GROUP MENTORING

SERGIO
GROUP - B4
15.06.2020

GROUP - B4

E2014	Devin
E2128	Uğur
E2151	Fırat
E2174	Durmus
E2252	Hayati Yusuf
E2263	Ramazan Rafet
E2332	Adem
E2366	Vedat

GÜNDEM

- Grup İsmi
- One-on-One Meeting için Calendly’den Randevu Alınması
- Path tercihlerinin Bildirilmesi
- HTML ve CSS
- AGILE
- Jira Software Kurulumu
- Python
- Linux

GRUP İSMİ

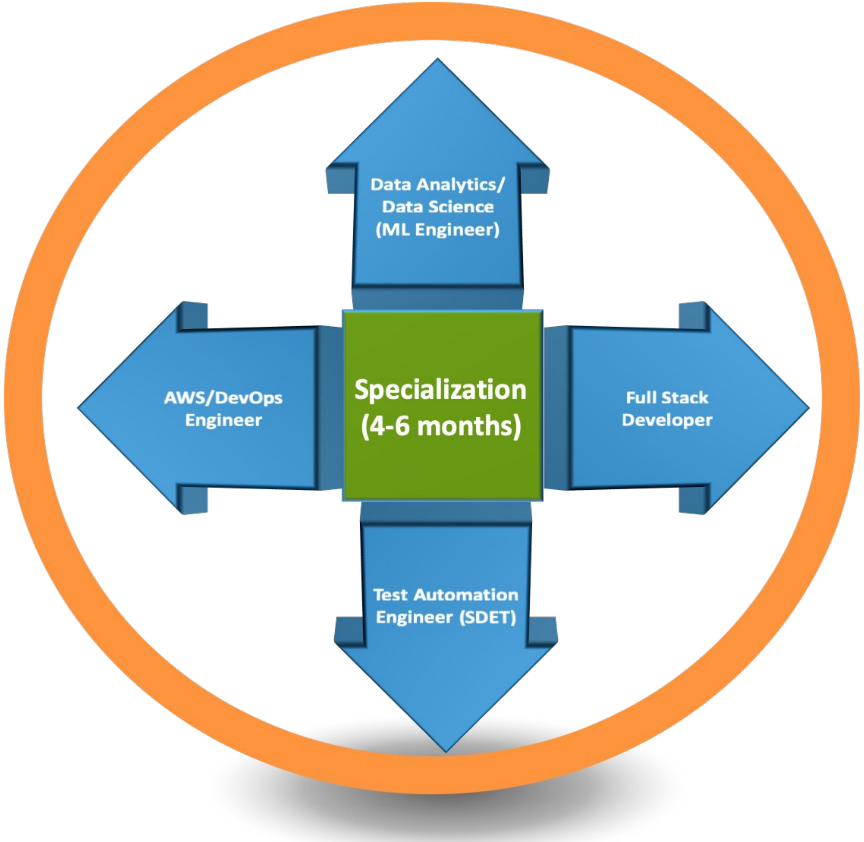
BAGER

ONE-on-ONE MEETING

https://calendly.com/sergio_taco

PATH TERCİHLERİNİN İLETİLMESİ

16 HAZİRAN 2016
15.00



Specialization (4-6 months)

Data Analytics/ Data Science (ML Engineer)

AWS/DevOps Engineer

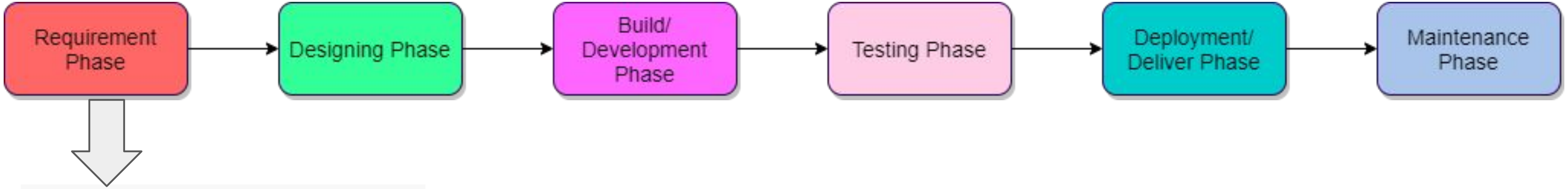
Full Stack Developer

Test Automation Engineer (SDET)

HTML ve CSS

- Assignments
- Pre-Class Material
- Workshop
- Lab
- Quiz

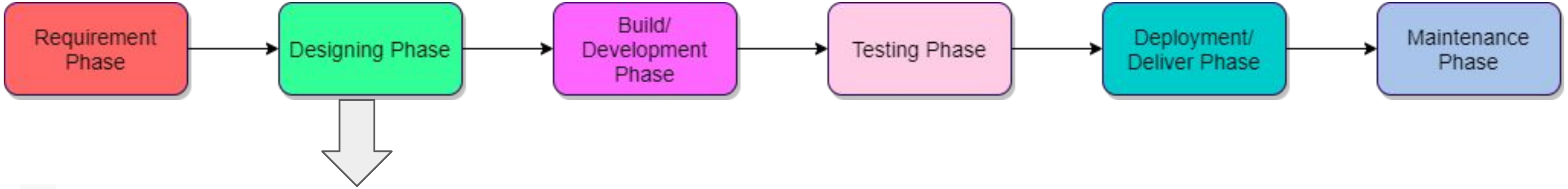
SDLC



1. REQUIREMENT PHASE :The first and the most critical phase. In this phase;

- The client specifies requirements, specifications, expectations and any other special requirement related to the product or software.
- The business manager or project manager gathers all of this information and also prerequisites.
- Once all the information gathered, the next step is to clearly define and document the product requirements and get them approved by the customer or the market analysts.
- This is done through an SRS (Software Requirement Specification) document. It consists of all the necessary requirements to be designed and developed during the project life cycle.

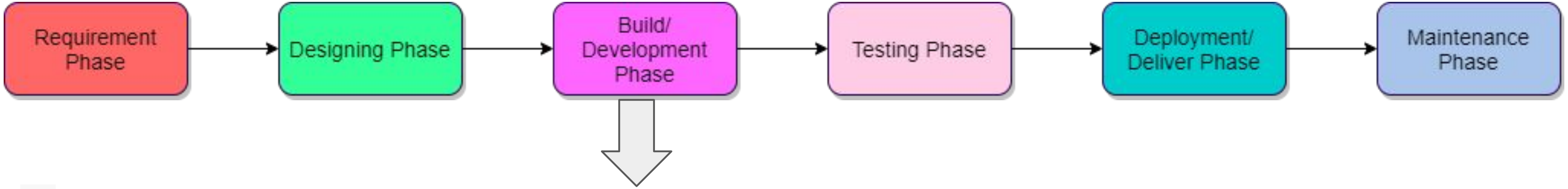
SDLC



2. DESIGN PHASE: In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

- This is the high priority phase in a system's development life cycle because the logical designing of the system is converted into physical designing.
- The output of the requirement phase is a list of things that are required and the design phase gives the way to accomplish these requirements.
- The decision of all required essential tools such as programming language like Java, .NET, PHP; database like Oracle, MySQL; a combination of hardware and software to provide a platform on which software can run without any problem is taken in this phase.
- There are several tools and techniques used for describing system design, such as Flowchart, Data flow diagram (DFD), Data dictionary, Structured English, Decision table, and Decision tree.

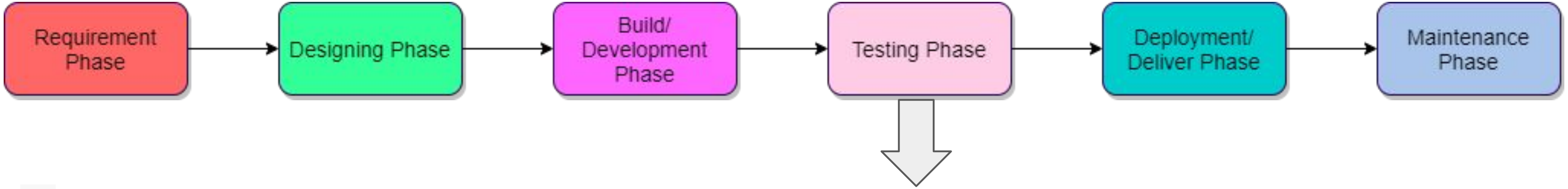
SDLC



3. BUILD/DEVELOPMENT PHASE: After the successful completion of the requirement and design phase, the next step is to implement the design into the development of a software system.

- This phase is also known as coding phase.
- Developers start to build the entire system by writing code using the chosen programming language.
- Work/task is divided into small units or modules, and coding starts by the team of developers according to the design and the requirements of the client to produce the desired result.
- Coding Phase is the longest phase of the SDLC process, and it requires a more focused approach for the developer.

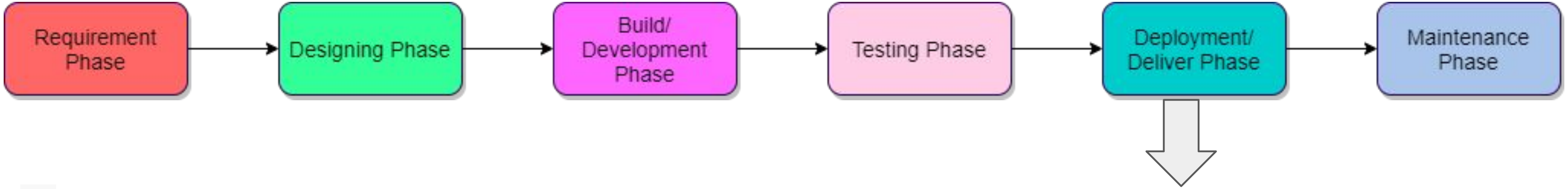
SDLC



4. TESTING PHASE: This phase is where you focus on investigation and discovery.

- The testing team starts testing the functionality of the entire system. This is done to verify that the software works and gives the result as per the requirements addressed in the requirement phase or not.
- The development team makes a test plan to start the test. This test plan includes all types of essential testing such as integration testing, unit testing, acceptance testing, and system testing.
- If there is a bug/defect detected in the software, or it is not working as expected. The testing team gives detailed information to the development team about the issue. If the defect is valid or worth fixing, it will be fixed and the development team replaces it with the new one. It also needs to be verified.

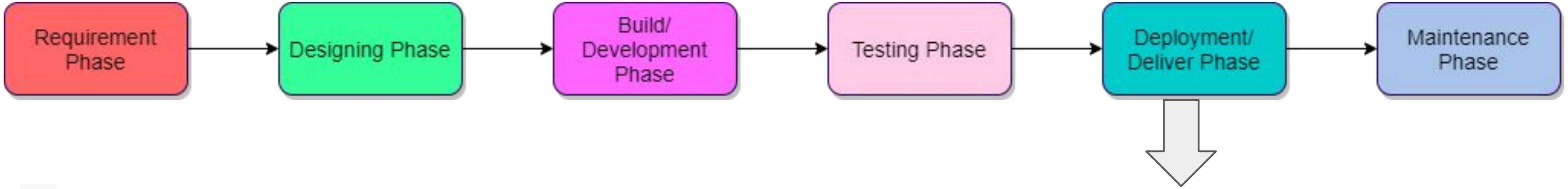
SDLC



5. DEPLOYMENT/DELIVER PHASE: When software testing is completed with a satisfying result and there are no remaining issues in the working of the software, it is delivered to the customer.

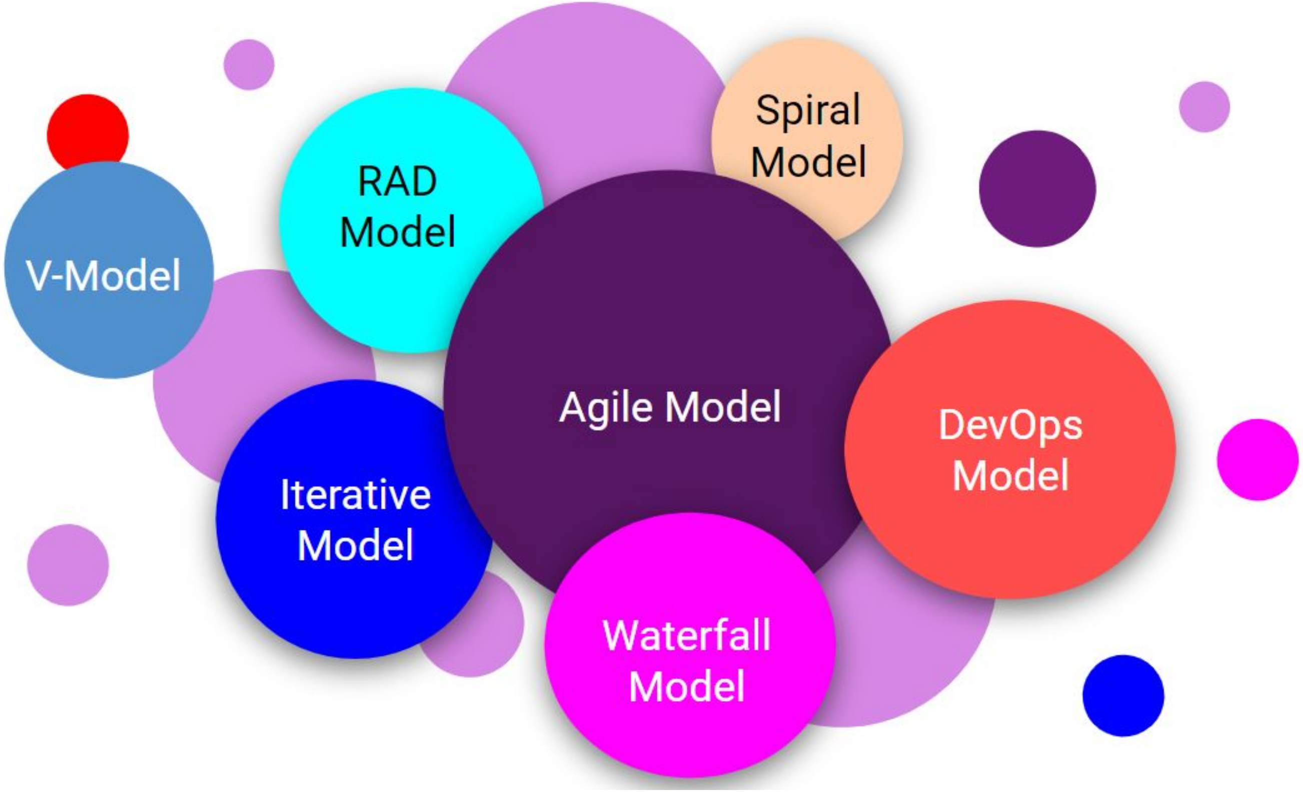
- As soon as customers receive the product, they are recommended first to do the beta testing. In beta testing, customers can require any changes which are not present in the software but mentioned in the requirement document to make it more user-friendly.
- Besides this, if any type of defect is encountered while a customer using the software, the development team will be informed to fix this problem. If it is a critical defect, the development team solves it in a short time. Otherwise, it will wait for the next version.
- After the solution of all types of bugs and changes, the software finally deployed to the end-user.

SDLC



6. MAINTENANCE PHASE: The last phase of the process SDLC is the maintenance phase where the process continues until the software's life cycle comes to an end. When a customer starts using software, actual problems start to show up. At that time, there's a need to solve these problems. Maintenance Phase also includes making changes in hardware and software to maintain its operational effectiveness like to improve its performance, enhance security features and address customer's requirements.

SDLC MODELS


Spiral Model
RAD Model
V-Model
Agile Model
DevOps Model
Iterative Model
Waterfall Model

SDLC MODELS

1. **WATERFALL MODEL**
2. **ITERATIVE MODEL**
3. **SPIRAL MODEL**
4. **V-MODEL**
5. **RAD MODEL**
6. **AGILE MODEL**
7. **DEVOPS MODEL**

AGILE

AGILE NEDİR : Software projects face the risk of failure during the life cycle due to misuse of resources, wrong prioritization and inaccurate understanding of customer demands. This has led companies to experiment with a variety of new methods to produce more productive projects.

Agile is a new approach to project management methods resulting from these experiments. This method is based on an incremental and iterative approach that helps teams deliver a product to their customers faster and more desirable.

In this respect, the framework of agile promises significant advantages over traditional methods. In recent years, most companies have adopted the agile approach, which they find more successful than traditional methods.

In this context, we can say that almost every software developer has used the agile method in some form.

AGILE

Dividing Product Development Life Cycle into Small Components

Agile provides a light framework for development teams. That helps them focus on fast delivery while maintaining functionality. In Agile, the product development life cycle is divided into small components (called iterations), so the product can be easily and rapidly developed and tested. In this way, changes can be made without having to wait for the final product.

AGILE

Customer's Desired Requirements

The agile method provides that the product is optimized throughout the development process. Using iterative planning and getting feedback ensures that a delivered product is tailored to the customer's requirements. The status of the project is measured and evaluated throughout the process to easily adapt to changing needs. Measurement and evaluation provide accurate and early visibility into the development of each project.

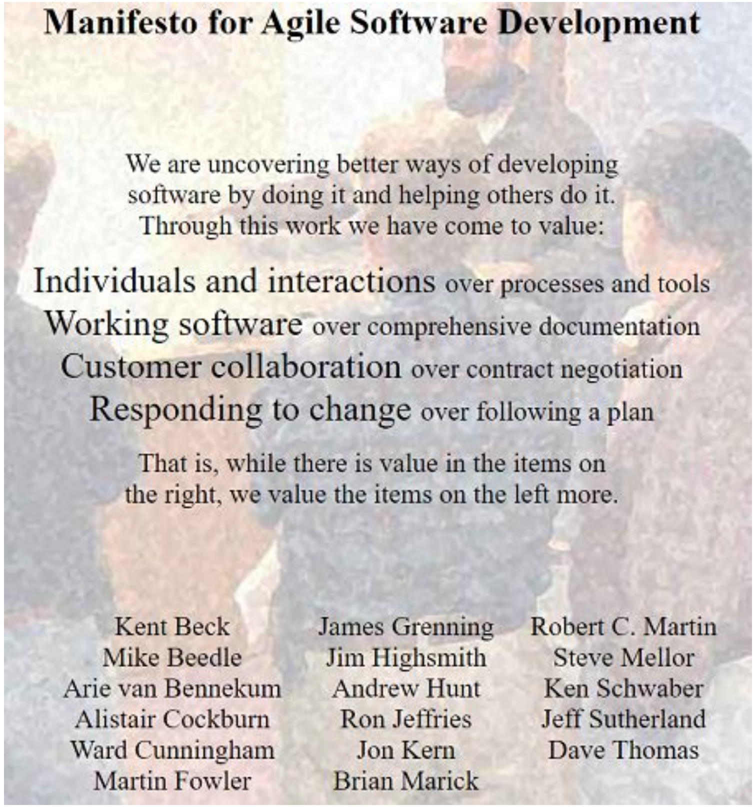
AGILE

1. **Individuals and interactions over processes and tools.**

2. **Working software over comprehensive documentation.**

3. **Customer collaboration over contract negotiation.**

4. **Responding to change over following a plan.**



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 and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

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

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

AGILE

Agile Principles

In the months following the publication of the Agile Manifesto, the original signatories continued to communicate. They augmented the four values of the manifesto with the following 12 principles.

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 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

AGILE

Agile Principles

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.

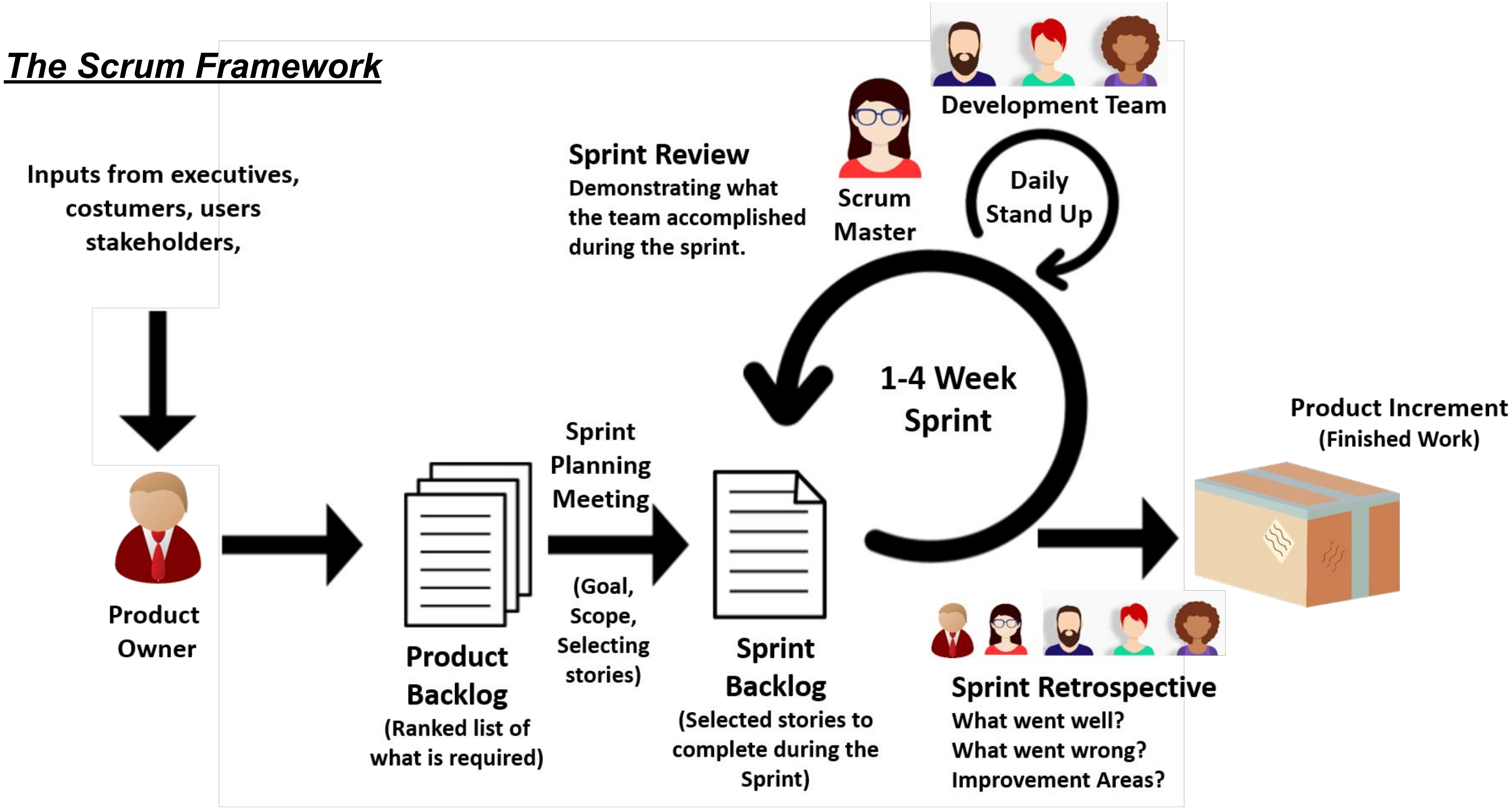
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Scrum

Table of Contents

- ▶ Sprint Concept
- ▶ Scrum Roles
- ▶ Epic, User Story and Task
- ▶ Scrum Artifacts
- ▶ Scrum Ceremonies

Sprint Concept



Scrum Roles

Product Owner

Scrum Master

Development Team

Product Owner

Product Owner

- Business representative
- Represents stakeholders
- Voice of the customer
- Works together with stakeholders
- Prioritizes the product backlog

Scrum Master

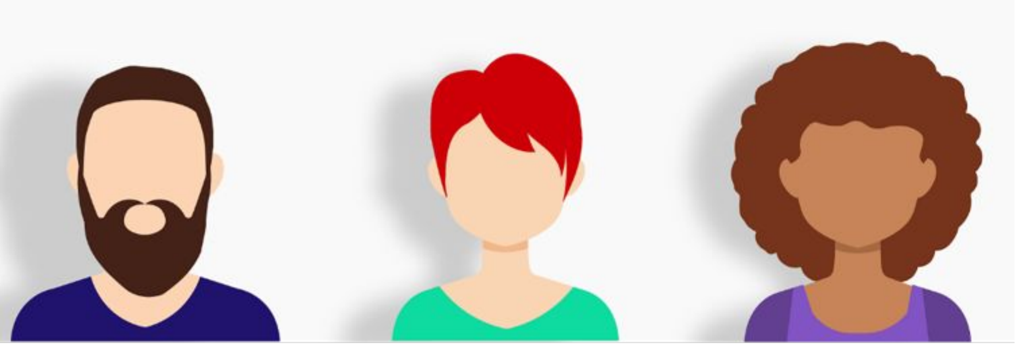
Scrum Master

- Coaches the team
- Clears any obstacles encountered
- Helps team members focus on what they do
- Ensures that the team is working properly

Scrum Master

is not Scrum
KOOS, WHAT DOES A SCRUM MASTER DO ALL DAY?
WE PRACTISE THE ART OF ACTIVELY DOING NOTHING!
... CAN I BE A SCRUM MASTER TOO?

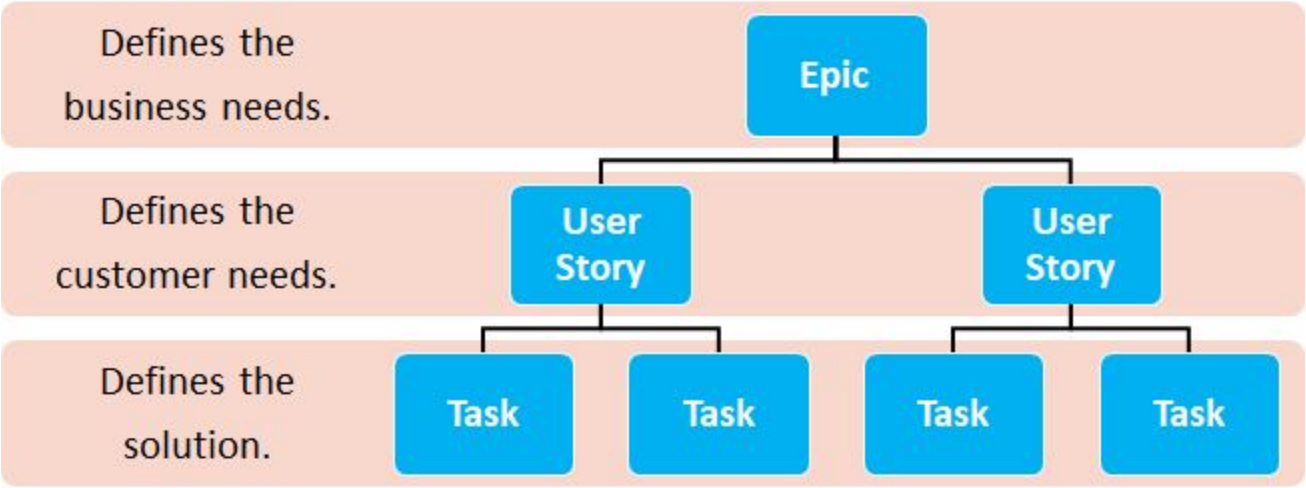
Development Team



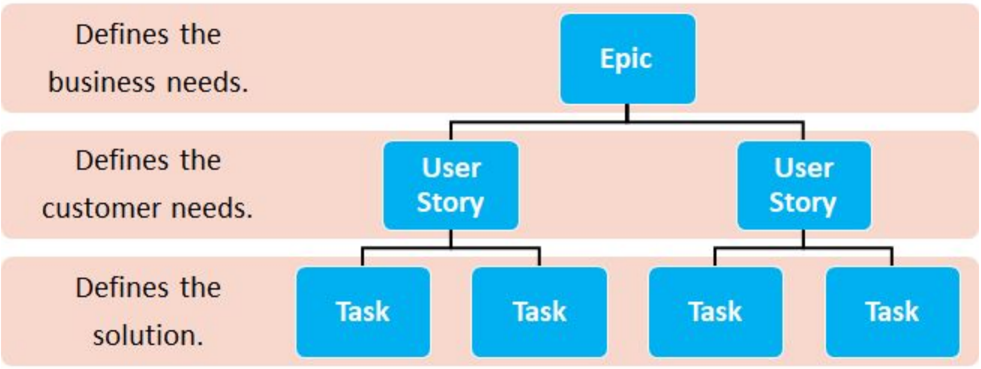
Development Team

- Consists of 3-9 people
- Performs daily tasks

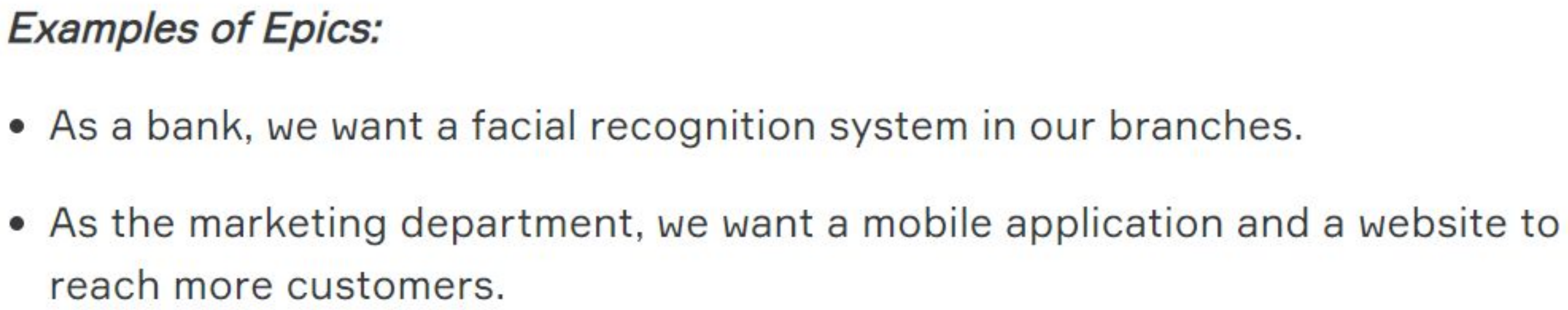
Epic, User Story and Task



Epic



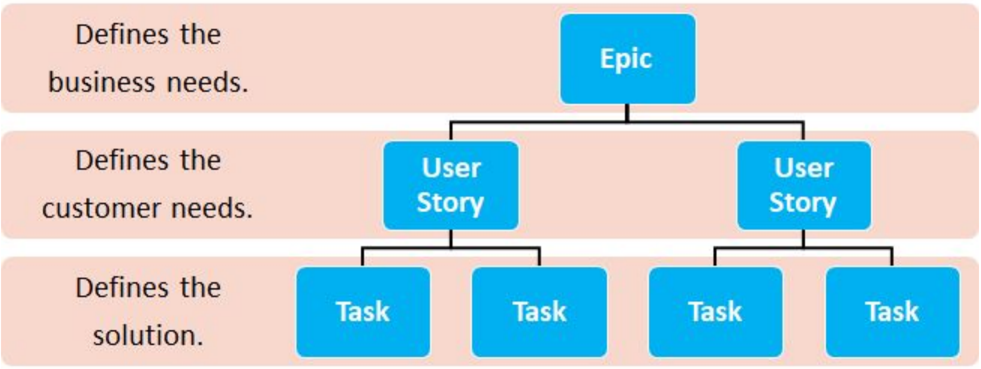
- **A set of jobs that cannot be easily achieved in a single sprint.**



Examples of Epics:

- As a bank, we want a facial recognition system in our branches.
- As the marketing department, we want a mobile application and a website to reach more customers.

User Story



Defines the business needs.

Epic

Defines the customer needs.

User Story

User Story

Defines the solution.

Task

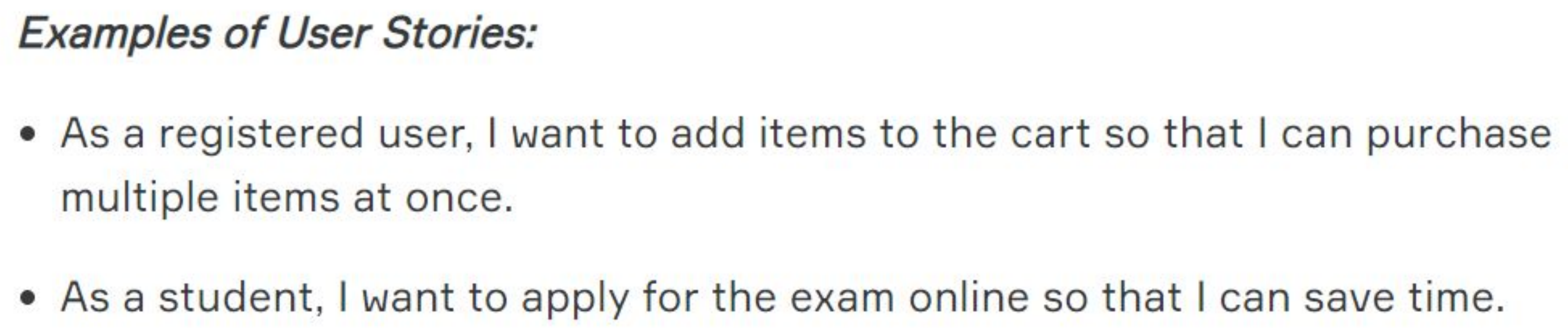
Task

Task

Task

- **Describes a software feature from the customer’s perspective**

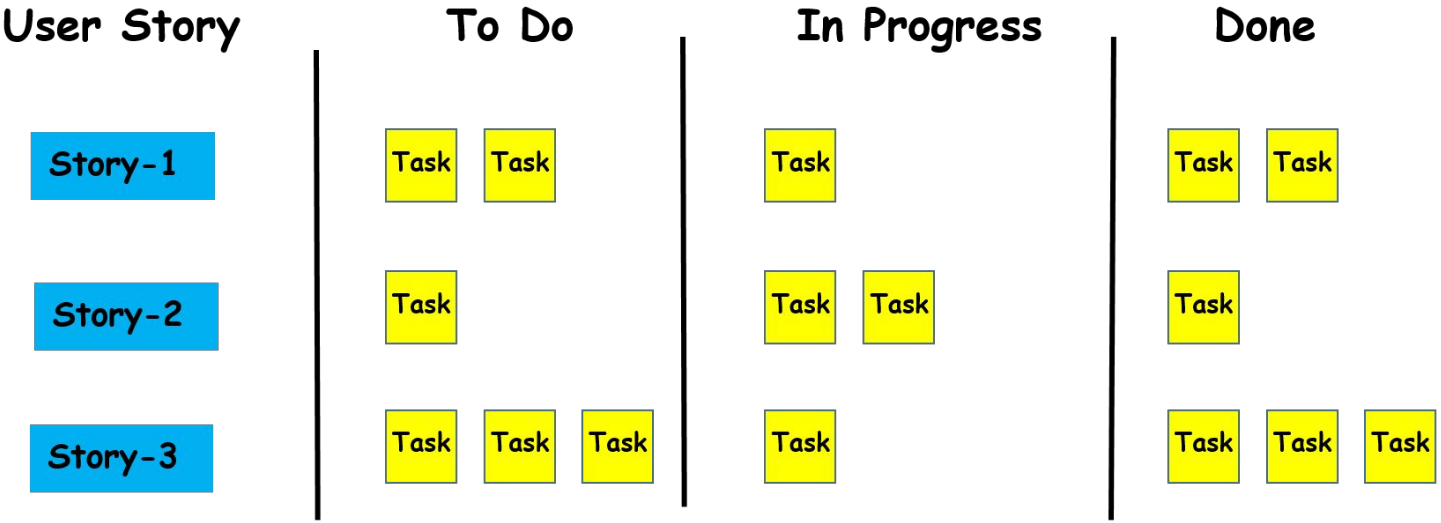
- **Includes the “*type of user*”, *“what they want”*, and *“why they want it”.***



Examples of User Stories:

- As a registered user, I want to add items to the cart so that I can purchase multiple items at once.
- As a student, I want to apply for the exam online so that I can save time.

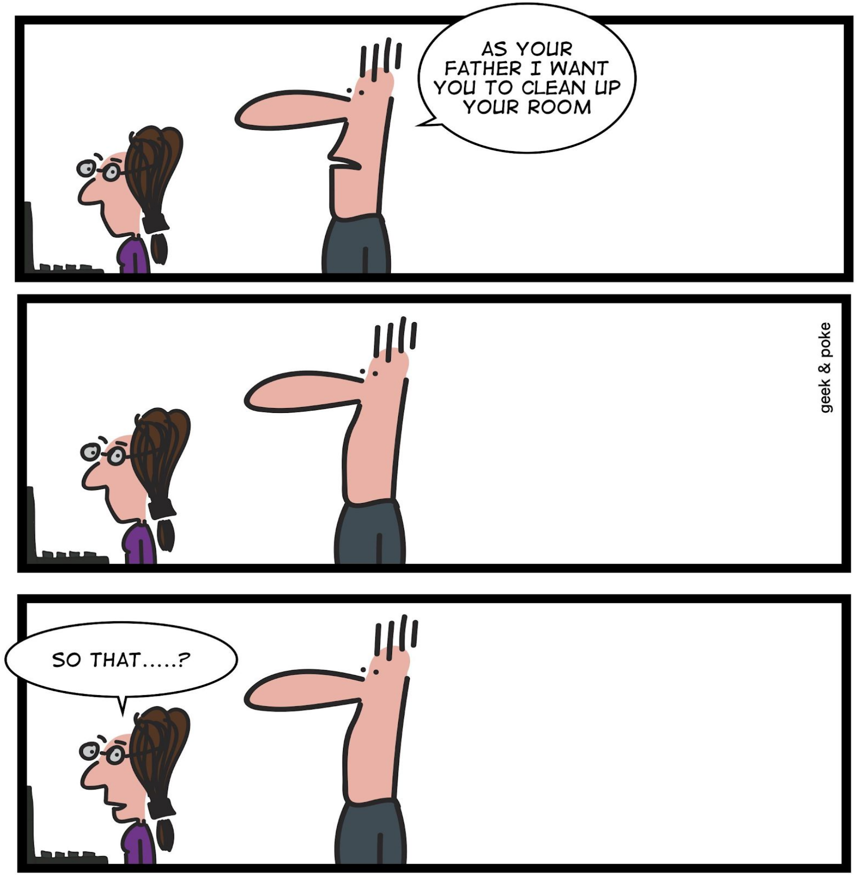
Scrum Board



User Story	To Do	In Progress	Done
Story-1	Task Task	Task	Task Task
Story-2	Task	Task Task	Task
Story-3	Task Task Task	Task	Task Task Task

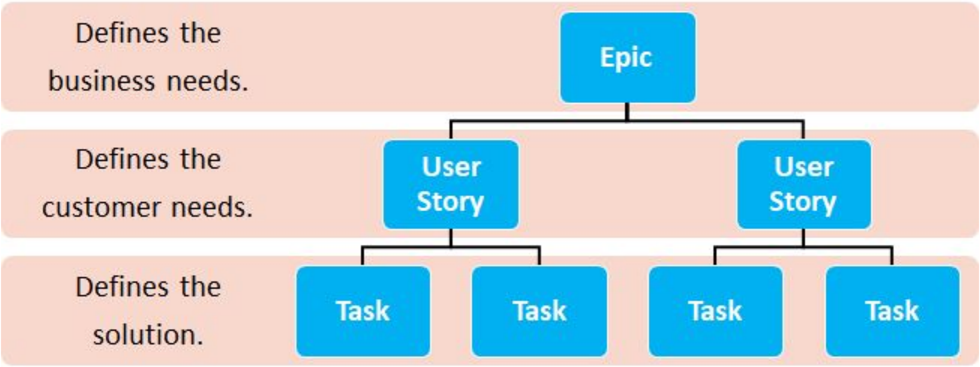
- **Manual or electronic**

User Story



MAKE SURE YOUR USER STORY IS CORRECTLY PHRASED

Tasks



- **Detailed pieces of work that are necessary to realize a user story.**
- **Define the solutions for customer needs.**

Epic, User Story and Task

What do you think?

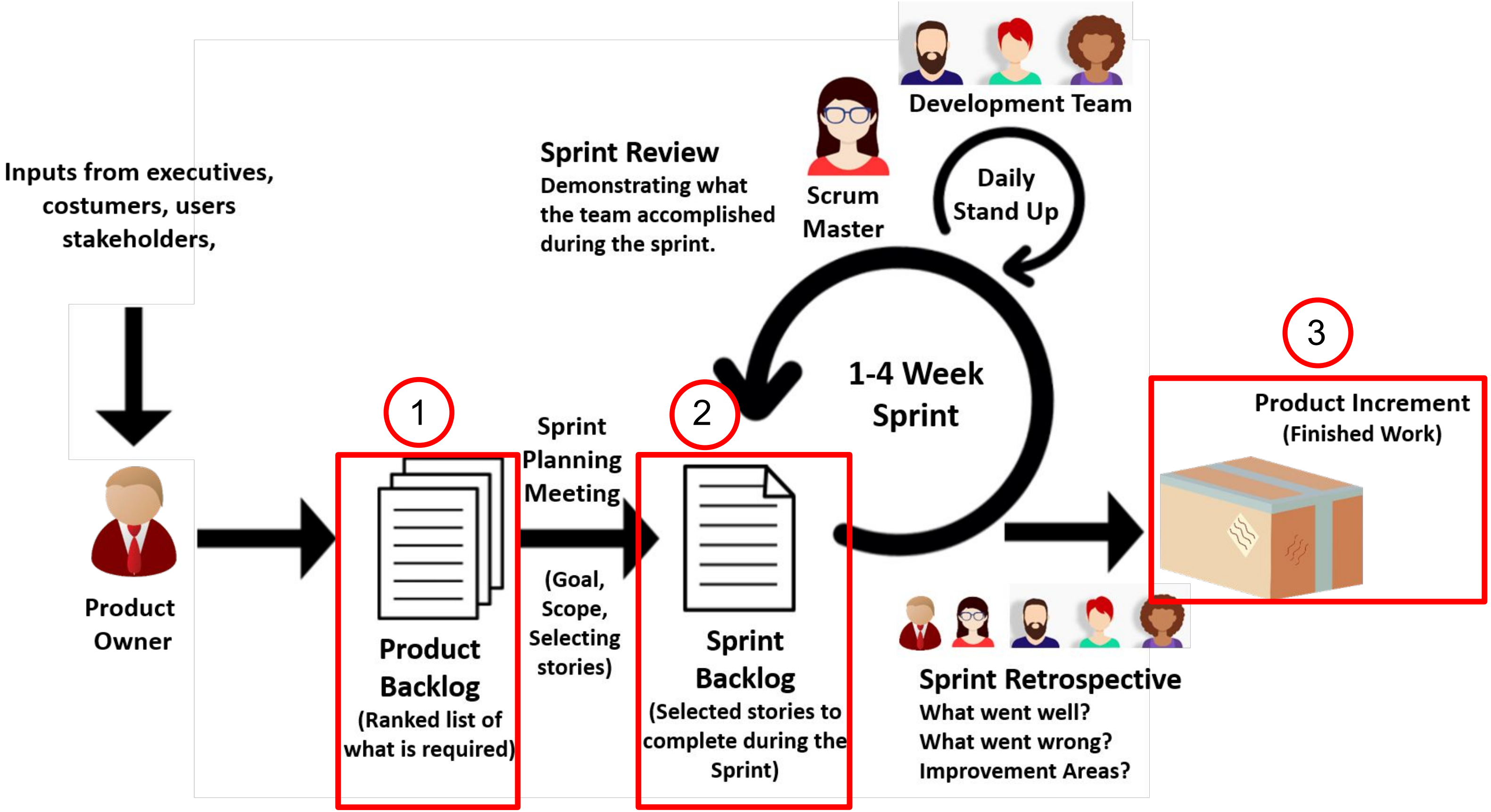
Which one maybe NOT appropriate?

Categorize the following events as Epic, User Story or Task:

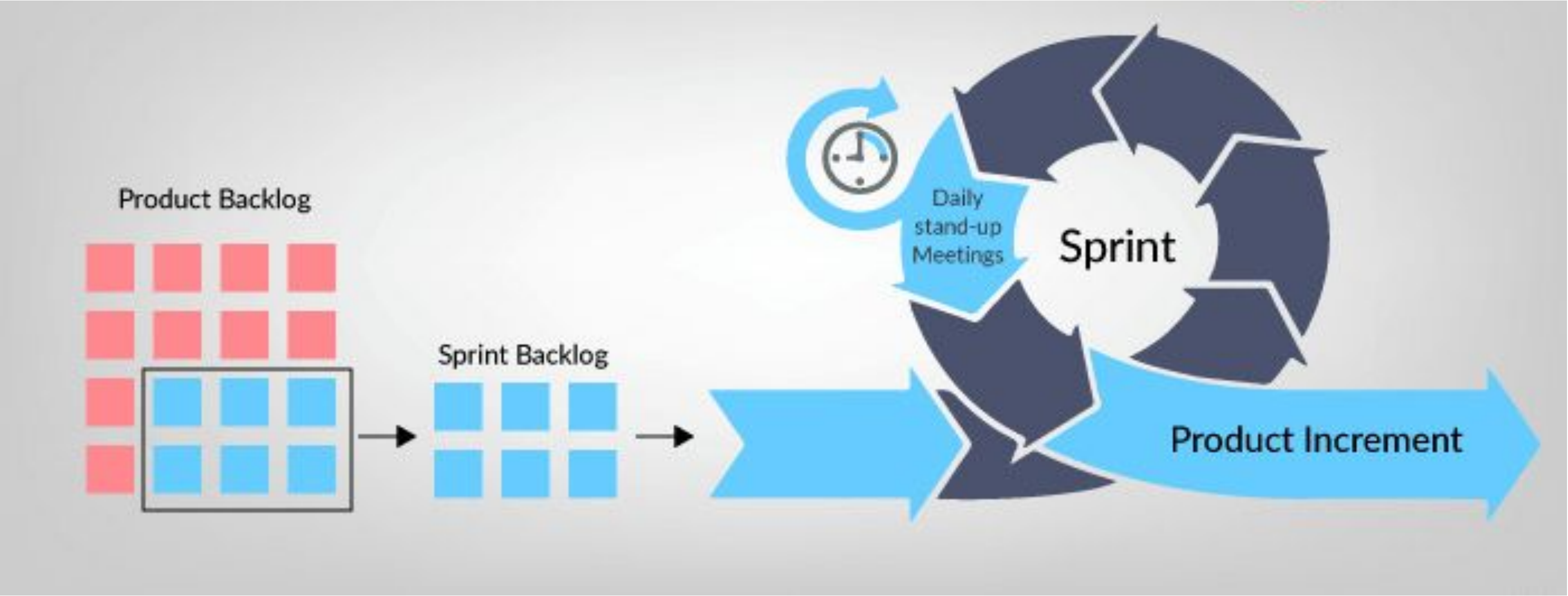
- Making a Website
- Creating a homepage for the website
- Making a button for the homepage.

Students choose an option

Main Scrum Artifacts

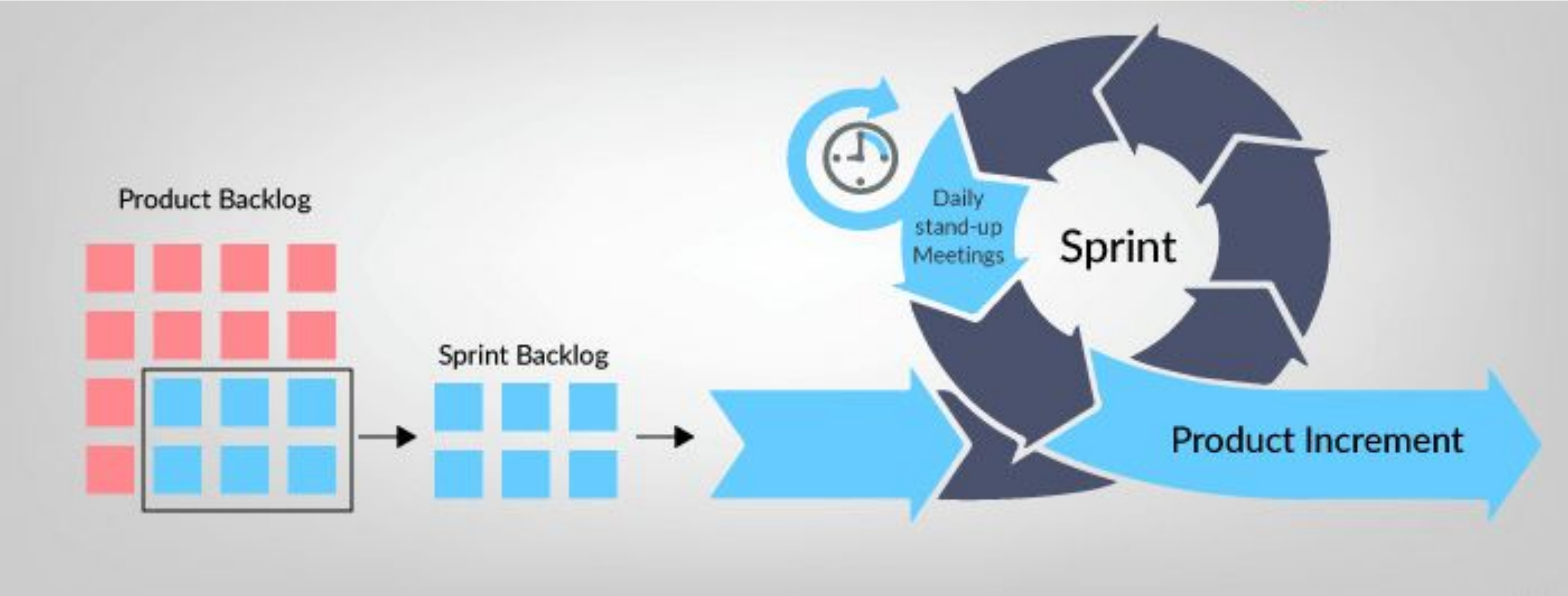


Product Backlog



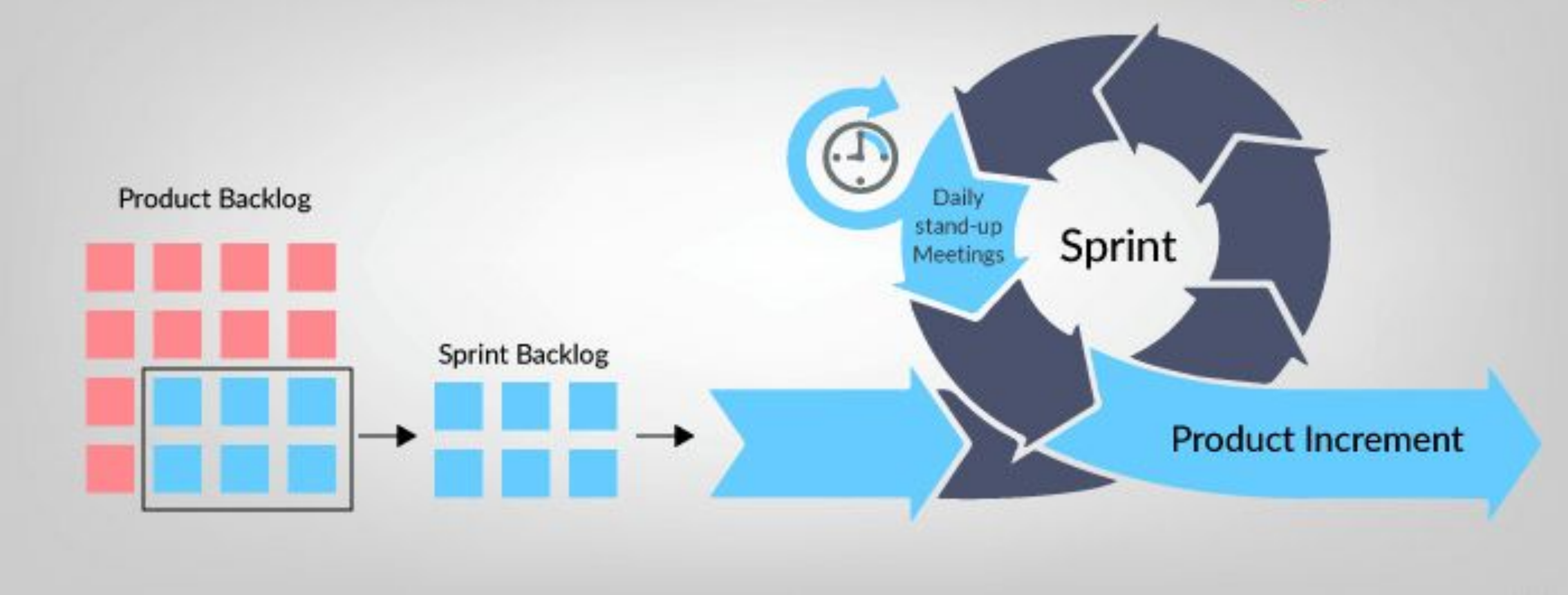
- **List of everything to complete the project**
- **Beside all user stories, includes technical tasks**

Sprint Backlog



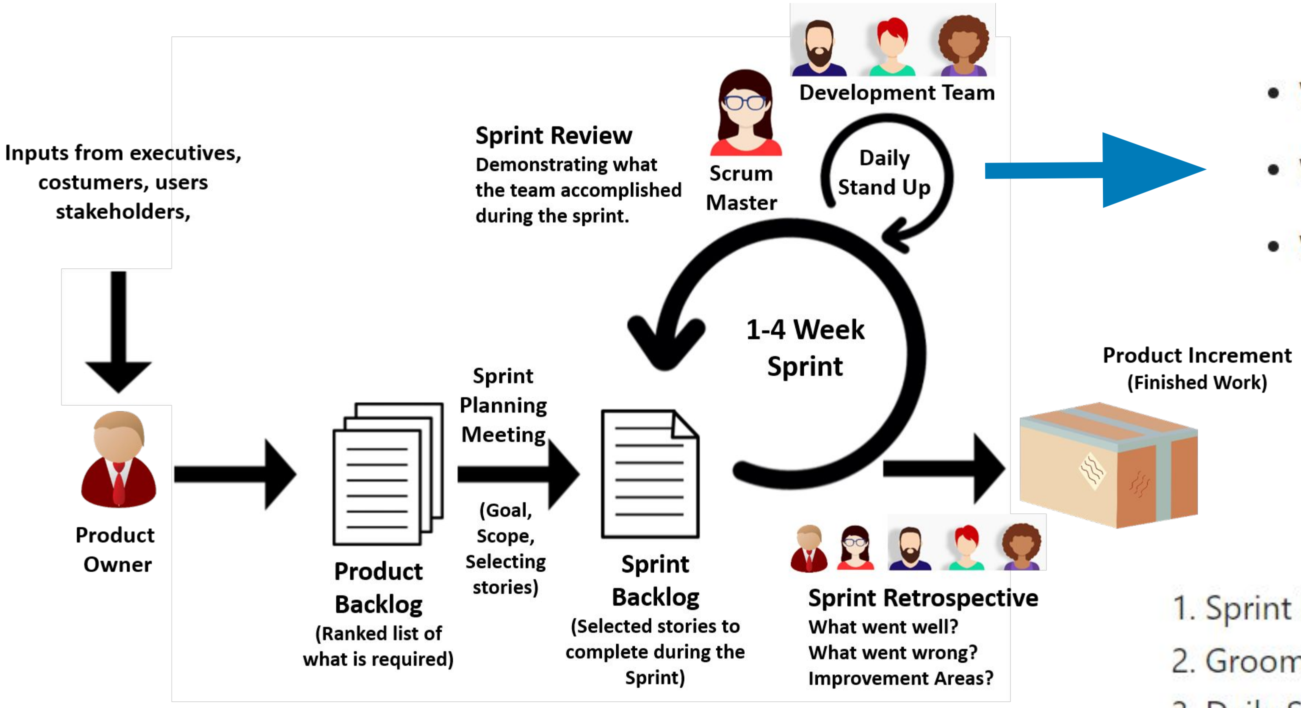
- **A subset of the product backlog.**
- **Not flexible, unchanged during the sprint period.**

Product Increment



- **Sum of all the product backlog items finished during the sprint**

Scrum Ceremonies

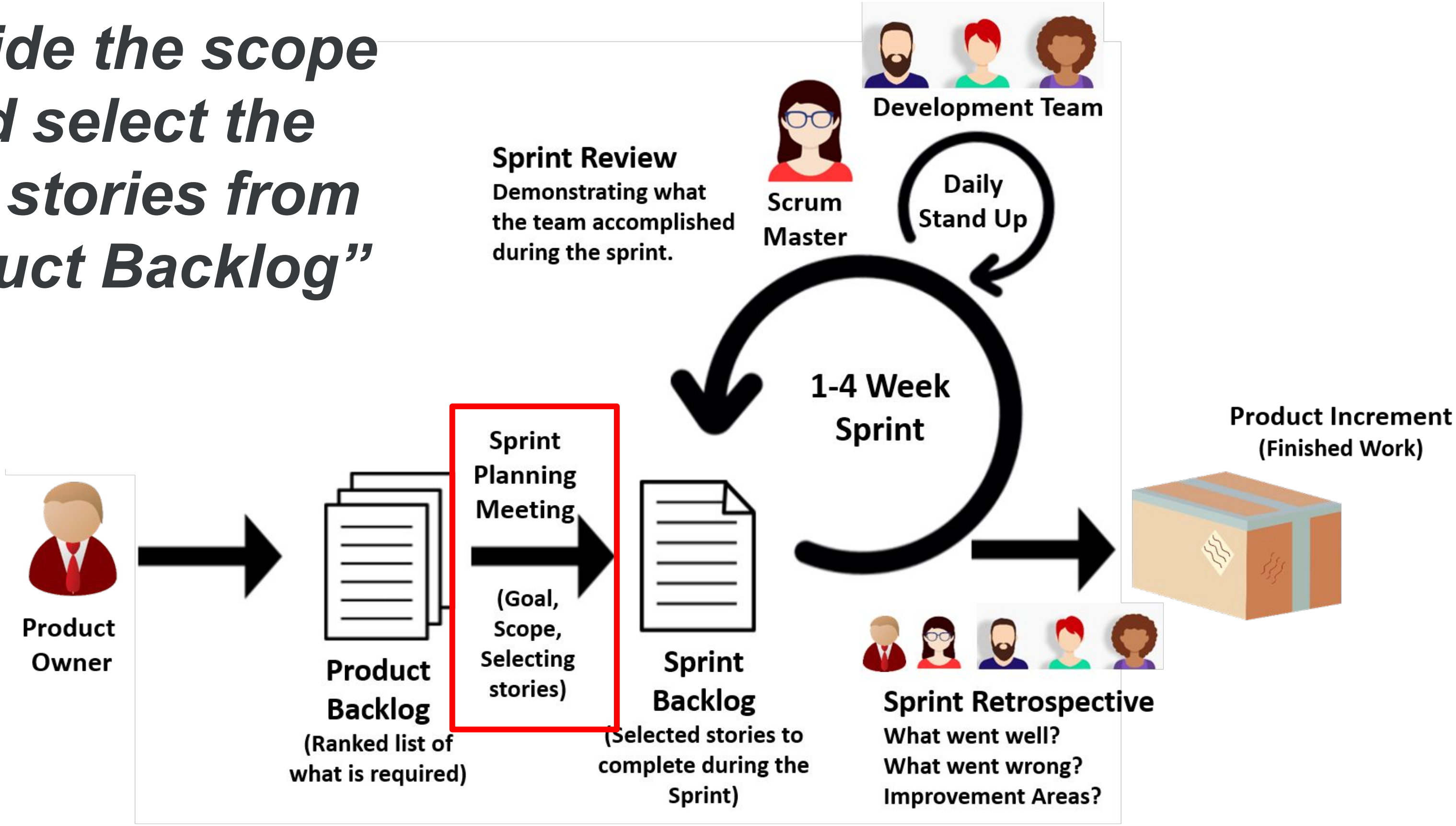


- What they completed yesterday,
- What they will focus on today,
- What are the things impeding them?

1. Sprint Planning Meeting,
2. Grooming Meeting,
3. Daily Stand Up Meeting or Daily Scrum,
4. Sprint Review Meeting,
5. Sprint Retrospective Meeting.

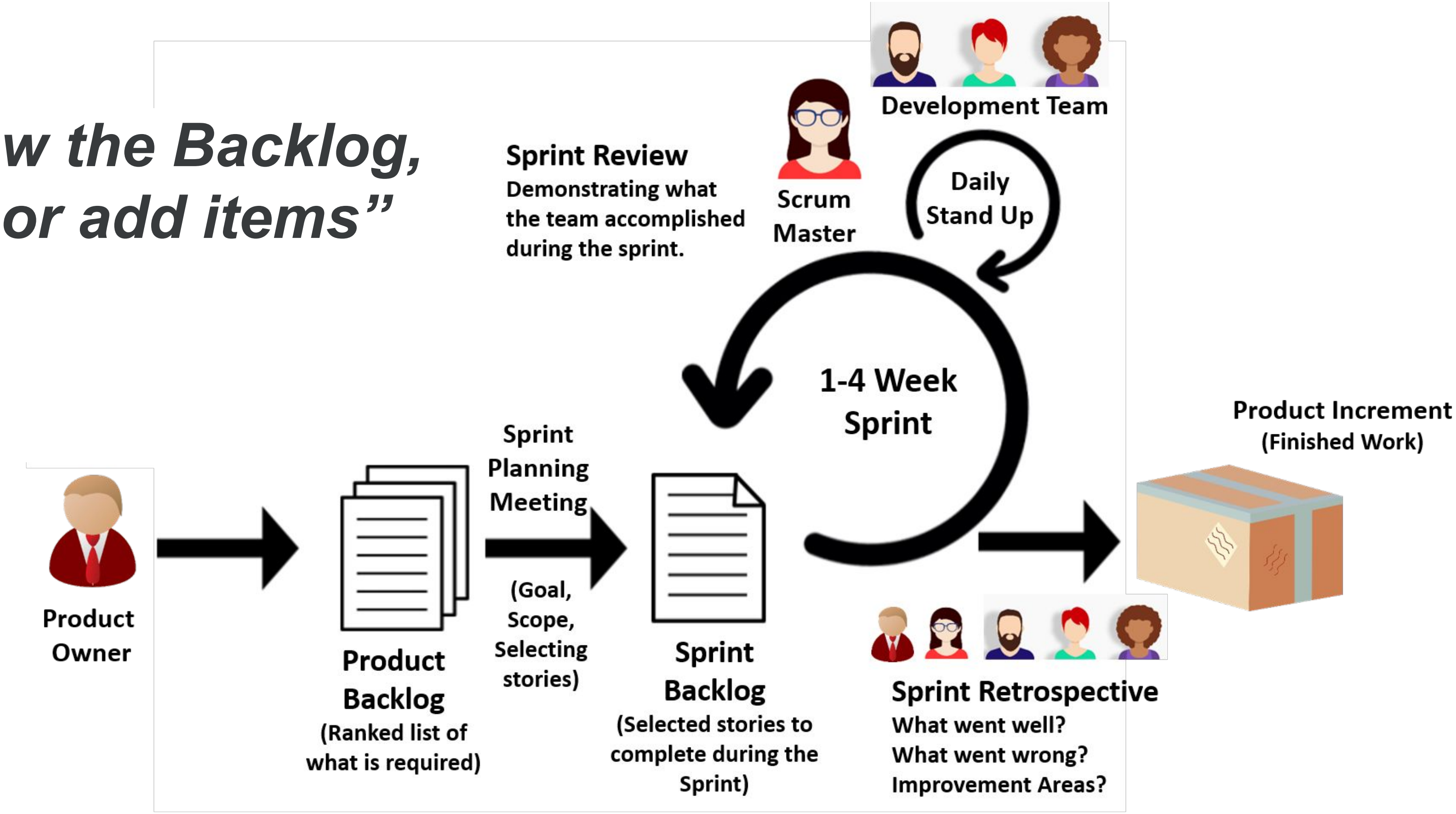
Sprint Planning Meeting

“Decide the scope and select the user stories from Product Backlog”


Development Team
Scrum Master
Sprint Review
Demonstrating what the team accomplished during the sprint.
Daily Stand Up
1-4 Week Sprint
Product Increment (Finished Work)
Product Owner
Product Backlog
(Ranked list of what is required)
Sprint Planning Meeting
(Goal, Scope, Selecting stories)
Sprint Backlog
(Selected stories to complete during the Sprint)
Sprint Retrospective
What went well?
What went wrong?
Improvement Areas?

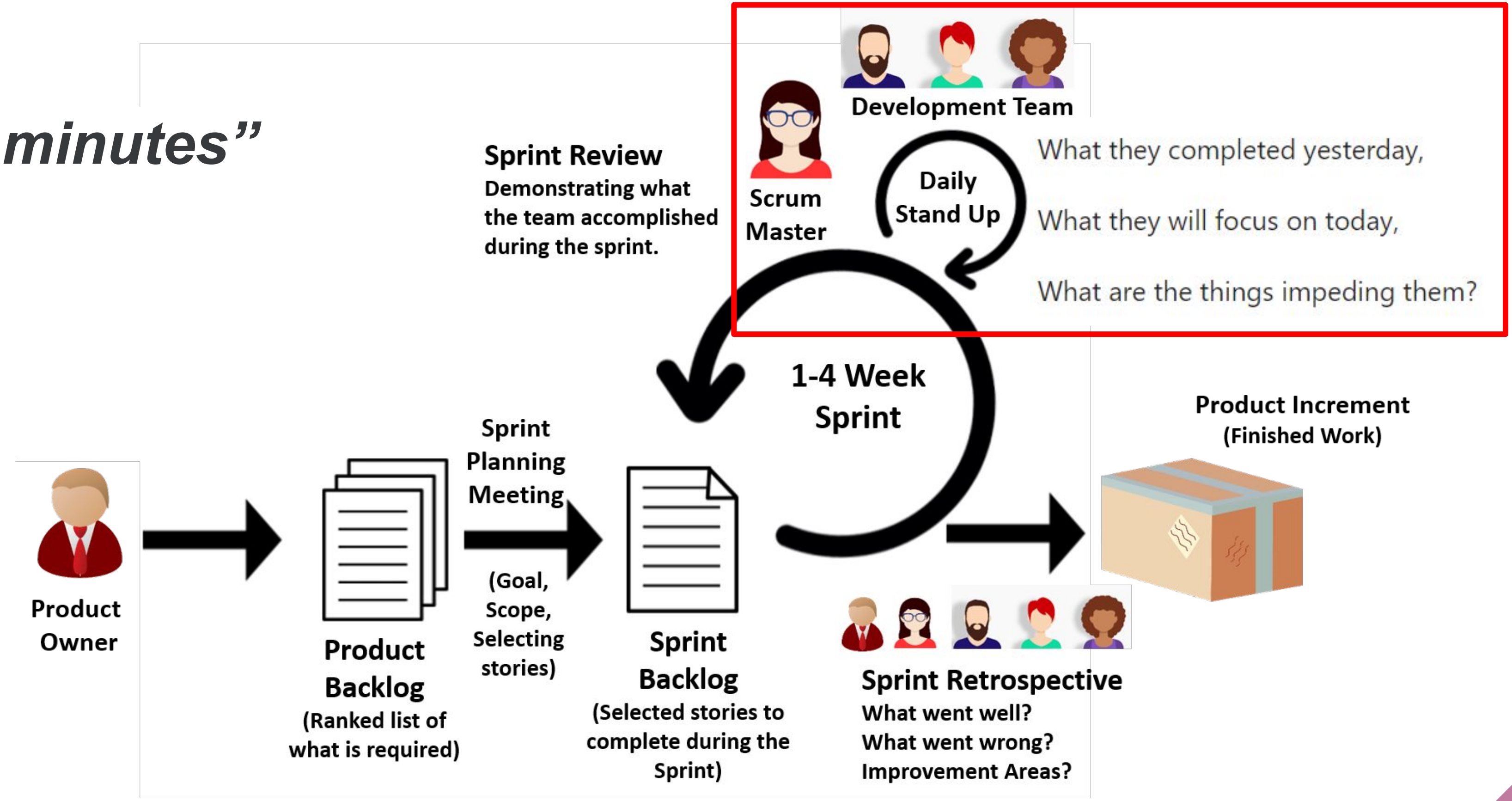
Grooming Meeting

"Review the Backlog, delete or add items"

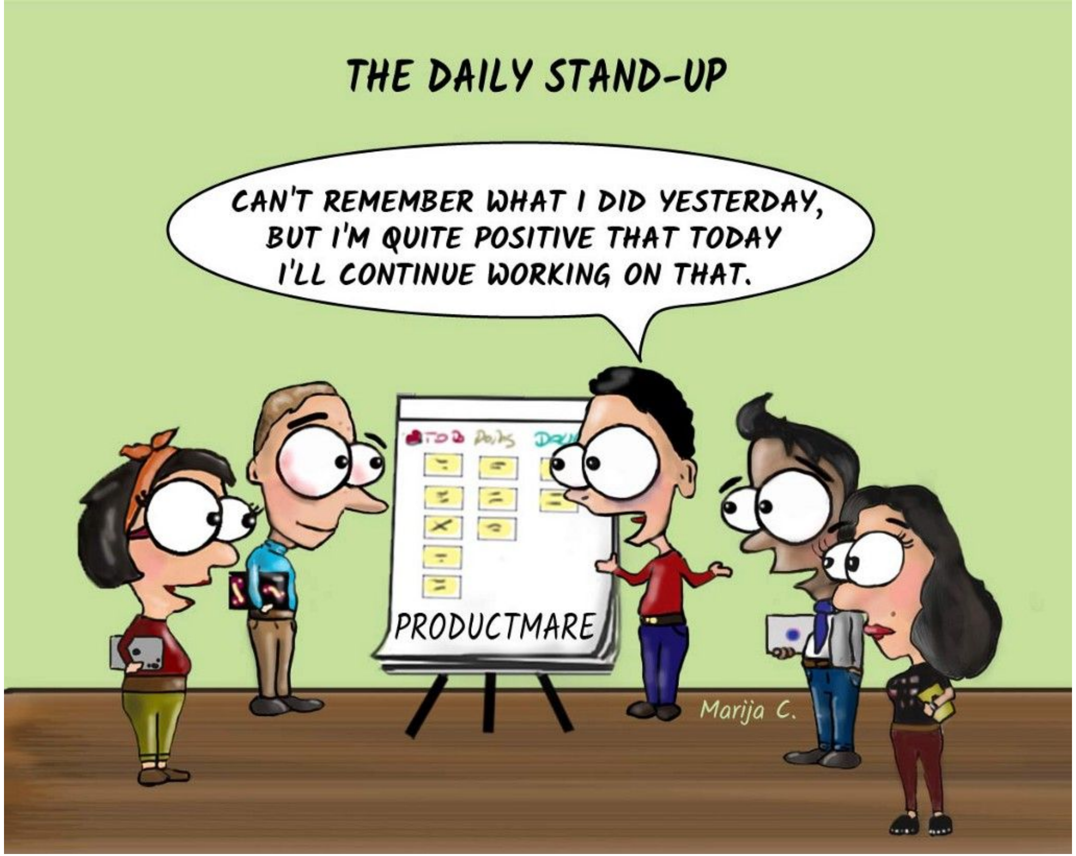


Daily Stand Up Meeting

"15 minutes"

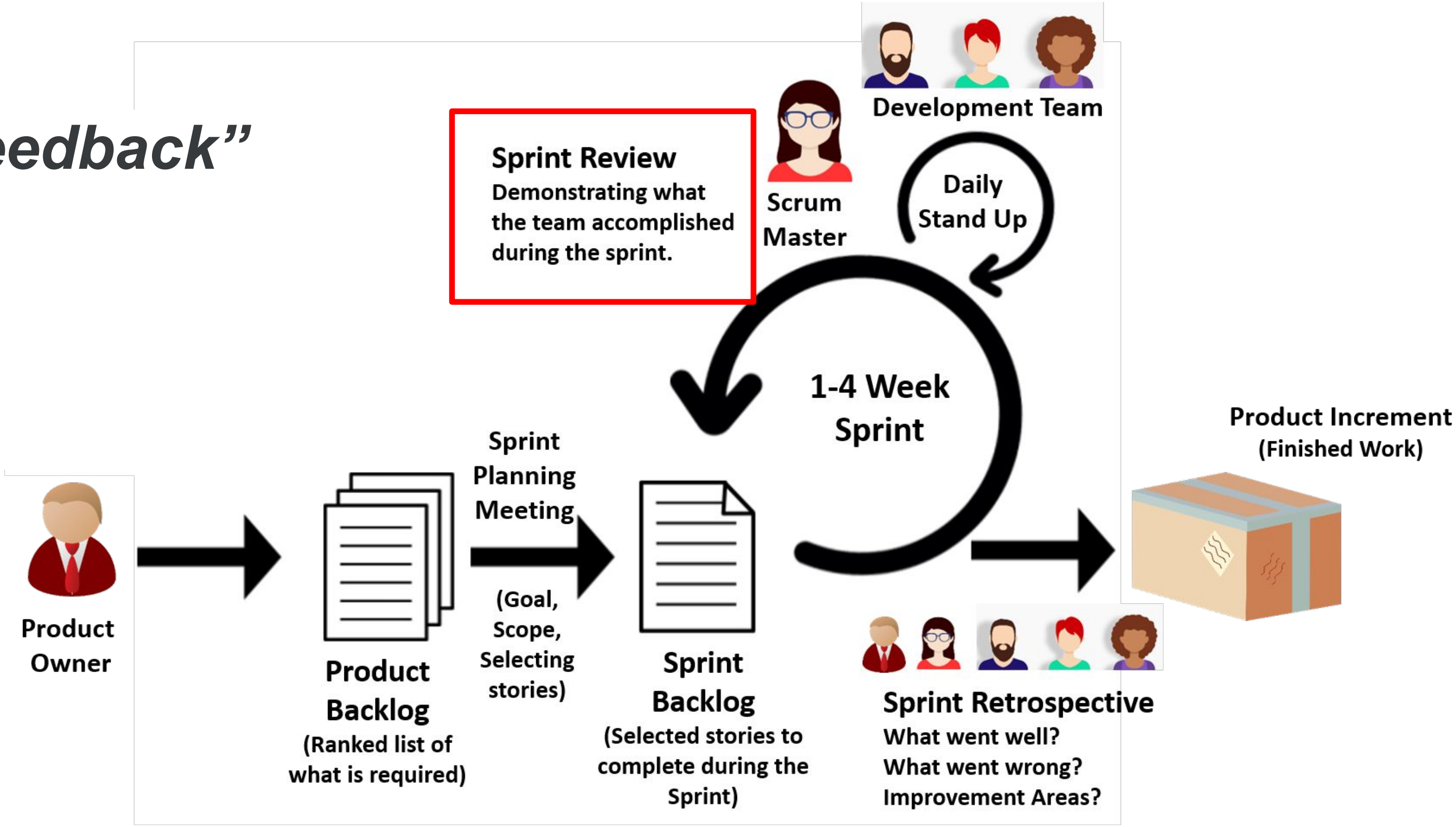

Development Team
Scrum Master
Daily Stand Up
What they completed yesterday,
What they will focus on today,
What are the things impeding them?
Sprint Review
Demonstrating what the team accomplished during the sprint.
1-4 Week Sprint
Product Increment (Finished Work)
Product Owner
Product Backlog (Ranked list of what is required)
Sprint Planning Meeting
(Goal, Scope, Selecting stories)
Sprint Backlog (Selected stories to complete during the Sprint)
Sprint Retrospective
What went well?
What went wrong?
Improvement Areas?

Daily Stand Up Meeting


THE DAILY STAND-UP
CAN'T REMEMBER WHAT I DID YESTERDAY, BUT I'M QUITE POSITIVE THAT TODAY I'LL CONTINUE WORKING ON THAT.
TODO Doing Done
PRODUCTMARE
Marija C.

Sprint Review Meeting

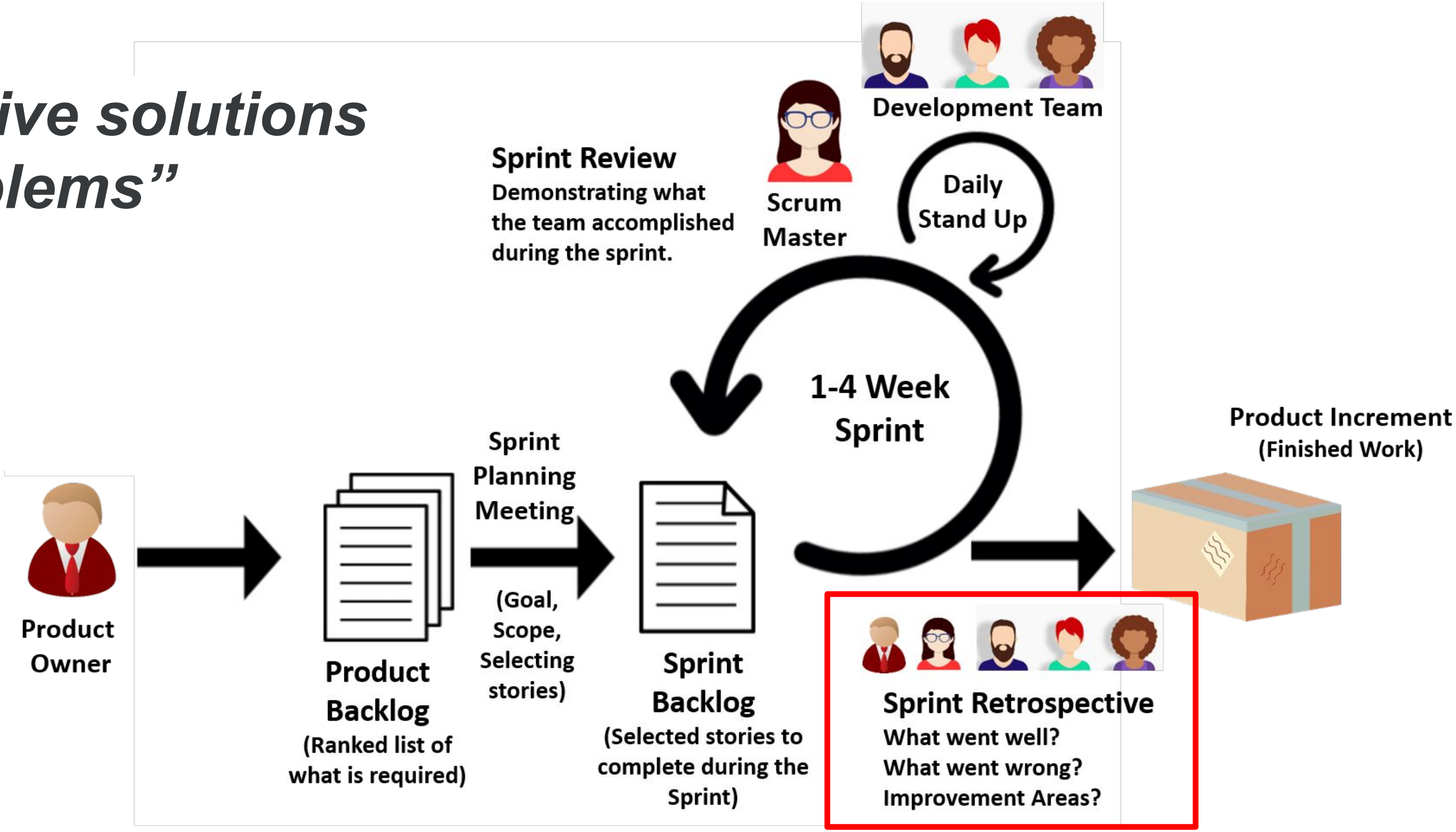
"Get Feedback"



Sprint Review
Demonstrating what the team accomplished during the sprint.
Scrum Master
Development Team
Daily Stand Up
1-4 Week Sprint
Product Increment (Finished Work)
Product Owner
Product Backlog (Ranked list of what is required)
Sprint Planning Meeting
(Goal, Scope, Selecting stories)
Sprint Backlog (Selected stories to complete during the Sprint)
Sprint Retrospective
What went well?
What went wrong?
Improvement Areas?

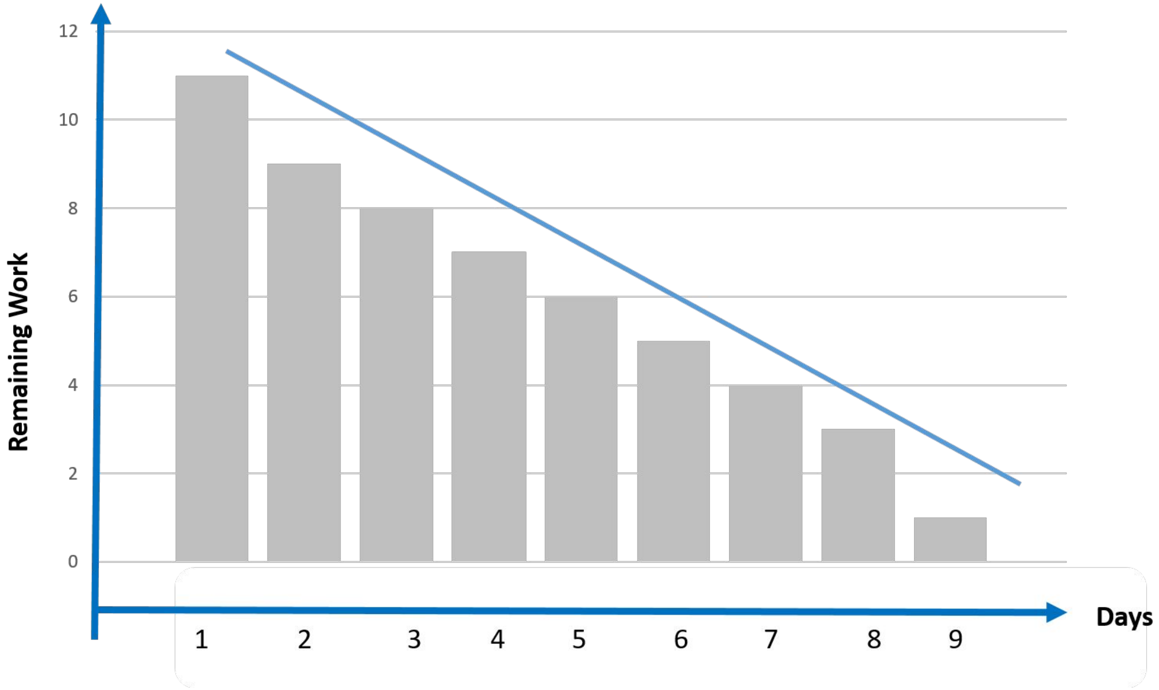
Sprint Retrospective Meeting

“Effective solutions to problems”



Development Team
Sprint Review
Demonstrating what the team accomplished during the sprint.
Scrum Master
Daily Stand Up
1-4 Week Sprint
Product Increment (Finished Work)
Sprint Planning Meeting
(Goal, Scope, Selecting stories)
Product Owner
Product Backlog
(Ranked list of what is required)
Sprint Backlog
(Selected stories to complete during the Sprint)
Sprint Retrospective
What went well?
What went wrong?
Improvement Areas?

Burndown Chart



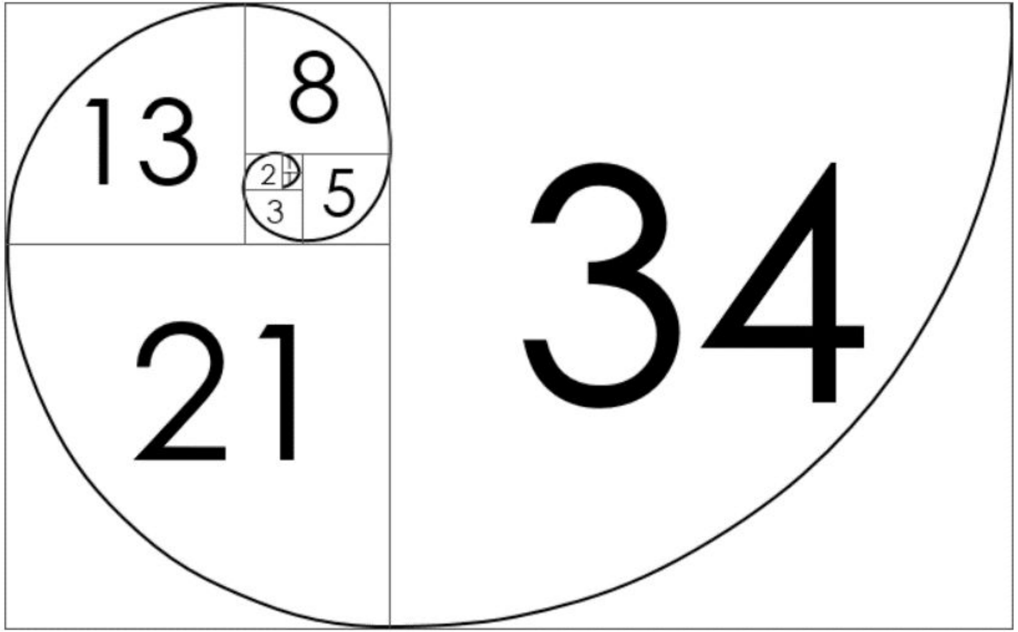
Graphical demonstration of remaining work versus time.

Timeboxing

A timebox is a period of time in which a team works to achieve a goal.

Instead of waiting until the goal is reached, the timeboxing approach STOPS when the time limit is reached.

Story Points

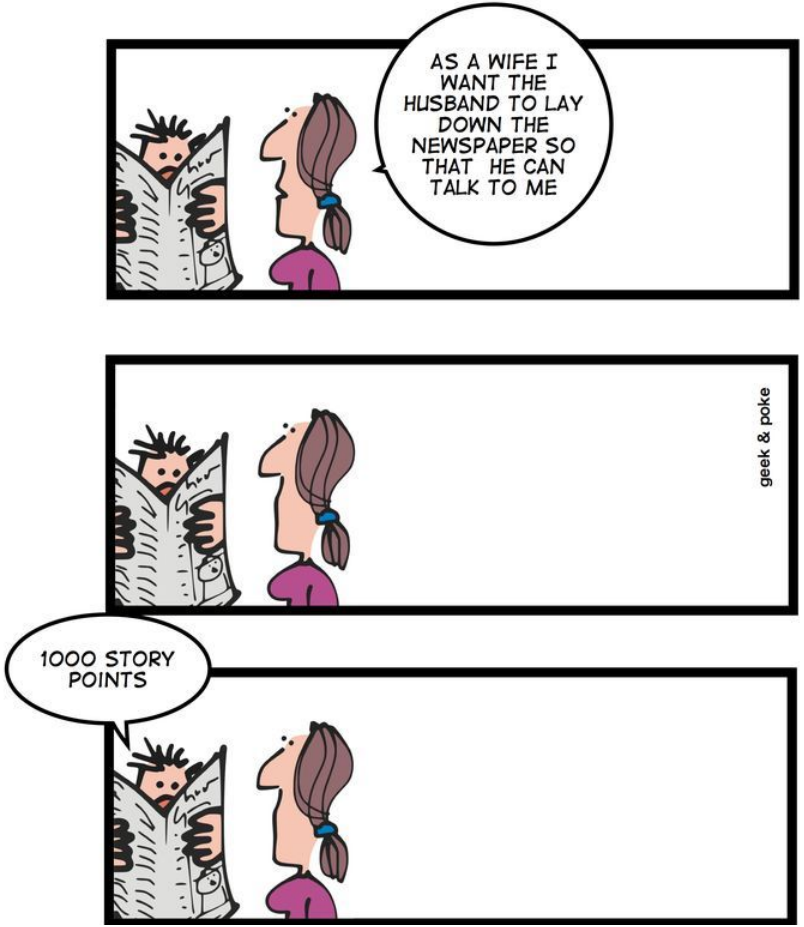


- Numeric values that indicate the difficulty level of the user stories

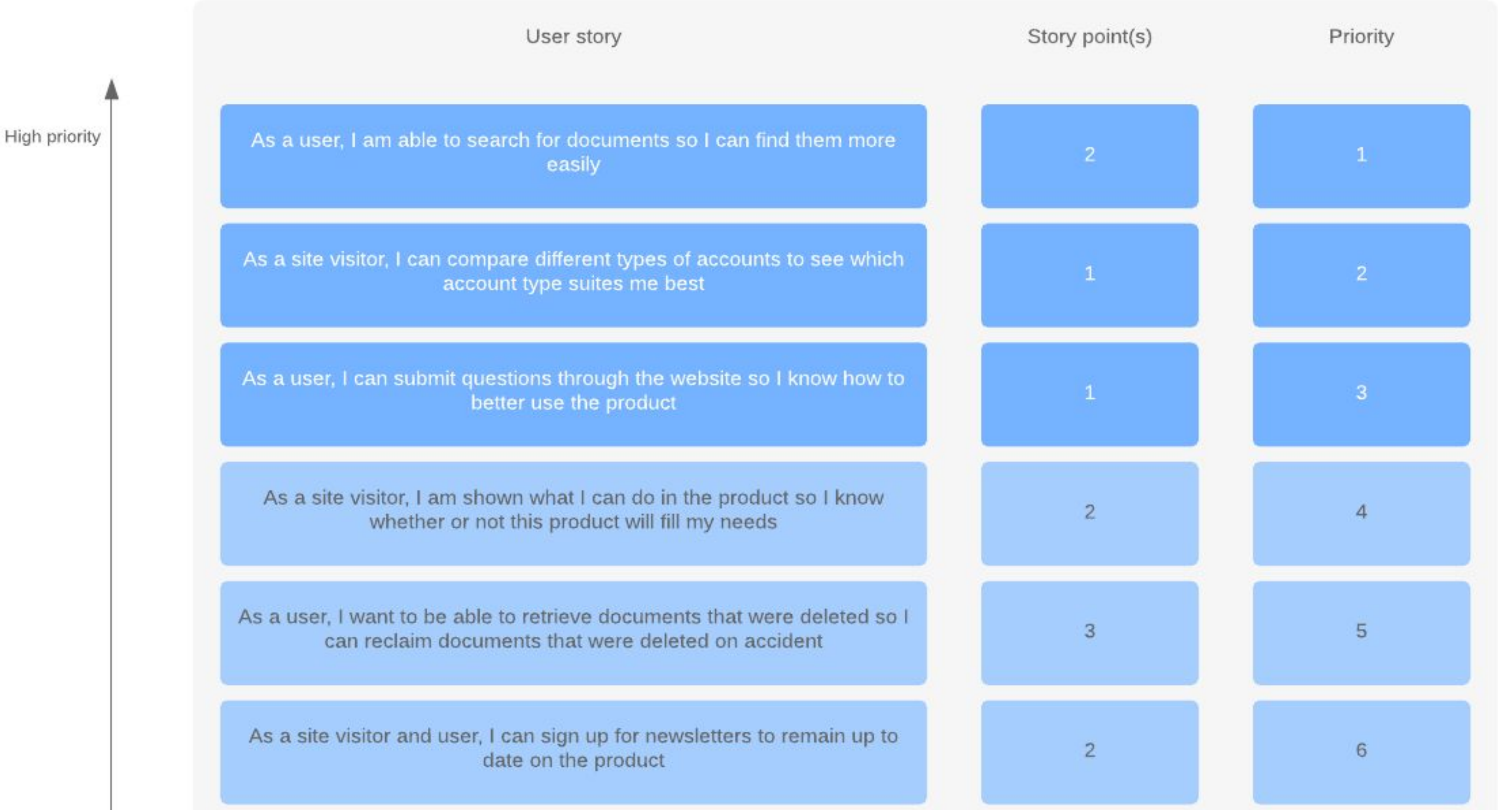
- Assigned to each user story using the Fibonacci numbers

Story Points

Story point describes the difficulty level !


AS A WIFE I WANT THE HUSBAND TO LAY DOWN THE NEWSPAPER SO THAT HE CAN TALK TO ME
geek & poke
1000 STORY POINTS

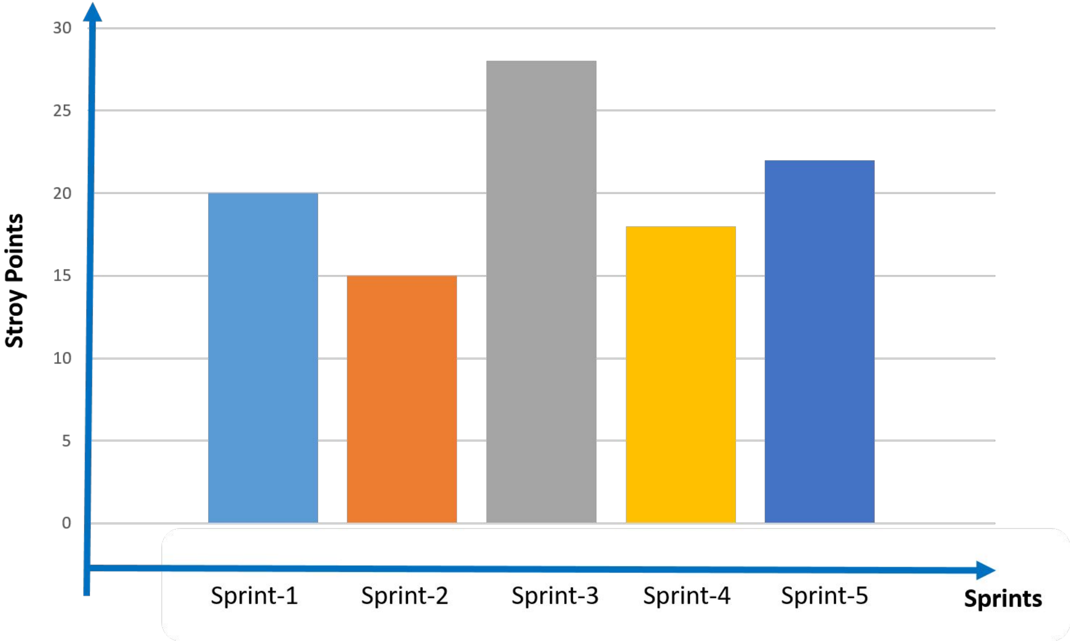
Story Points in a Product Backlog



High priority

User story	Story point(s)	Priority
As a user, I am able to search for documents so I can find them more easily	2	1
As a site visitor, I can compare different types of accounts to see which account type suites me best	1	2
As a user, I can submit questions through the website so I know how to better use the product	1	3
As a site visitor, I am shown what I can do in the product so I know whether or not this product will fill my needs	2	4
As a user, I want to be able to retrieve documents that were deleted so I can reclaim documents that were deleted on accident	3	5
As a site visitor and user, I can sign up for newsletters to remain up to date on the product	2	6

Velocity

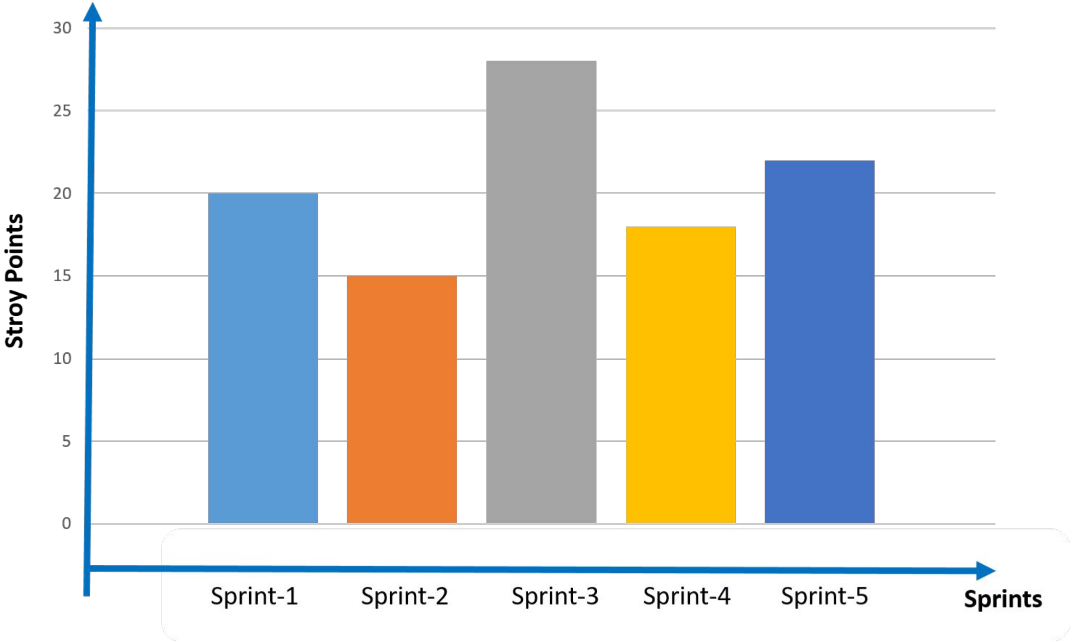


- Key metric in scrum

- Summing up all story points at the end of each sprint

- Making an estimation of how long the project will take to complete.

Velocity



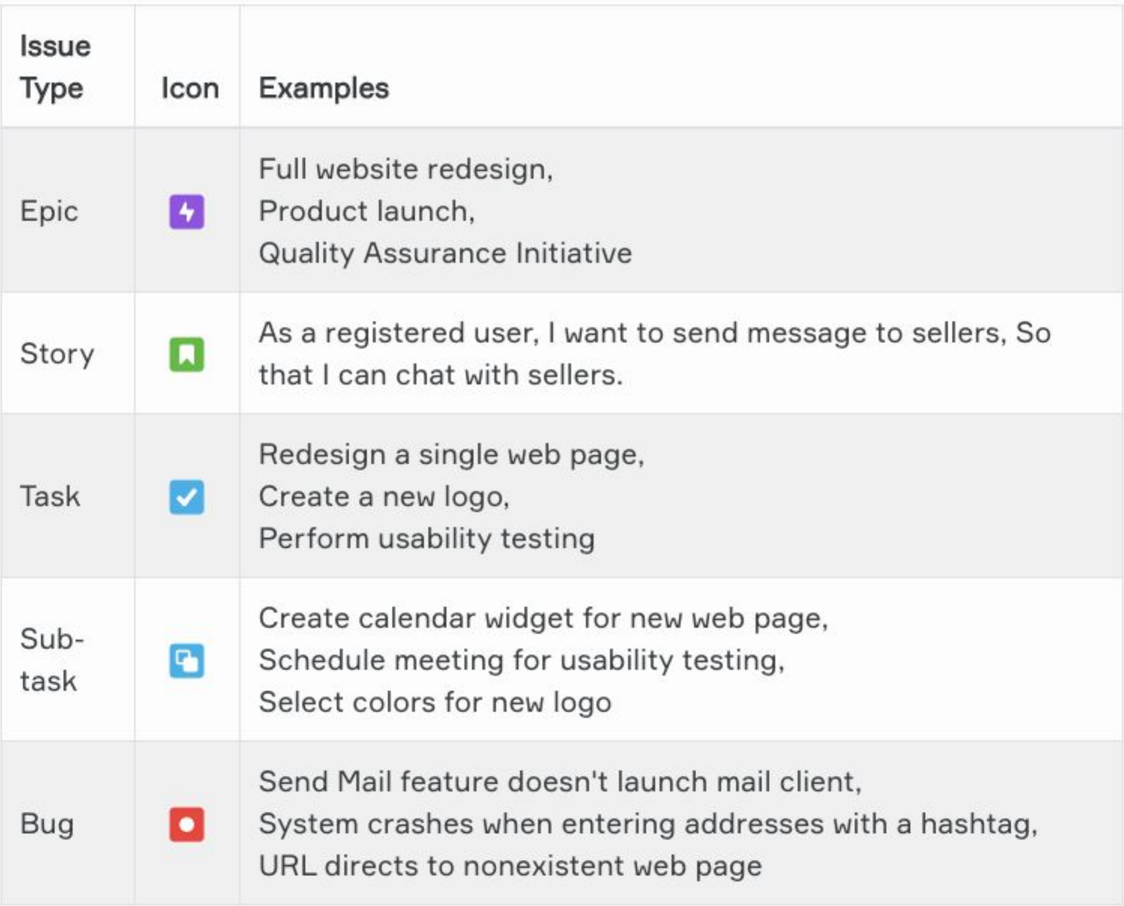
What do you think?

Students choose an option

JIRA

DERS NOTLARI

QUESTIONS



Issue Type	Icon	Examples
Epic	⚡	Full website redesign, Product launch, Quality Assurance Initiative
Story	🔖	As a registered user, I want to send message to sellers, So that I can chat with sellers.
Task	☑	Redesign a single web page, Create a new logo, Perform usability testing
Sub-task	⧉	Create calendar widget for new web page, Schedule meeting for usability testing, Select colors for new logo
Bug	●	Send Mail feature doesn't launch mail client, System crashes when entering addresses with a hashtag, URL directs to nonexistent web page

QUESTIONS

What is Python?

What is PEP 8?

Aşağıdakilerden hangileri PEP 8 convention ‘a göre doğrudur

	A	B
I.	print('continent')	print ('continent')
II.	if y == 3 : print x , y ; x , y = y , x	if y == 3: print x, y; x, y = y, x
III.	df[0,] or foo = (2,)	df[0,] or foo = (2,)

QUESTIONS

What are the comments and how do you write it in Python?

Why are giving variable name and write comment on your code so important?

QUESTIONS

What are the numerical data types in Python and their properties?

Describe the Boolean types in detail.

QUESTIONS

6. Aşağıdaki kodların çıktıları nedir?

a. print(type(32.6))

b. print(type(True))

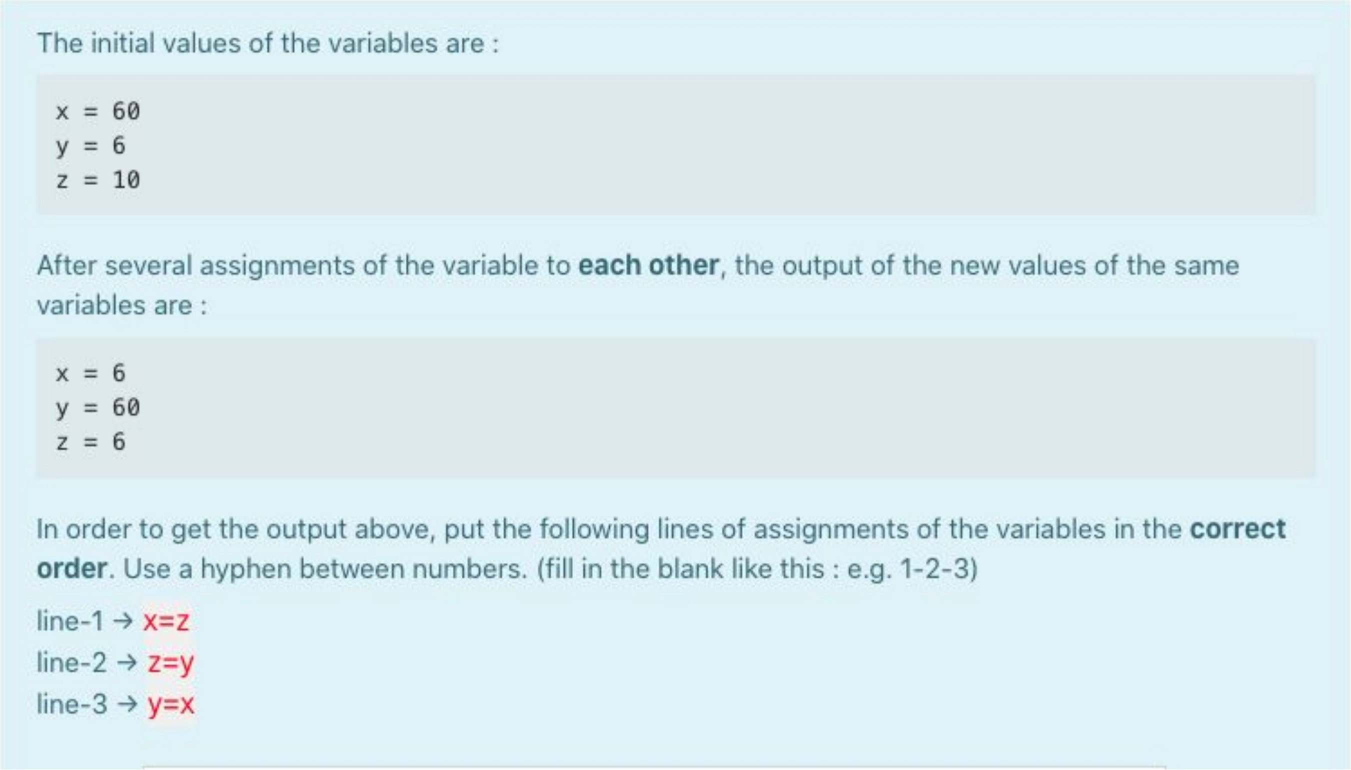
c. print(type(‘True’))

d. print(type(‘56.8’))

e. print(type(3+4j))

QUESTIONS

Uygulanacak dönüşümün sırasını yazınız



The initial values of the variables are :

```
x = 60
y = 6
z = 10
```

After several assignments of the variable to **each other**, the output of the new values of the same variables are :

```
x = 6
y = 60
z = 6
```

In order to get the output above, put the following lines of assignments of the variables in the **correct order**. Use a hyphen between numbers. (fill in the blank like this : e.g. 1-2-3)

line-1 → x=z

line-2 → z=y

line-3 → y=x

LINUX

1. Windows Subsystem for Linux Installation Guide for Windows 10
2. Linux CENTOS 7 online without installing
3. Linux with Docker Container