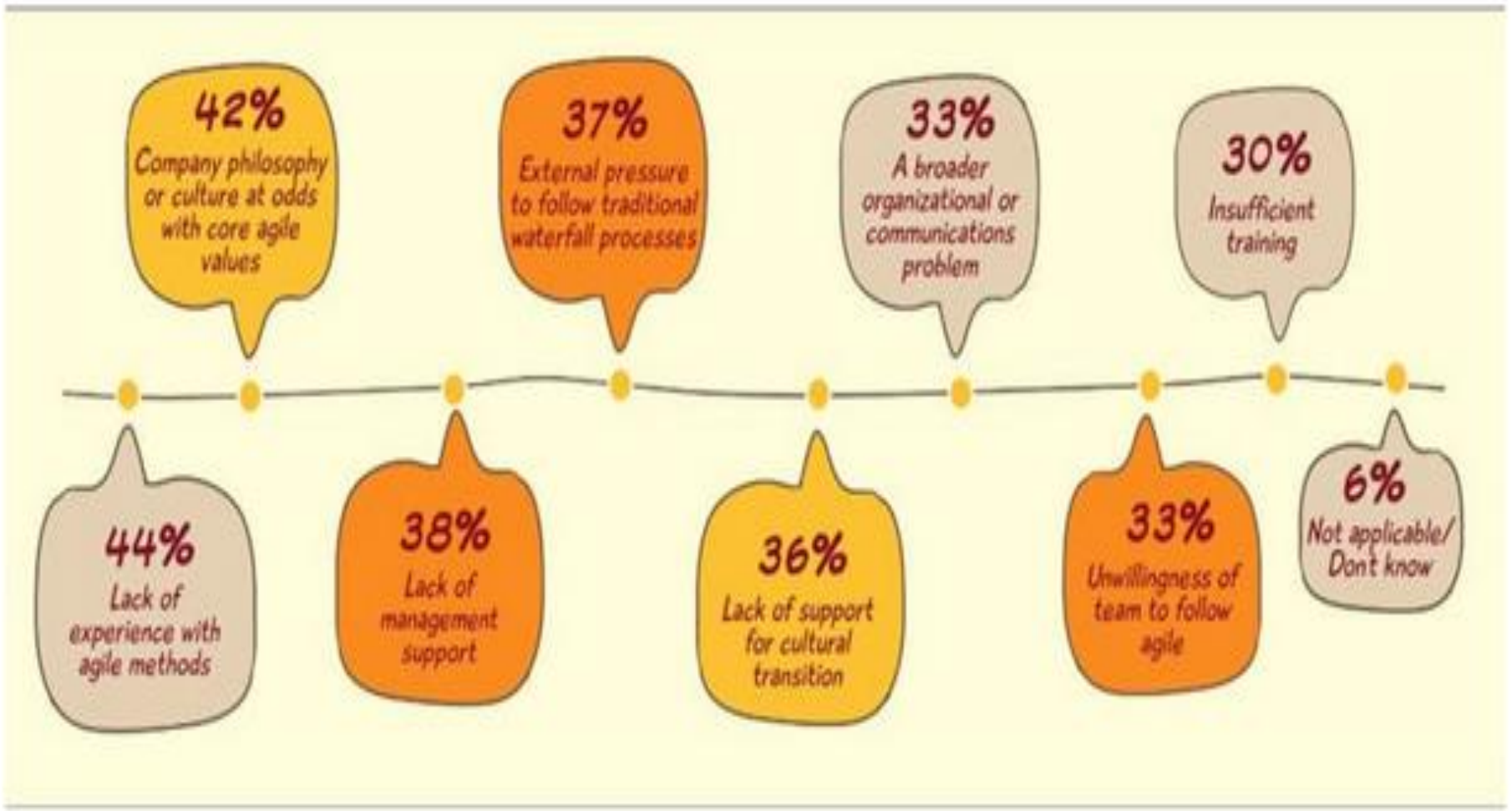
What's The Same?
Traditional S/w and Agile Development Process

- Vision
- Life cycle
- Requirements
- Schedule
- Team
- Communication mechanisms

Few points in favor of Agile

- Faster Time To Market
- Early ROI
- Feedback From Real Custer
- Build The Right Product
- Early Risk Reduction
- Better Quality
- Culture AND Morale
- Efficiency
- Customer Satisfaction
- Alignment
- Emergent Outcome
- Predictability

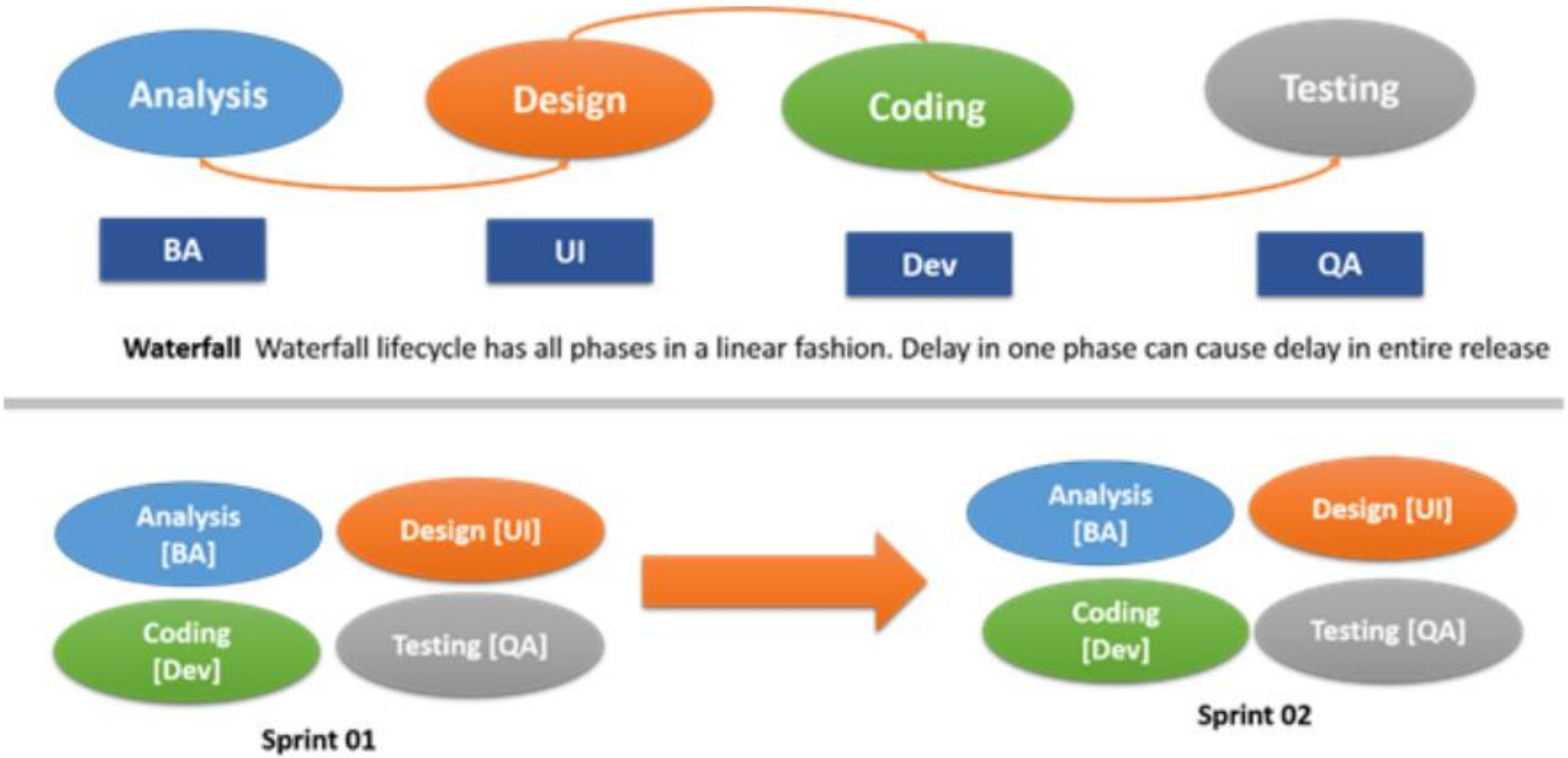
Failure of Agile



Incremental Delivery Over Big Bang Delivery

Traditional Vs Agile Triple Contraints

12 Principles of Agile



Waterfall vs Agile Scrum

Why Agile : Few questions to asks?

One should ask Five questions before adopting Agile:

- Q #1) Do you have the patronage from Top management?
- Q #2) Can you switch to Task culture?
- Q #3) What's the size and nature of your projects?
- Q #4) Do your customers and suppliers (developers) buy-in the notion?
- Q #5) How big is your project team?

Various Agile Framework

Understanding Agile

Various Agile Framework

- **Extreme Programming (XP)** is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.
- **Feature Driven Development (FDD)** is an agile framework that, as its name suggests, organizes software development around making progress on features. Features in the FDD context, though, are not necessarily product features in the commonly understood sense. They are, rather, more akin to user stories in Scrum. In other words, "complete the login process" might be considered a feature in the Feature Driven Development (FDD) methodology.
- Test-driven development (TDD) is a **development technique where you must first write a test that fails before you write new functional code**. TDD is being quickly adopted by agile software developers for development of application source code.
- **Dynamic System Development Method(DSDM)** considers the need to react quickly throughout the development of the product while incorporating the constraints often imposed by corporate cultures and processes. DSDM comprises of 8 principle: Focus on Business Need, Deliver on Time, Collaborate, Never Compromise Quality, Build Incrementaly from Firm Foundation,, Develop iteratively, Communicate continuously, Demonstrate control.
- **The Crystal agile framework** is for software development. It places focus on people over processes, to empower teams to find their own solutions for each project rather than being constricted with rigid methodologies.
- **Kanban i**s a framework that falls under the Agile methodology. It was developed in the late 1940s by a Japanese engineer named Taiichi Ohno. Agile Kanban Framework focuses on visualizing the entire project on boards in order to increase project transparency and collaboration between team members.

Agile Manifesto

Understanding Agile

Agile Manifesto

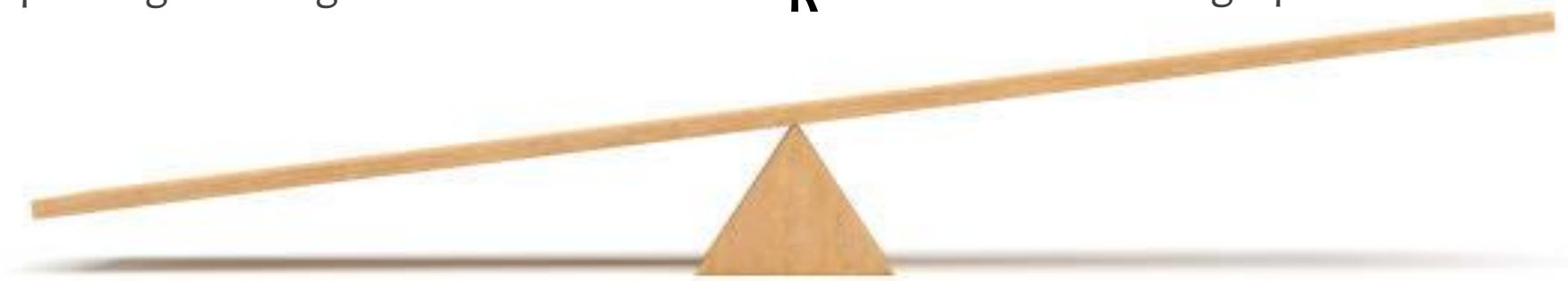
We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

4 Values of Agile

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

O
V
E
R

- Processes and tools
- Comprehensive documentation
- Contract negotiation
- Following a plan



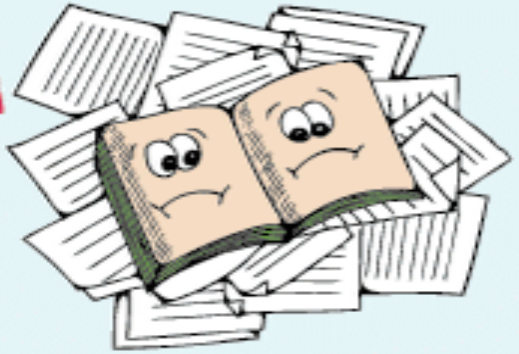
That is, while there is value in the items on the right, we value the items on the left more.

Manifesto for Agile Software Development*

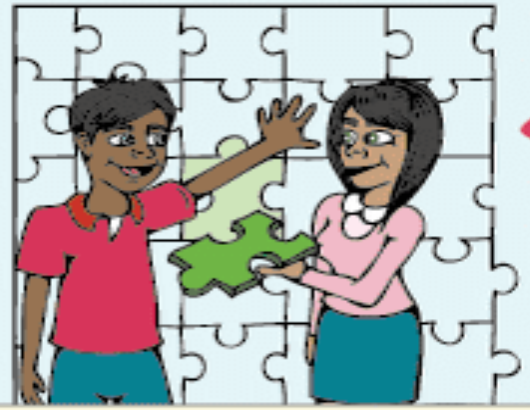
"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

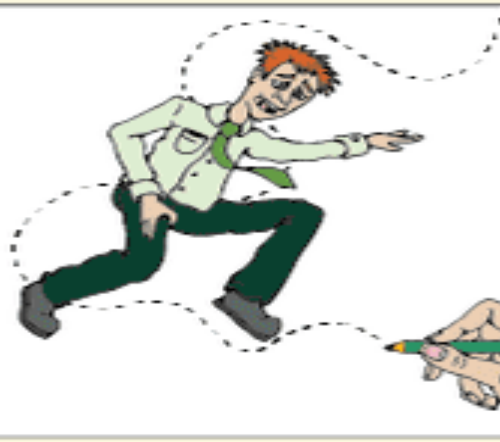
Individuals & interactions **Over** Processes & tools

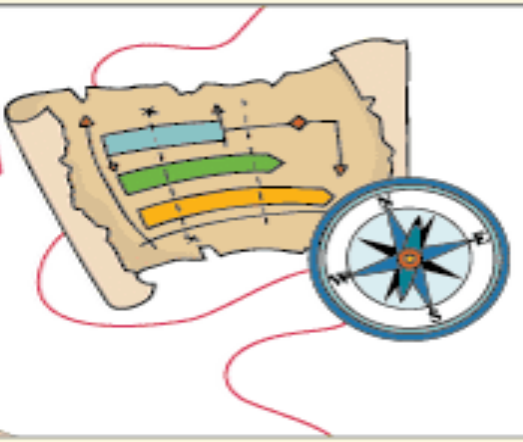


Working software **Over** Comprehensive documentation



Customer collaboration **Over** Contract negotiation





Responding to change **Over** Following a plan

While there is value in the items on the right, we value the items on the left more

12 Principles of Agile

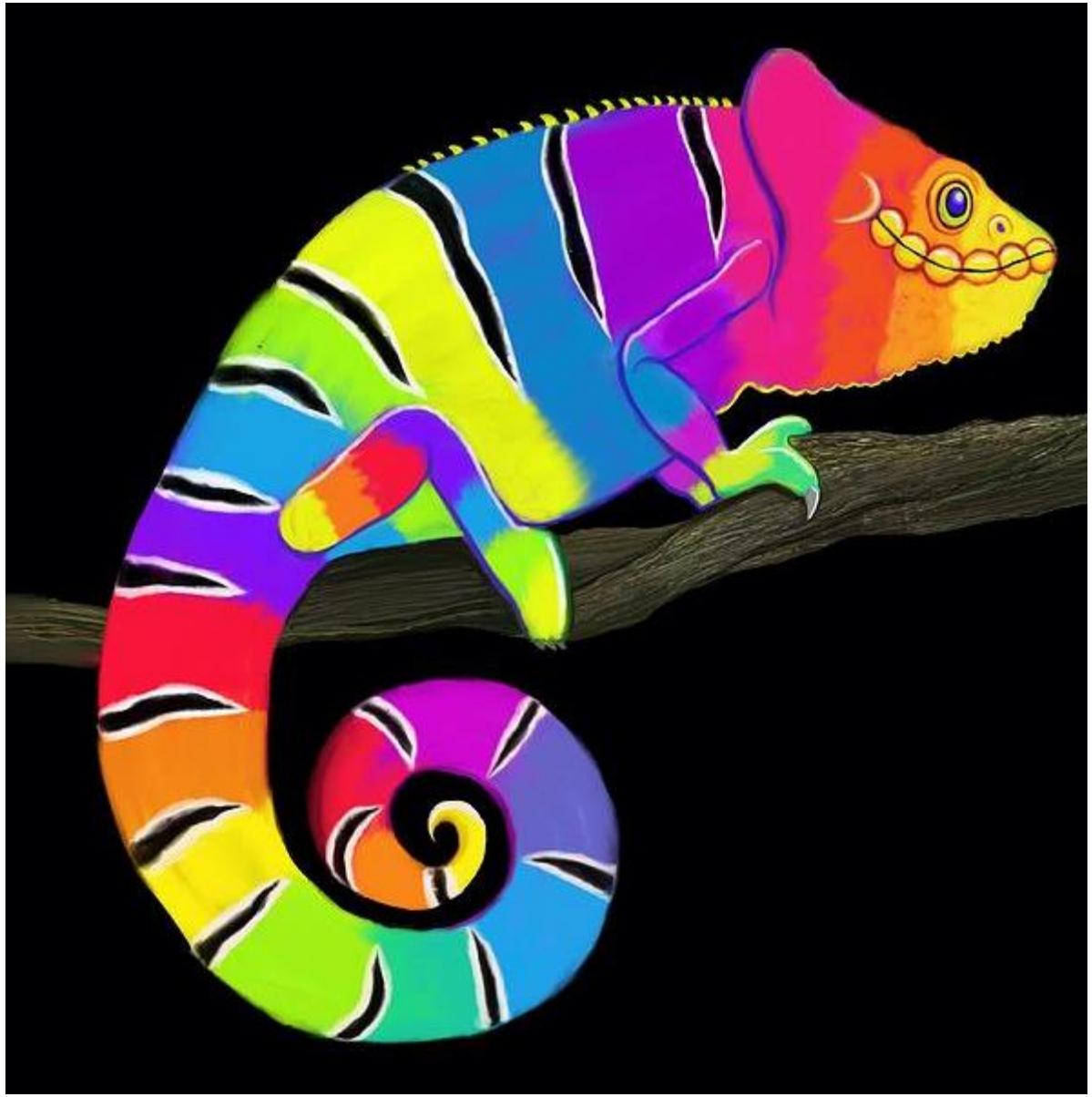
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

1st Principles of Agile

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2nd Principle of Agile

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

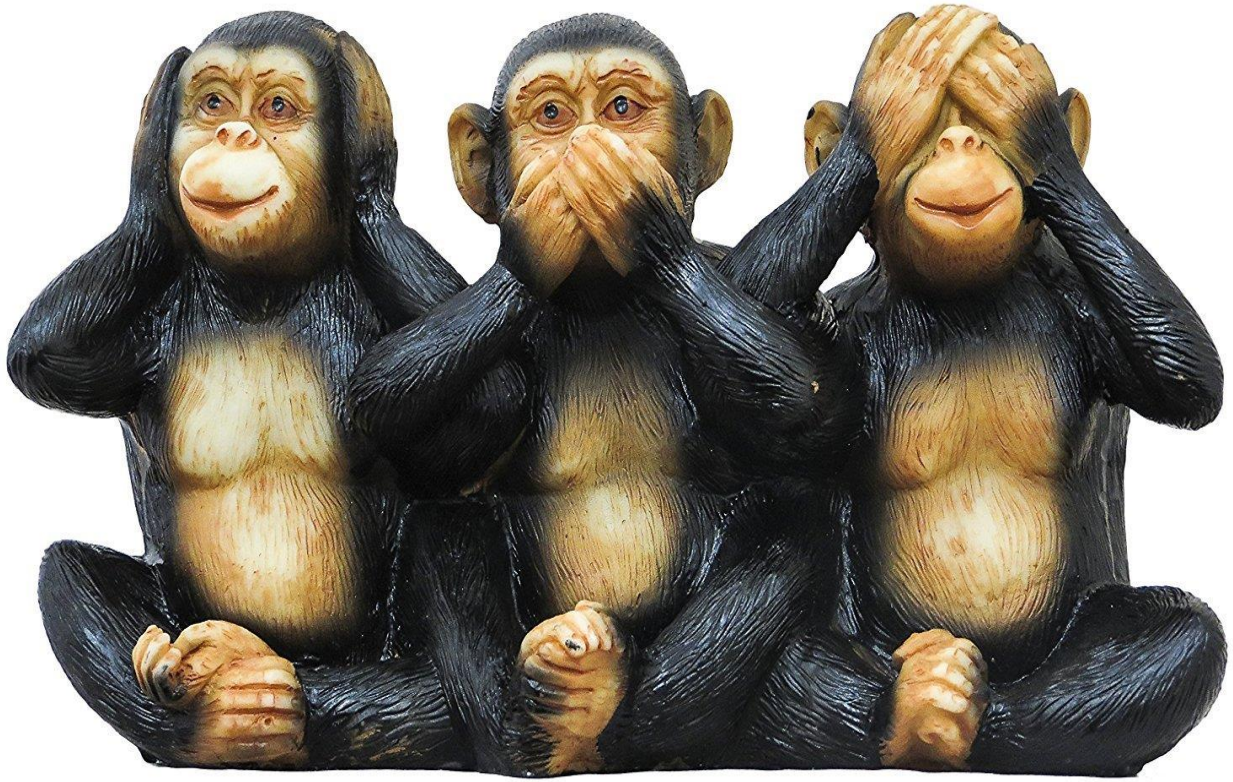


3rd Principle of Agile

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4th Principle of Agile

Business people and developers must work together daily throughout the project.



5th Principle of Agile

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6th Principle of Agile

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7th Principle of Agile

Working software is the primary measure of progress.

8th Principle of Agile

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9th Principle of Agile

Continuous attention to technical excellence and good design enhances agility.

10th Principle of Agile

Simplicity--the art of maximizing the amount of work not done--is essential.

11th Principle of Agile

The best architectures, requirements, and designs emerge from self-organizing teams.

12th Principle of Agile

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Introduction to Scrum Framework

Definition, 3Pillars, Values,Framework

Scrum Definition

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

It is a simple framework for effective team collaboration on complex products.

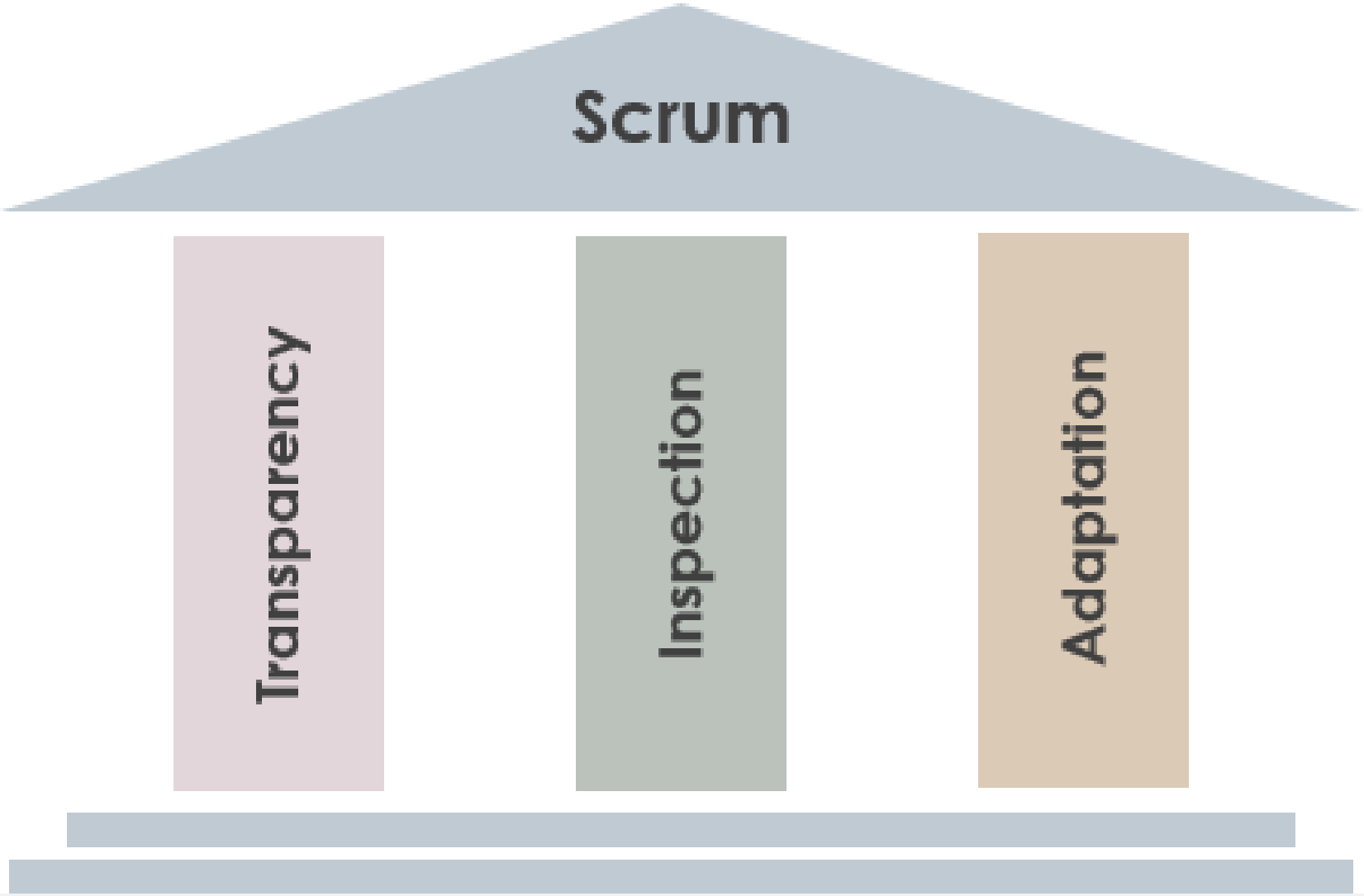
- Scrum is:
- Lightweight
- Simple to understand
- Difficult to master

Scrum is not a process, technique, or definitive method, but it is a framework within which you can employ various processes and techniques.

Scrum Definition

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software.
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

3 Pillars of Scrum



Transparency

Giving visibility to the significant aspects of the process to those responsible for the outcome.

Inspection

Timely checks on the progress toward a sprint goal to detect undesirable variances.

Adaptation

Adjusting a process as soon as possible to minimize any further deviation or issues.

Scrum Values

COURAGE
Scrum Team members have courage to do the right thing and work on tough problems

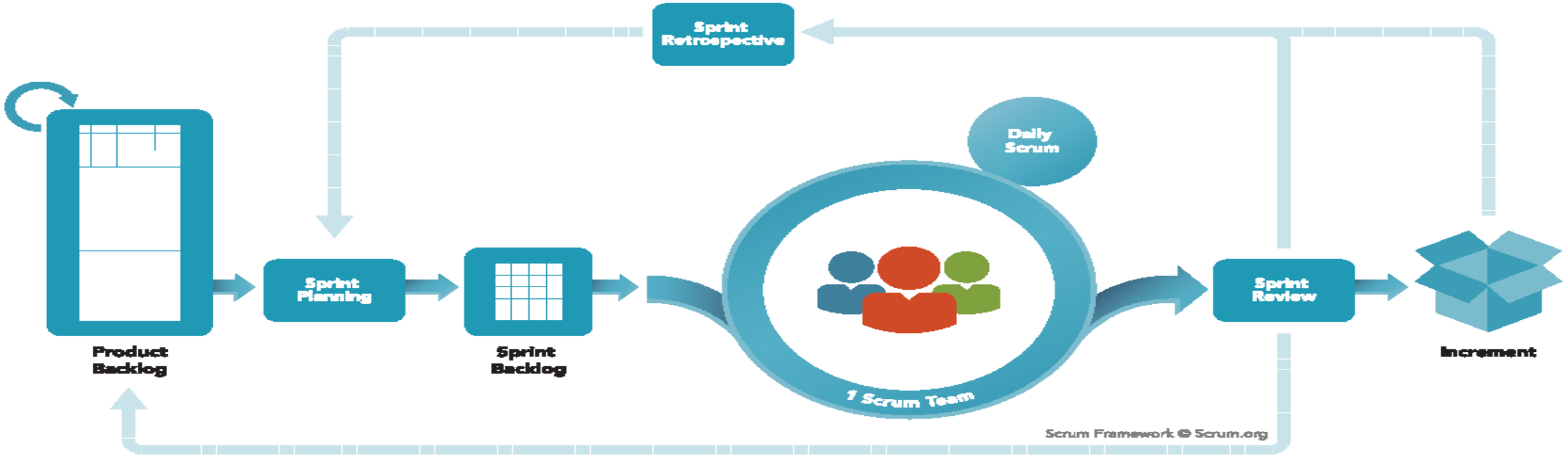
FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT
People personally commit to achieving the goals of the Scrum Team

RESPECT
Scrum Team members respect each other to be capable, independent people

OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

Scrum Framework



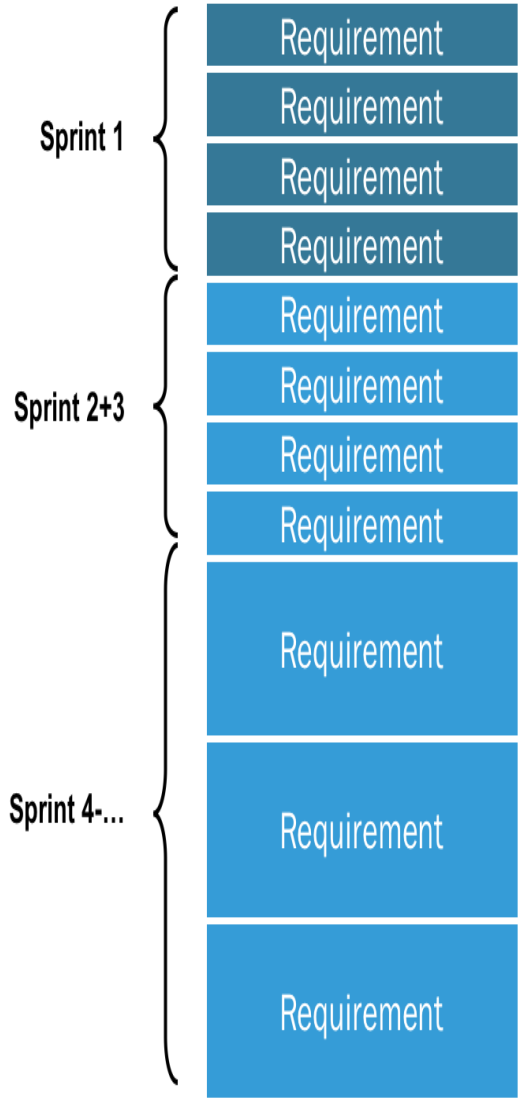
Scrum.org

Scrum Artifacts

Product Backlog, Spring Backlog, Increment

Product Backlog

1. List of everything that is needed in the product
2. Single source of requirement
3. Product owner is responsible for the product backlog
4. It is never complete
5. It constantly evolves as the product and environment evolves
6. Backlog refinement/Backlog grooming is an ongoing process to keep the Product backlog updated



Sprint Backlog

1. Subset of product backlog
2. It is a plan to achieve product increment and the sprint goal.
3. It creates visibility of all necessary work to achieve sprint goal
4. It provides a real time picture of the work that is planned to be accomplished.
5. The Sprint backlog can be modified only the Development Team.
6. Development team is the sole owner of the Sprint Backlog

Increment

1. Sum of all Product Backlog items completed during a particular sprint.
2. Value of increment of all previous sprints.
3. Must be Usable and done irrespective at the end of sprint.
4. Must be inspectable

Scrum Ceremonies

Planning, Daily Standup, Review, Retrospective

Sprint Ceremonies

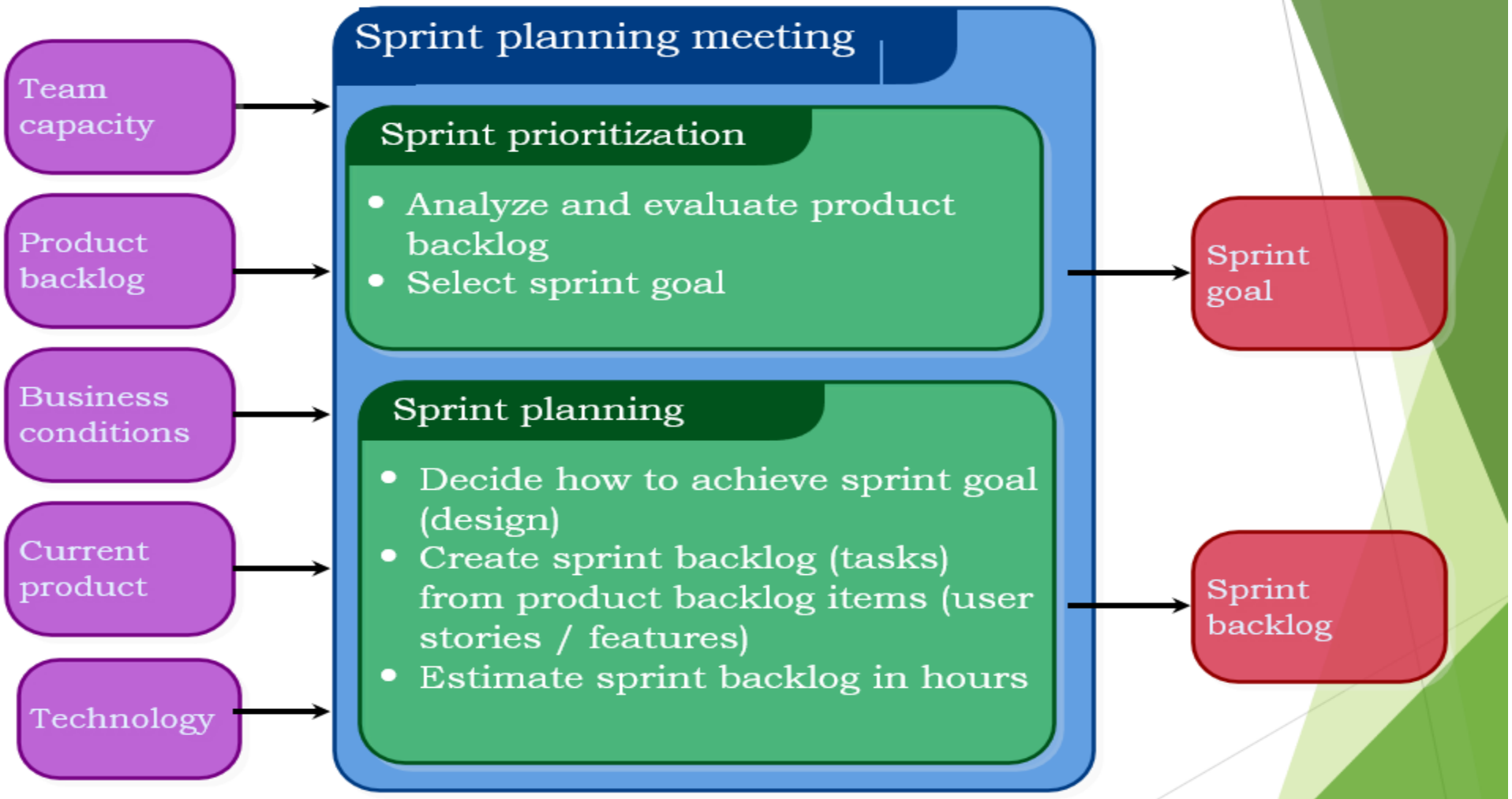
Duration recommended for 2 week sprint

	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
Purpose	• Create Sprint Backlog • Identify Sprint Goal	• Provide opportunity to discuss • Progress • Daily commitments • Identify and remove impediments	• Signoff/ Approval • Gather feedback	Inspect and Adapt
What	• Discuss User Stories • Story point Estimation • Sprint Capacity Planning	• What was done • What will be done • Blockers	• Demonstrate working software • Create increment	• Good • Bad • Ideas • Actions
Duration	No longer than 4 hours	Not longer than 15 min	2hrs recommended	Not longer than 1.5 hrs
Attendees	Entire Scrum Team	• Scrum Master Development Team • Product Owner(Optional) • Stakeholders(Optional)	• Entire Scrum Team • Stakeholders • Managers	• Scrum Master Development Team • Product Owner(Optional) • Stakeholders(Optional)

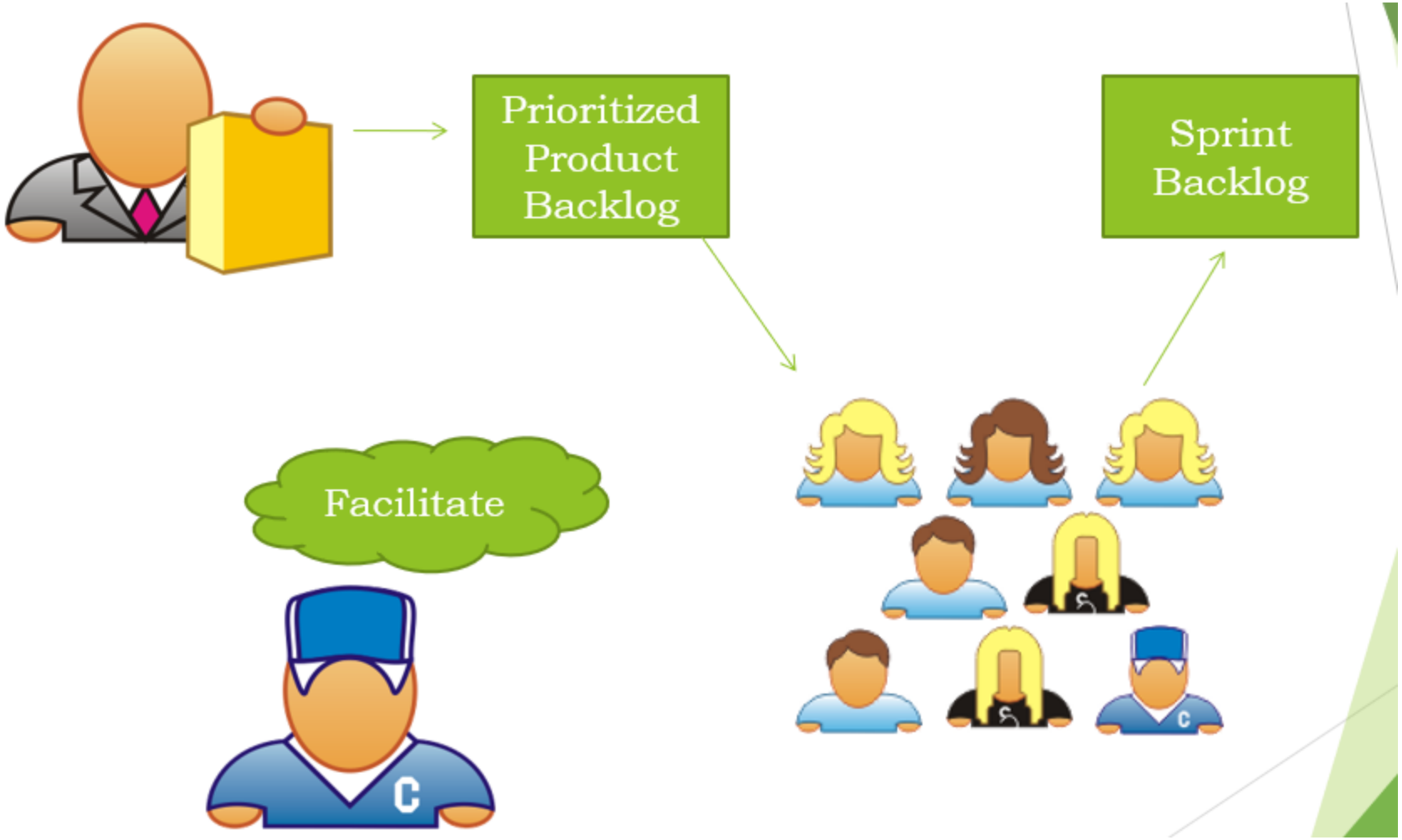
Sprint Planning

- The Team and the Product Owner collaborate to help the Team determine how much Product Backlog it can turn into functionality during the upcoming Sprint.
- The Team create plans (Sprint Backlog) by identifying tasks for converting selected Product Backlog into functionality

Sprint Planning



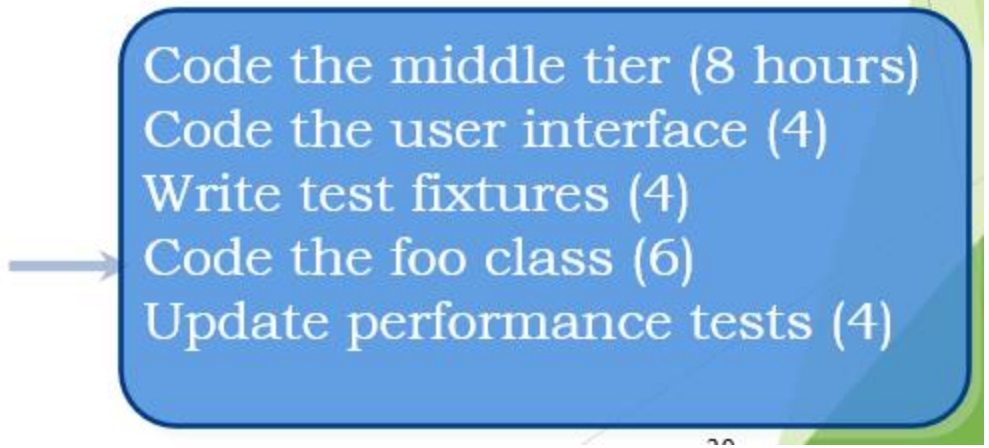
Sprint Planning



Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours)
  - Collaboratively, not done alone by the Scrum Master
- High-level design is considered

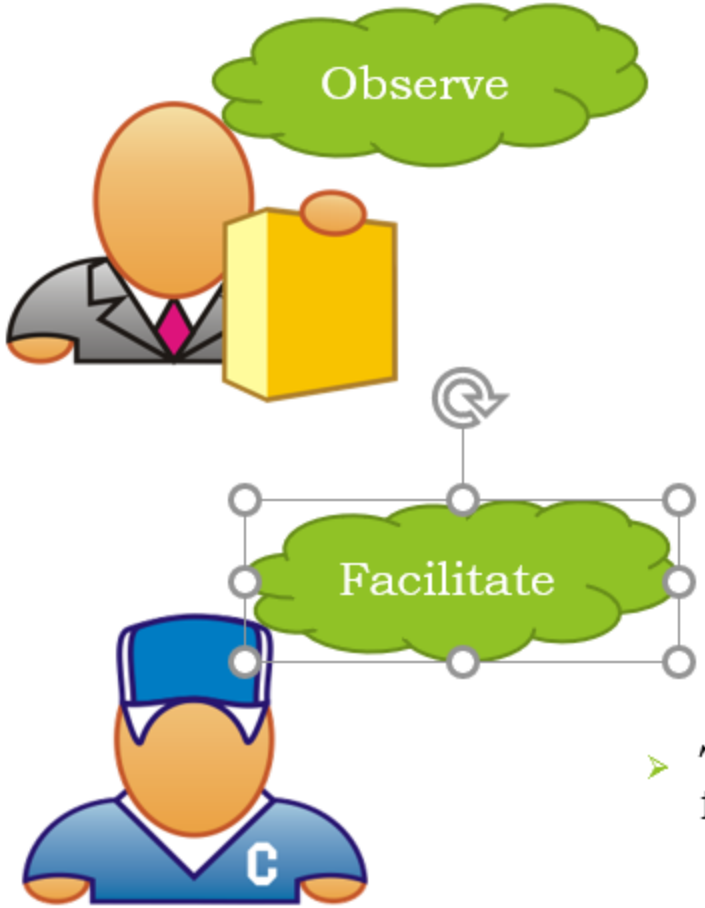
As a vacation planner, I want to see photos of the hotels.



Daily Scrum

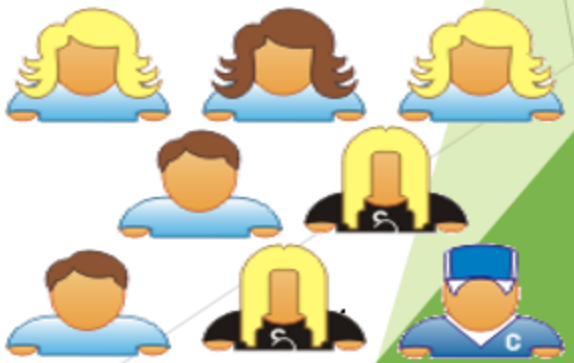
- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Only team members, Scrum Master, Product Owner, can talk
- Helps avoid other unnecessary meetings
- Team is responsible of conducting this meeting

Daily Scrum



1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the Scrum Master
  - They are commitments in front of peers



Sprint Review

- As described in the Scrum Guide, the purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations.
- The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.
- Informal
  - 2-hour prep time rule for 2 weeks Sprint
  - No slides
- Whole team participates
- Invite the world

Sprint Review

Sprint Retrospective

- The sprint retrospective is a recurring meeting held at the end of a sprint used to discuss what went well during the previous sprint cycle and what can be improved for the next sprint.
- The Agile sprint retrospective is an essential part of the Scrum framework for developing, delivering, and managing complex projects.
- Duration is 3 hours for 4 weeks Sprint or 1.5 hr. for 2-week sprint

- Participants
  - Scrum Master
  - Team
  - Product owner (Optional)

Start doing

Stop doing

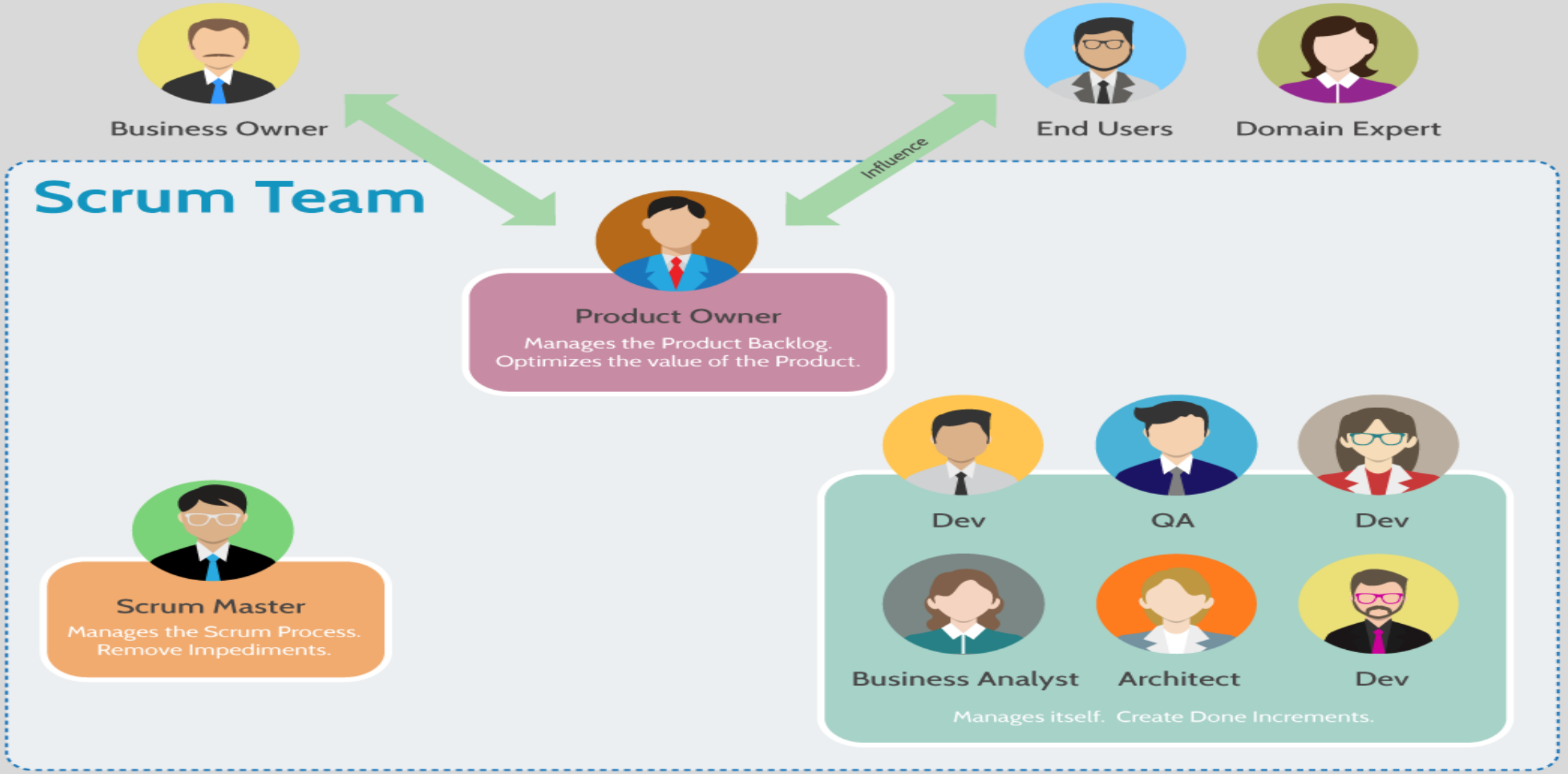
Continue doing

This is just one of many ways to do a sprint retrospective

Scrum Team

Product Owner, Scrum Master, Development Team

Scrum Team


Business Owner
End Users
Domain Expert
Influence
Scrum Team
Product Owner
Manages the Product Backlog.
Optimizes the value of the Product.
Scrum Master
Manages the Scrum Process.
Remove Impediments.
Dev
QA
Dev
Business Analyst
Architect
Dev
Manages itself. Create Done Increments.

Scrum Team

Roles and Responsibilities

	Roles and Responsibilities
Development Team	• They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality; • Development Teams are cross-functional, with all the skills as a team necessary to create a product Increment; • Scrum recognizes no titles for Development Team members, regardless of the work being performed by the person; • Scrum recognizes no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations or business analysis; and, • Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.
Product Owner	• Expressing Product Backlog items clearly. • Ordering the Product Backlog items to best achieve goals and missions. • Optimizing the value of the work the Team performs. • Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Team will work on further. • Ensuring that the Team understands items in the Product Backlog to the level needed.
Scrum Master	• ensure the process run smoothly • remove obstacles that impact productivity • organize the critical events and meeting

ThankYou

Balasubramaniyan Chandrasekar

balasubramaniyanc@maveric-systems.com