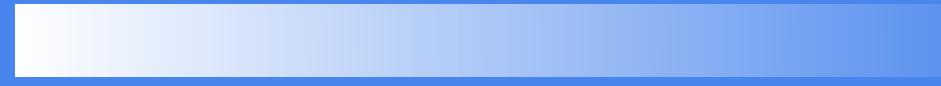


Today's Agenda



- Types of the companies in IT
- Types of the employment
- Types of people on the project
- How to become great in soft skills
- Intro to SDLC



TYPES OF IT COMPANIES

most common business
models



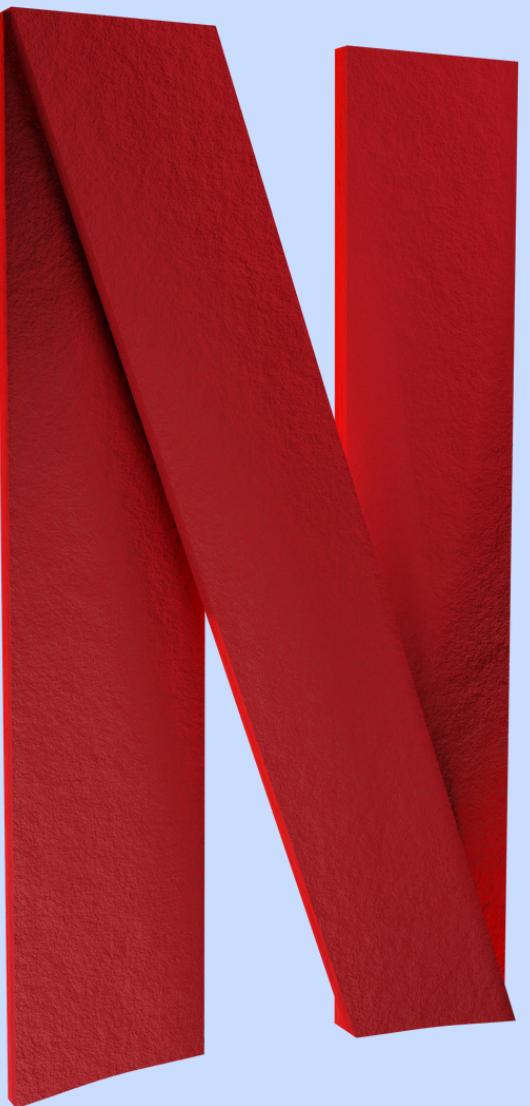
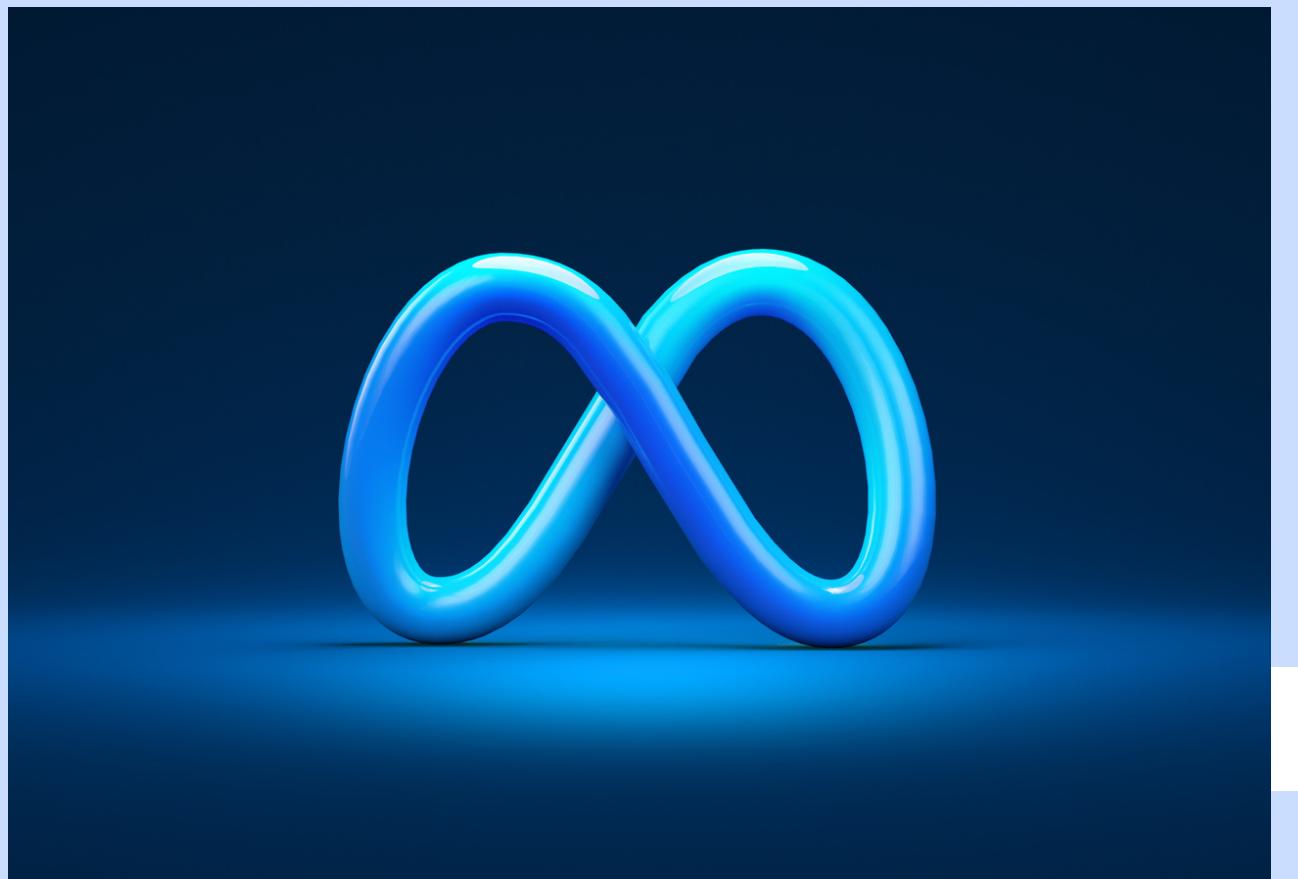
Soft skill is about talking, expressing, and connecting. You do these with confidence. Confidence comes from knowledge



Product-based

The companies whose focus is based solely on their own tech product and projects.

- New tech stacks
- updated tools and libraries
- Quality



• •
• •
• •
• •
• •
• •
• •
• •



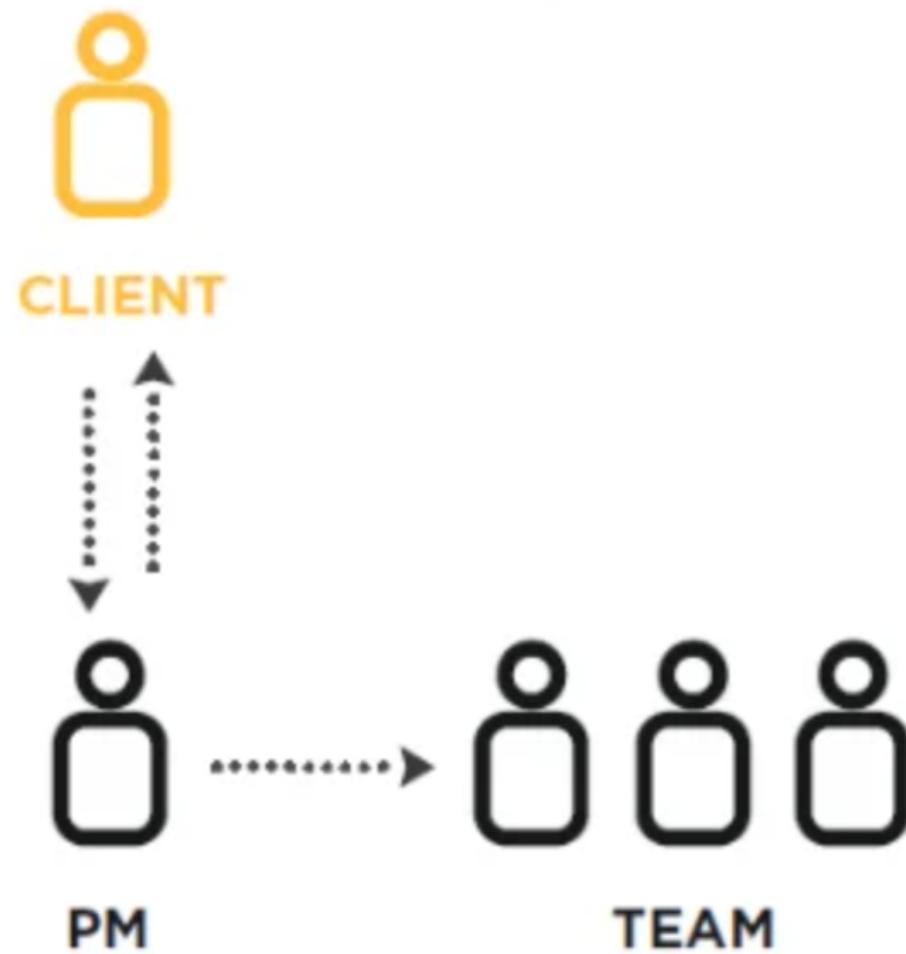
Outsourcing

The companies who get the full or partial project to develop from a client

- Onshore
- Offshore
- Hybrid



Outsourcing Model



Outstaff companies

Companies who have an agreement with a client to add team members to projects

- Onshore
- Offshore
- Hybrid



Deloitte.

Outstaffing Model



Types of the employment

- Recruiting layer
- Full time
- Contractor
- Contract to hire



Types of the employment

Recruiting

The company that has an agreement to find a candidate for the client's open positions.



Note: They get paid from the client. They are not technical



Types of the employment

Full-time job

You get hired directly by
a company.

- W2 employment
- Benefits
- No end of the contract
- Compensation annually

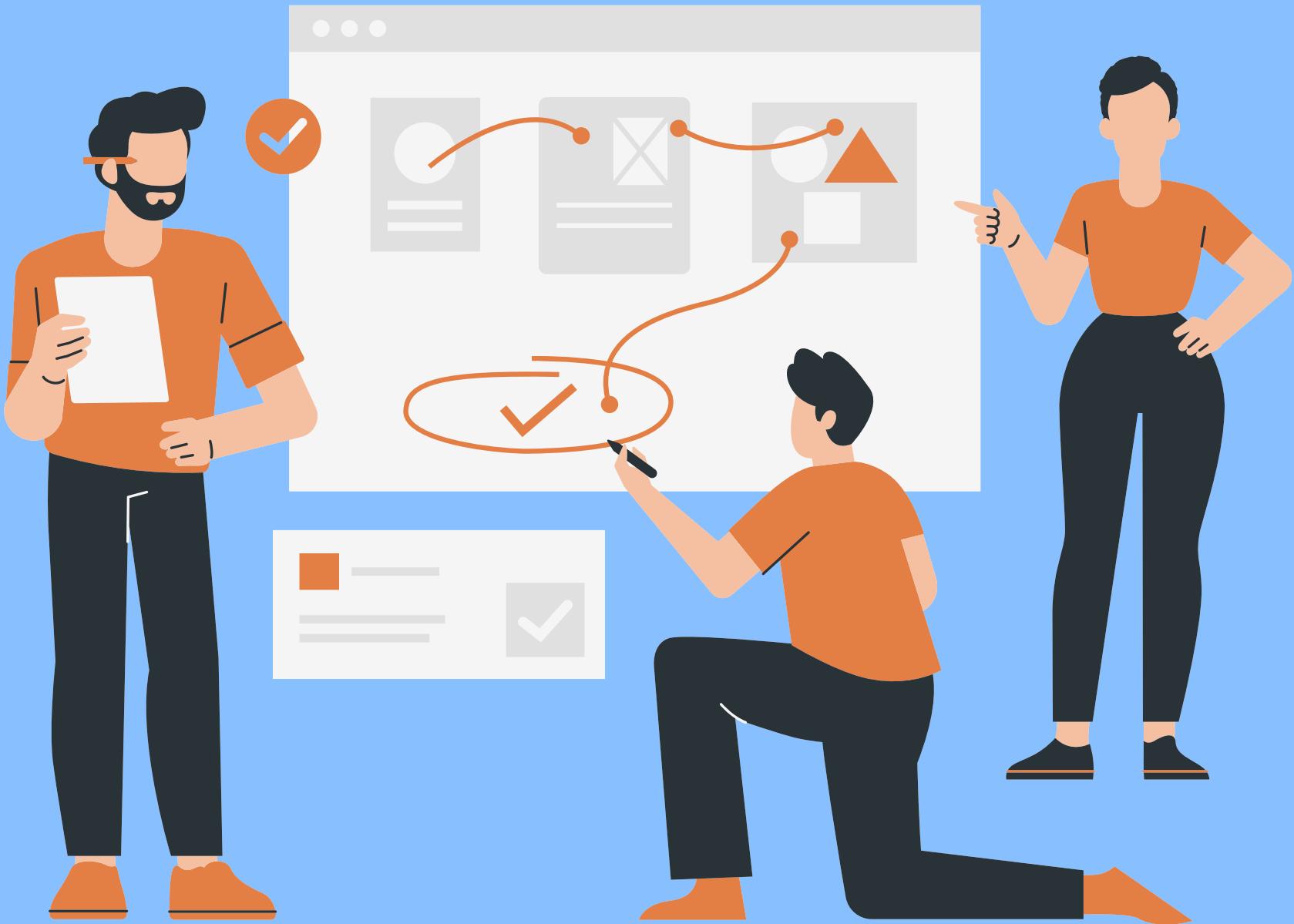


Types of the employment

Contractor/Vendor

You get hired by consulting company to work for a client

- 1099
- Benefits/no benefits
- Contract end date
- Compensation hourly



Types of the employment

Contract to hire

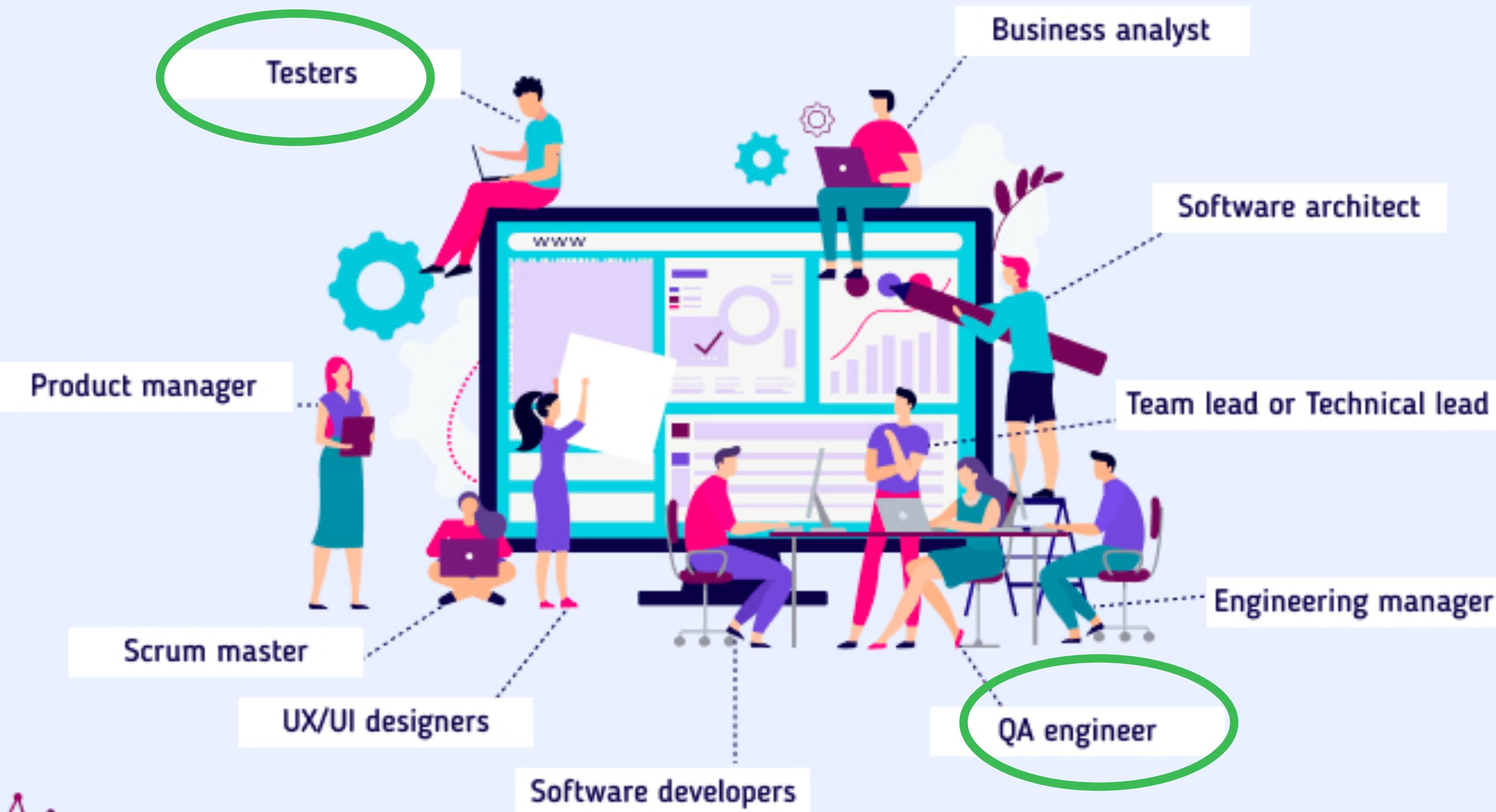
You get hired on a contract
based on a short period,
then switch fulltime

- Performance check



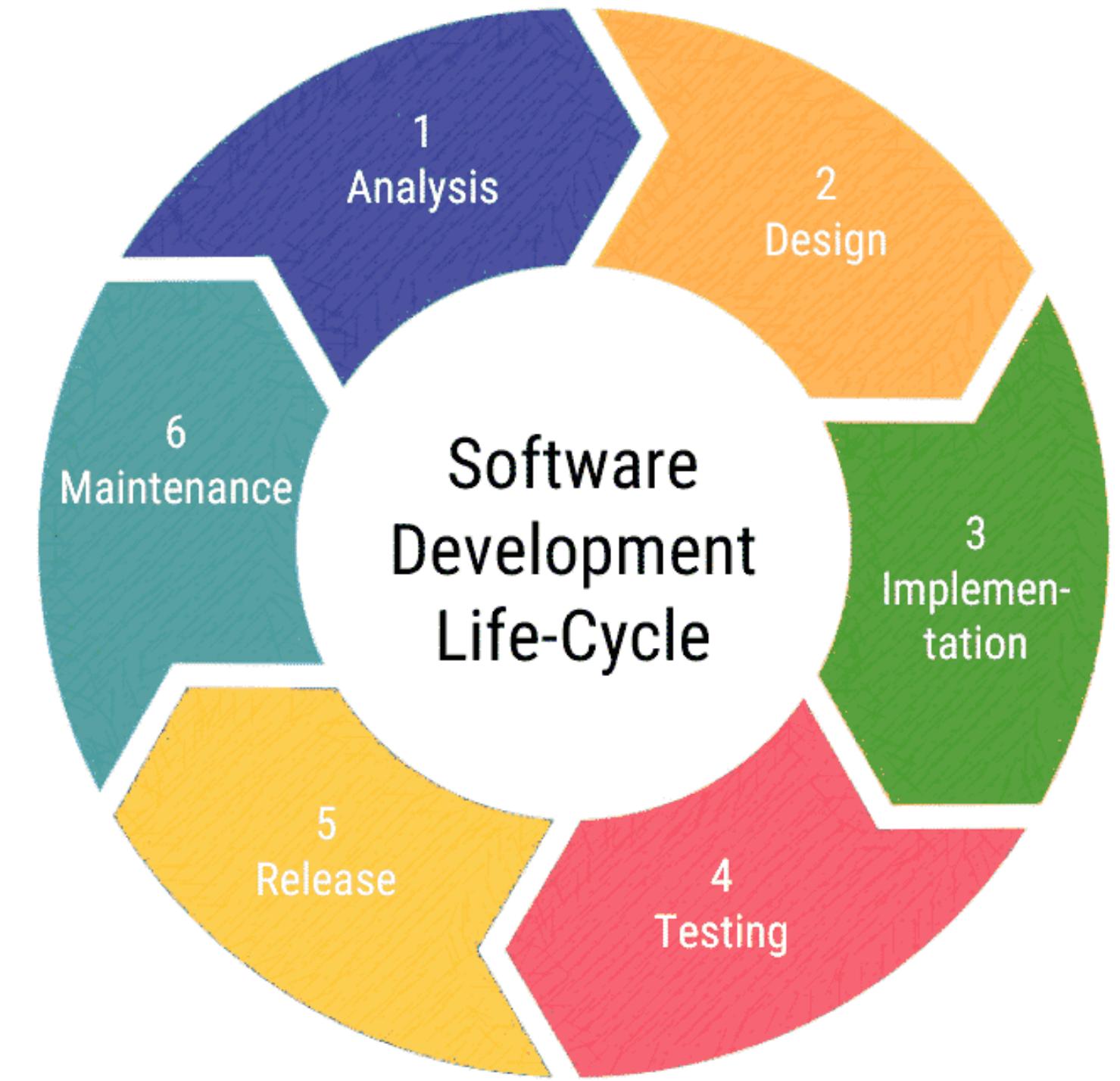
People on the project

Your perfect development team



SDLC **Software** **development lifecycle**

**The process of
building the software**



Analysis

Planning the software and gathering the requirement.

- Feasibility study
- competitor research
- SRS. Software requirement specification
- Performed by the product owner, business analyst, domain experts and technical management



Design and architecture

A senior Software developer or software architect(1 or more people) come up with the best possible technical design to build this software

- At this phase, no coding is involved. Only technical decisions, a reliable tech stack, and efficient solution is offered by the architect
- DDS: Design document specification. For documentation



Development, Implementation



Here, the development team does the coding

- Once the tech stack decision is made, development starts
- UI/UX designer does designs, layout
- Backend/Frontend development
- Mobile development
- DevOps engineer setups

Testing

Once developers build software, it's time for quality testing

- Before going to real users, testers use it fully as users and test the software
- There are 2 types of testers
- Manual Tester: usually nontechnical.
Tests all scenarios and user behaviors manually
- SDET: automates the process of testing



Production/Release

If Testers approve the quality of the software and tell the team that the software is good to go then the Software moves to **Production**



- Usually, there is a separate team who handles the production. DevOps and Release team
- Night time when users don't use
- Right after going to Production, testers still test to make sure nothing goes wrong

Software development methodologies

There are few development models practiced

- Agile

Quickly circling all phases with a limited number of items. Based on priorities

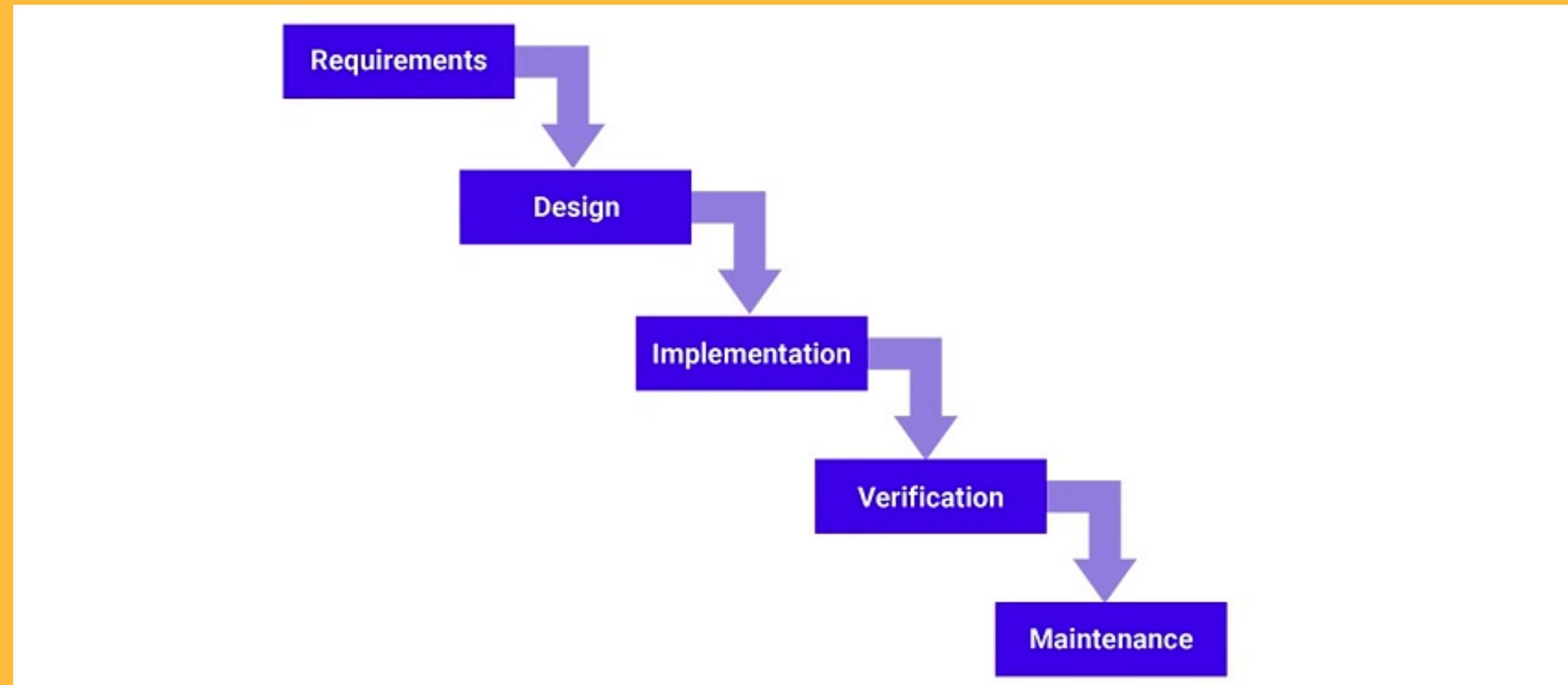
- Waterfall

Finishing all items one by one and moving the next phase

Software development methodologies

- Waterfall

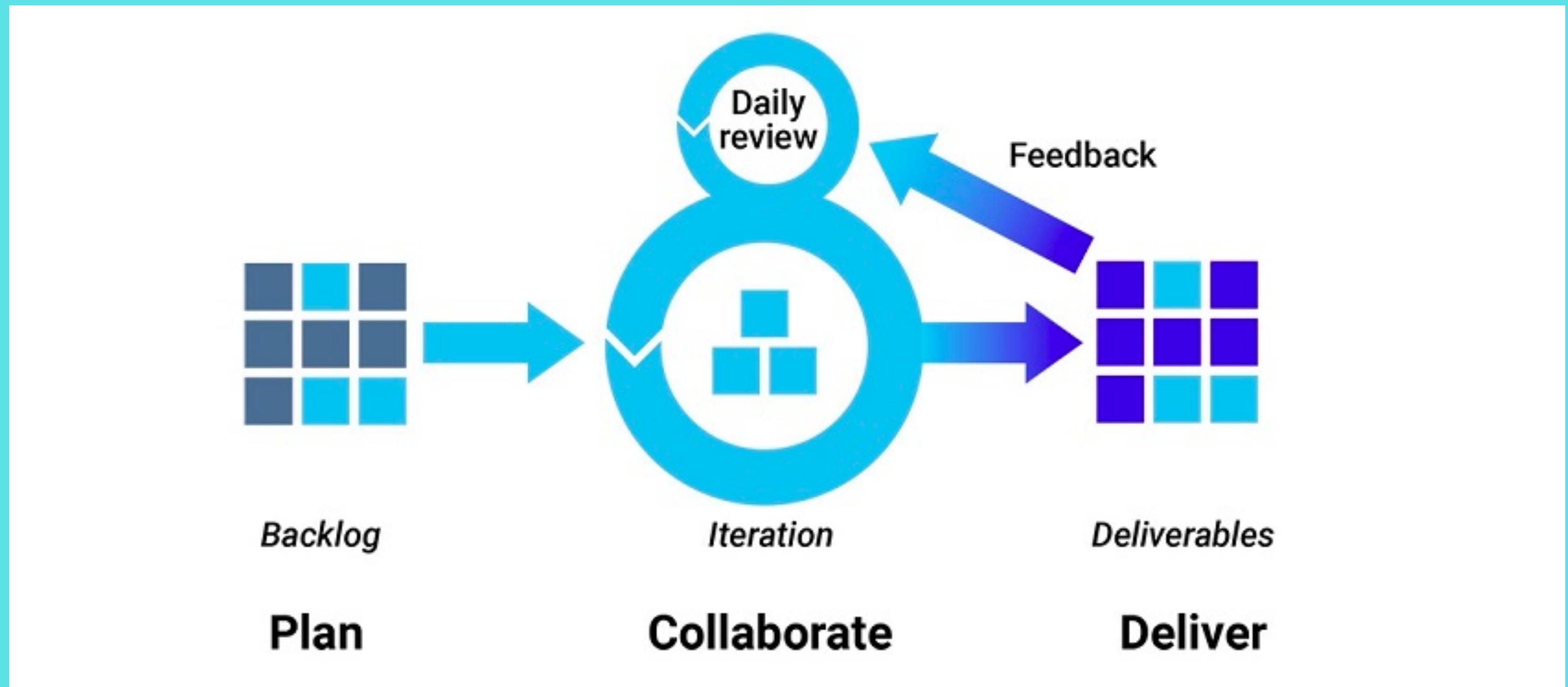
Finishing all items one by one and moving to the next phase



Software development methodologies

- Agile

Quickly circling all phases with a limited number of items. Based on priorities



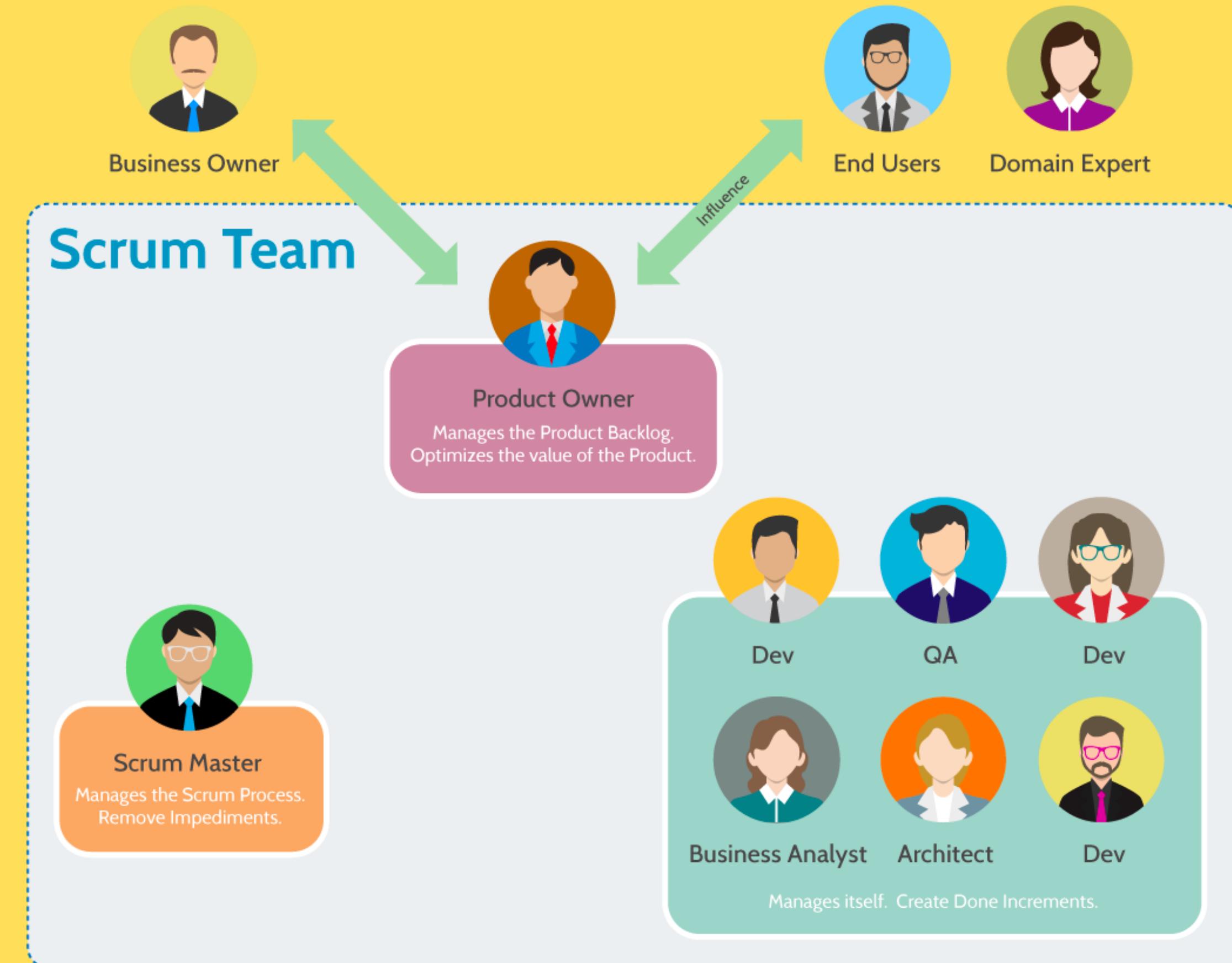
Agile | Scrum

- **Agile** is a Software development methodology that focuses on delivering software through rapid changes with cross-functional teams.
- **Scrum**. It is a process framework for Agile development methodology and the most widely-used one.

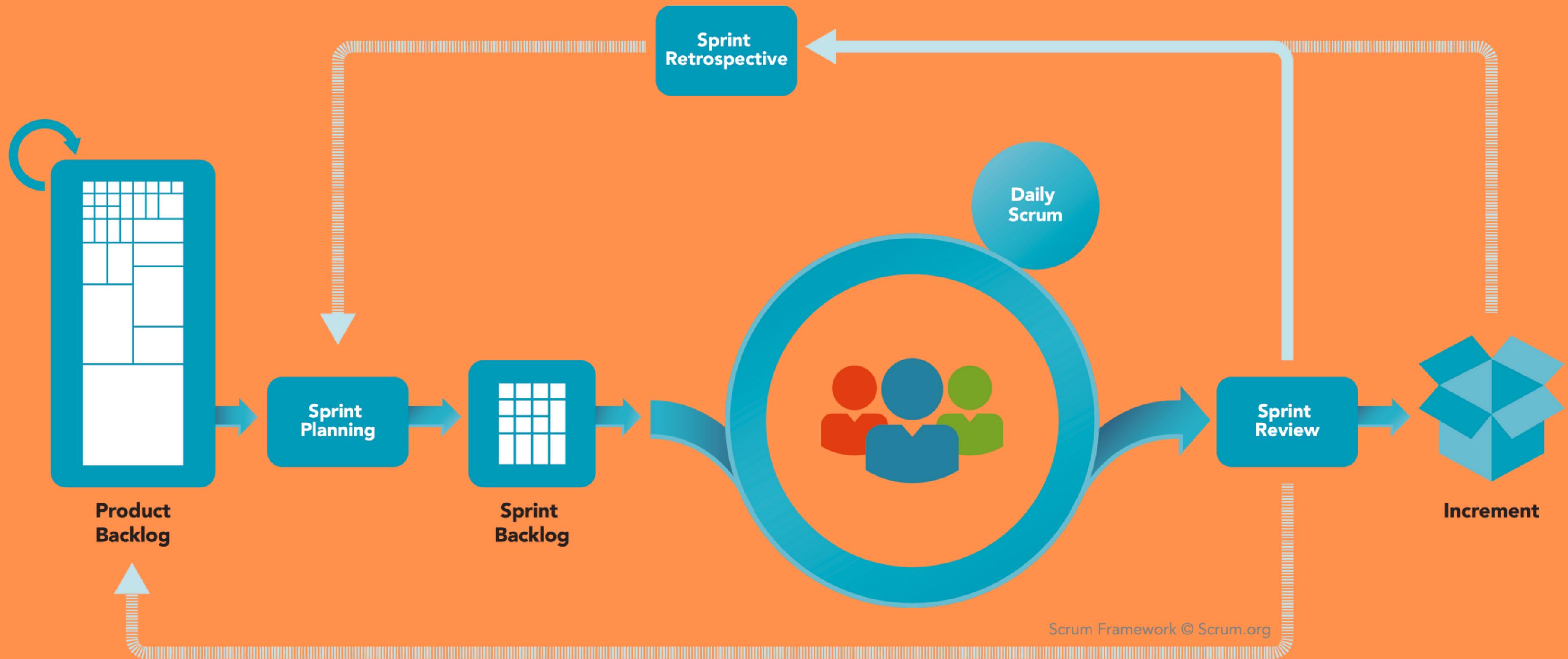


Agile | Scrum

- Scrum team:
a team that
does the
development



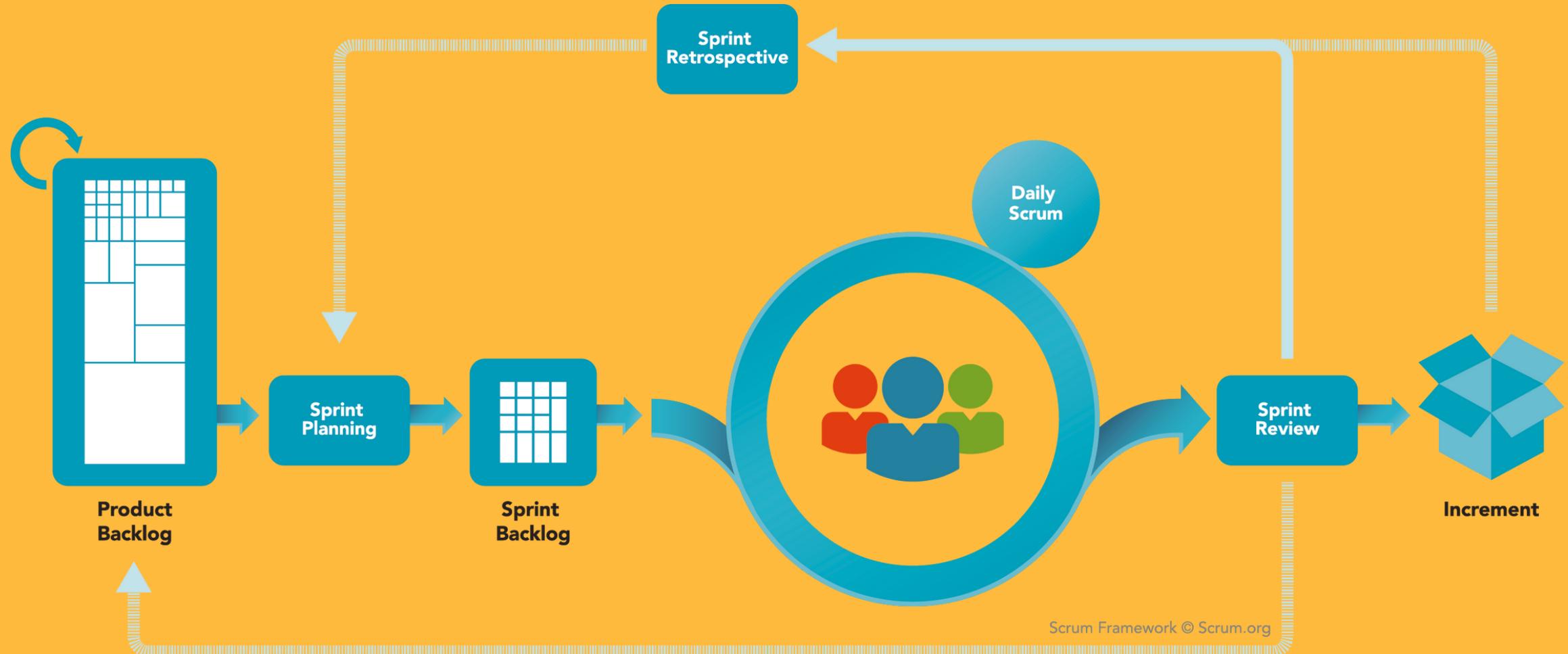
Agile | Scrum



Agile | Scrum

- **Sprint**

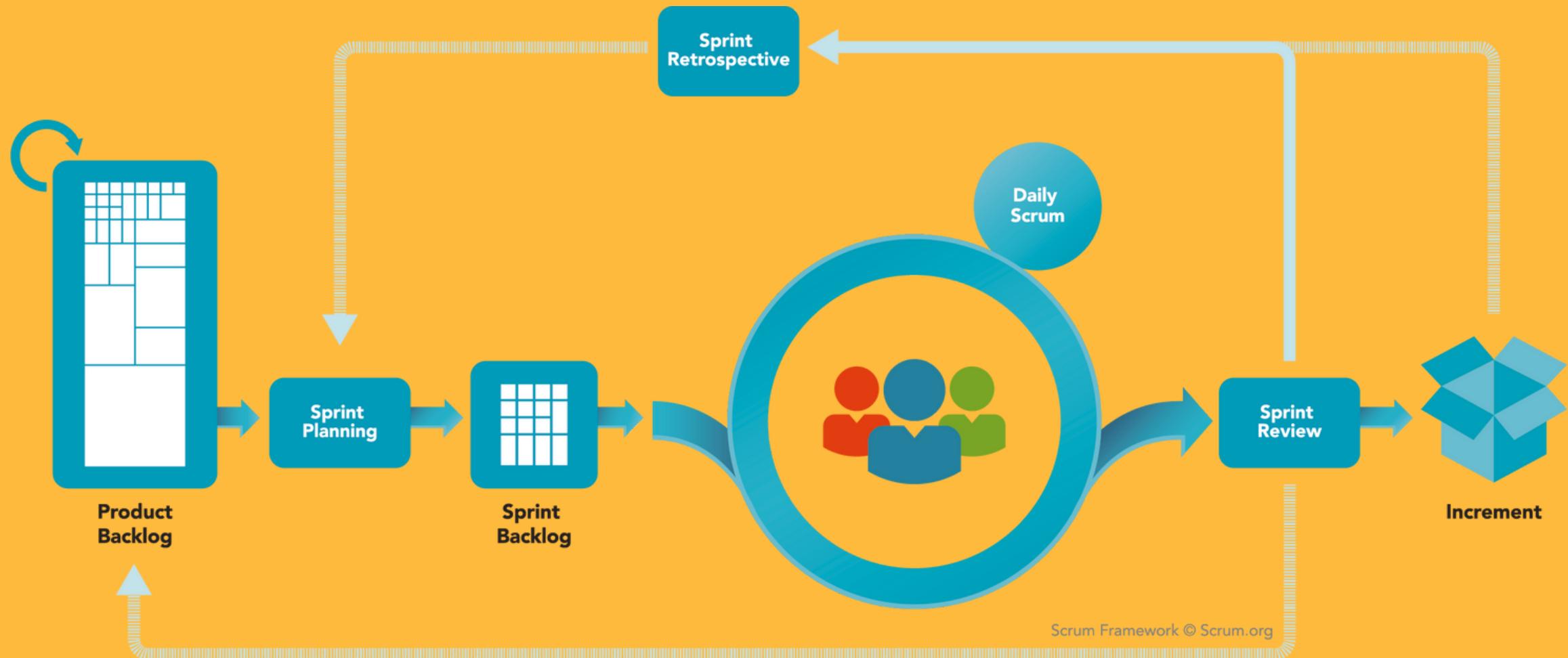
A short cycle of time where work is done. Usually 1,2,3 or 4 weeks.



Agile | Scrum

- **Product Backlog**

A list of tasks, features, functionalities, or changes that needs to be done to improve the product. Not all the items in the Product Backlog will be developed. It's a wish list from a Business owner



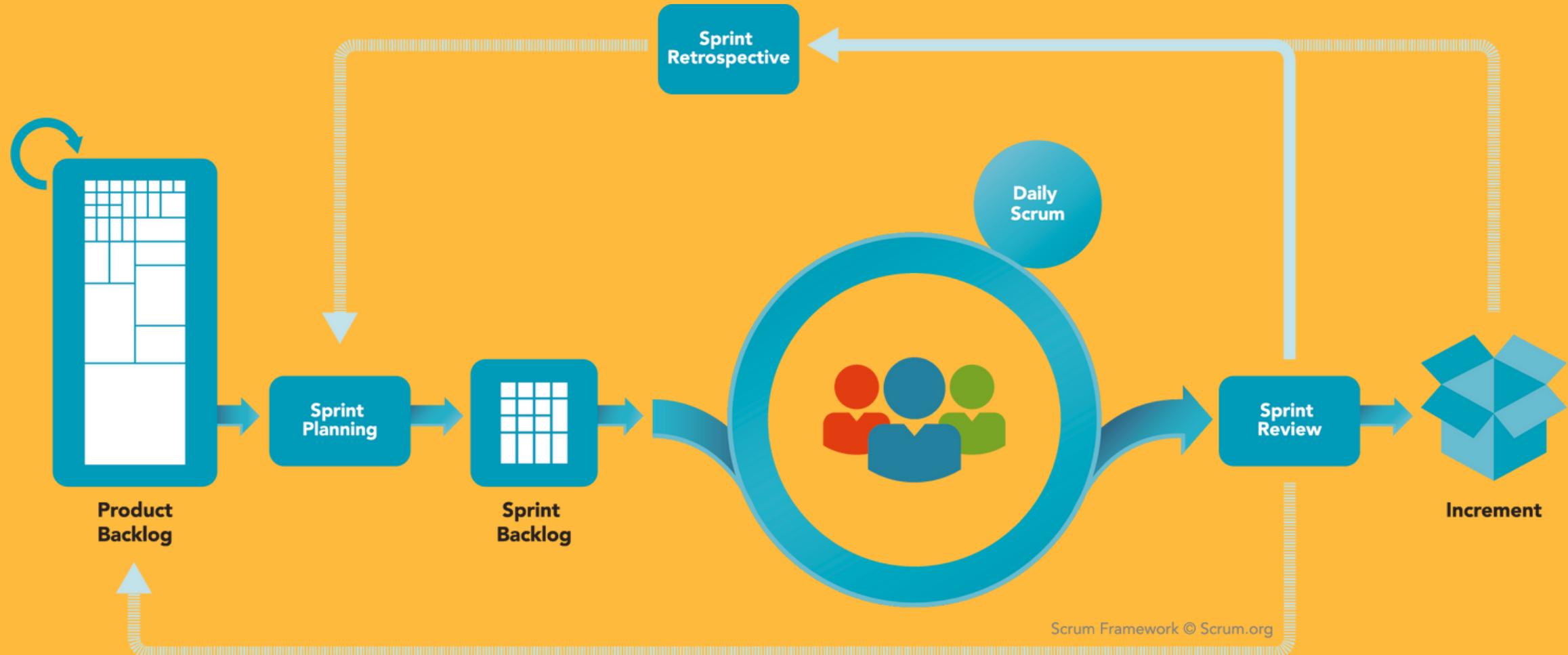
Agile | Scrum

- **Sprint Planning**

Prioritizing the items from the Product Backlog and adding to Sprint planning to finish in 1 sprint. Also other related work planning during that sprint

- **Sprint Backlog**

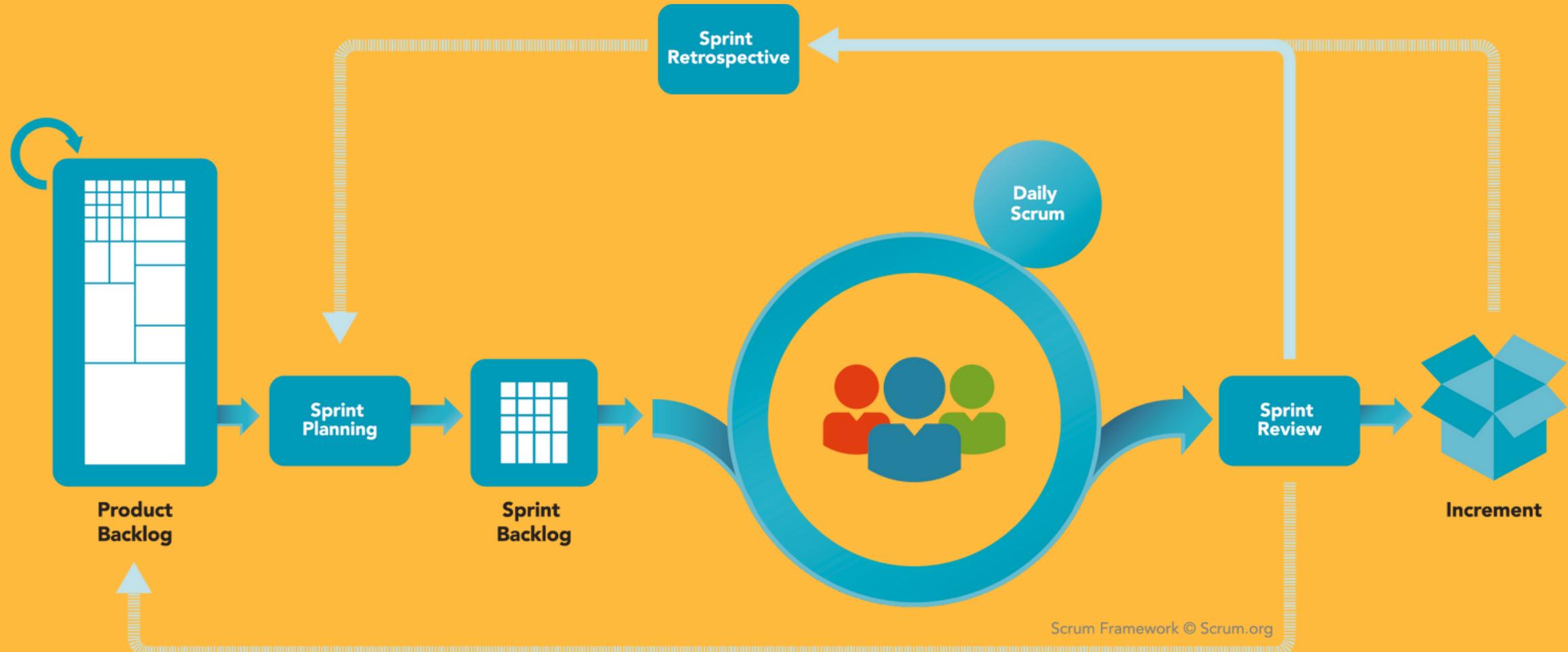
Only items that are assigned to the sprint



Agile | Scrum

- **Daily Scrum**

The daily quick 5 minutes meeting with the Scrum team led by the Scrum master to track the progress



- 3 questions asked

1. What did you do yesterday?
2. What are you gonna do today?
3. Are there any blockers?

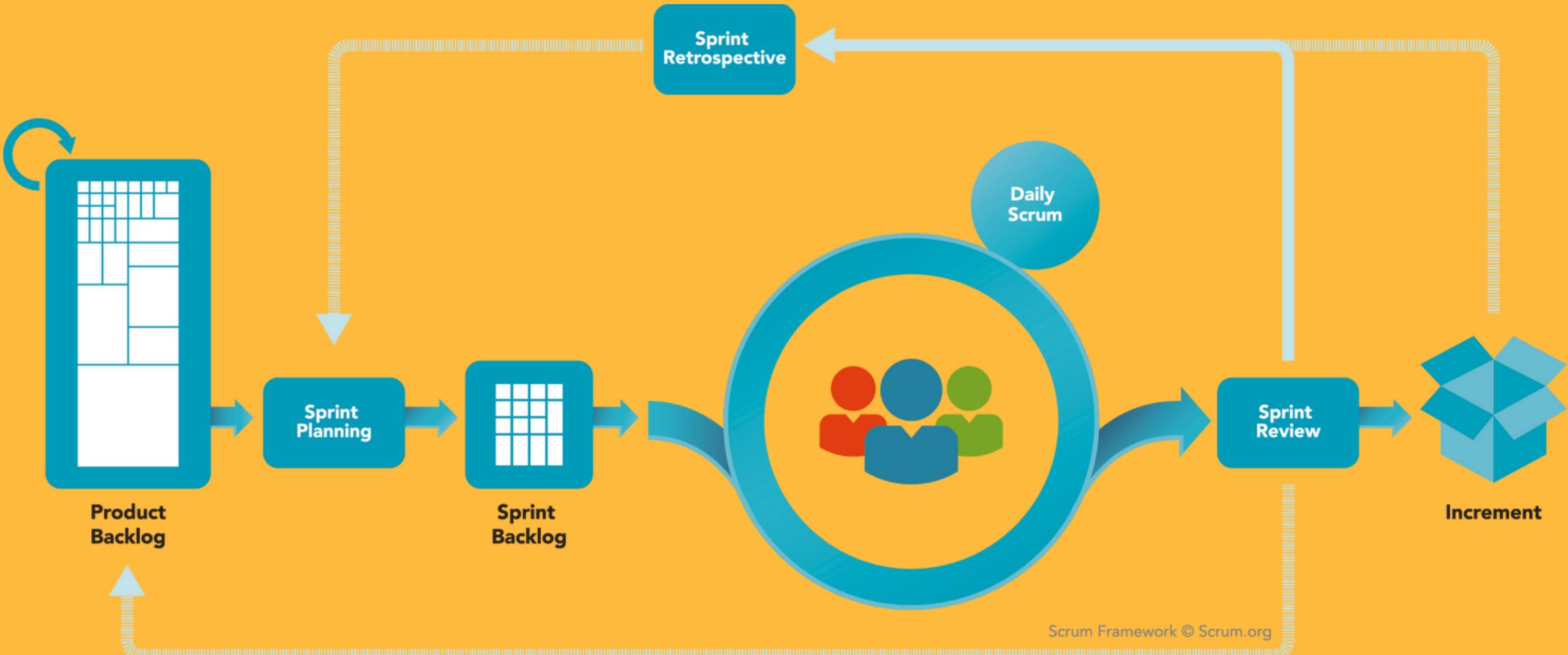
Agile | Scrum

- **Sprint Review**

Review of the work done during the sprint. Demo

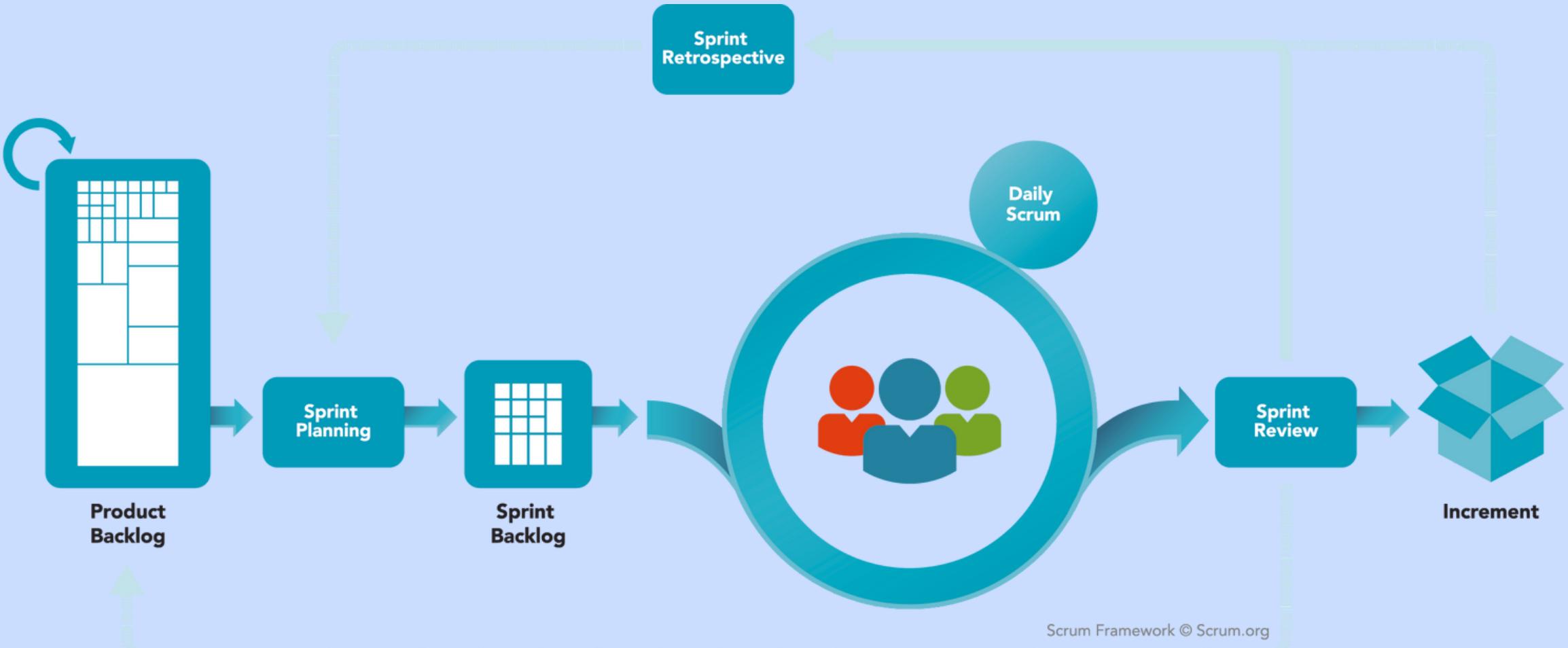
- **Sprint Retrospective**

The **scrum team** discussed what went good what went wrong and learn from mistakes to avoid in upcoming sprints



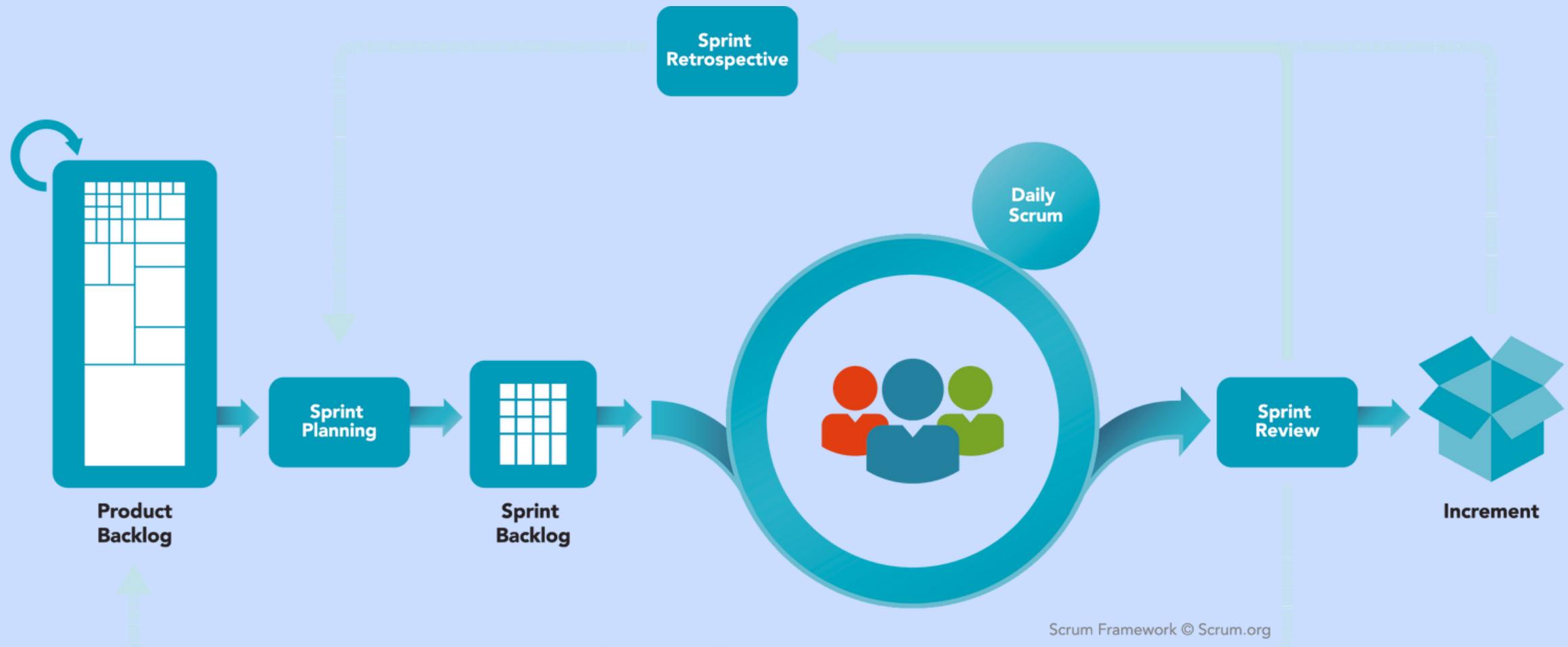
User story

- A user story is a term used in Scrum for software requirements or feature
- It describes the behavior of the end user. Hence, the user story from the user's perspective



User story

- As a user, I should be able to log in to the software, so that I can do all operations.
- As an admin, I should be able to create courses, so that I can add lessons and assignments, etc.
- As a student user, I should be able to submit an assignment response, so that I can complete the assignment



JIRA

Tool is used during development

Keywords

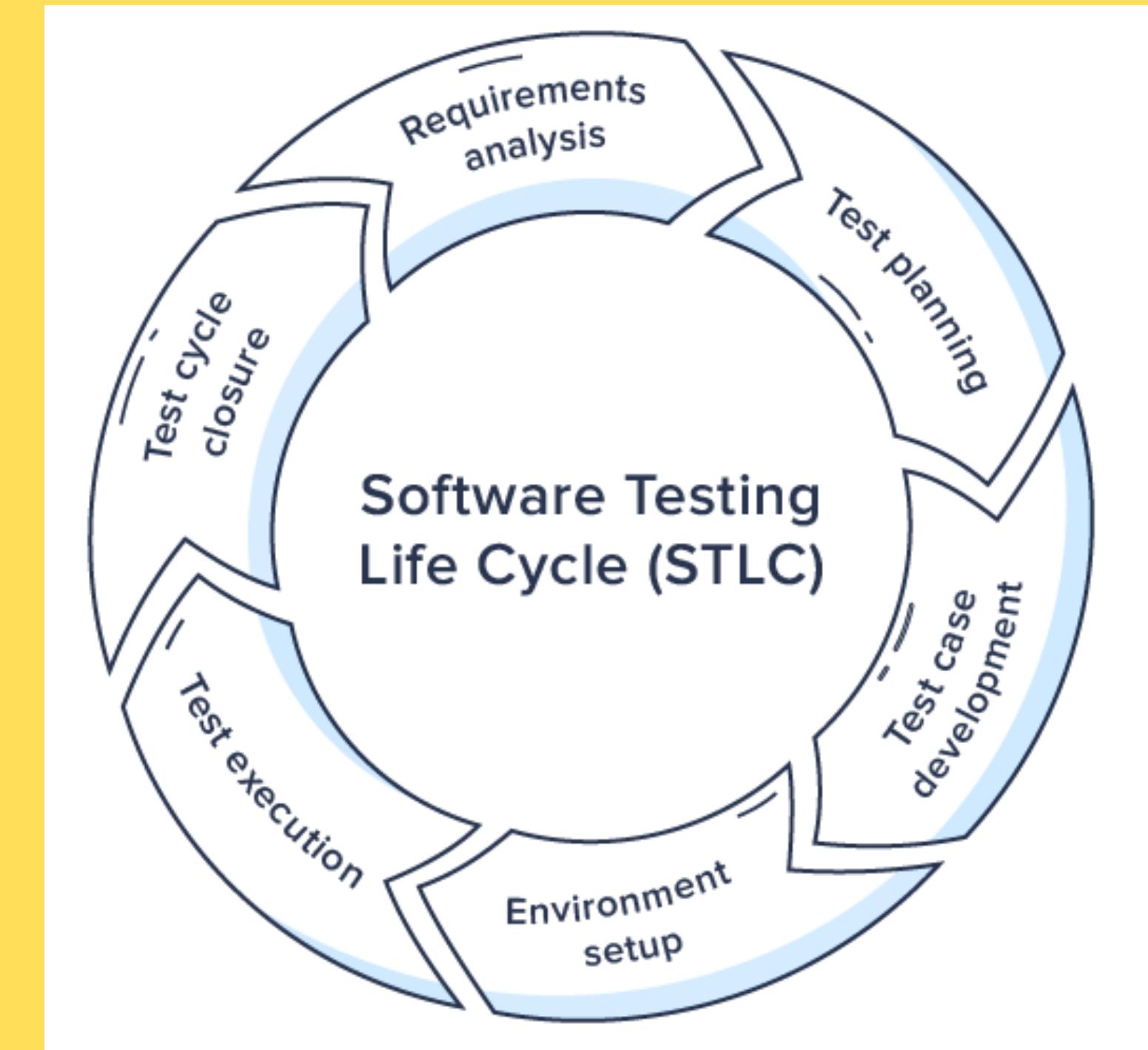
- Pointing system
- Estimation
- Ping
- Troubleshoot
- Root cause
- Cross-functional



STLC - Software Testing Life Cycle

The list of specific actions that need to be done to ensure that the software quality objectives are met!

- Requirement analysis
- Testing planning
- Test case development
- Test environment setup
- Test execution
- Test Cycle closure



STLC - Software Testing Life Cycle

Requirement analysis

- Learning how software should work
- Analyzing how certain functionality works
- Each item is identified whether automatable or not



STLC - Software Testing Life Cycle

Test Planning

- Who is going to work on which task?
- What are the necessary tools for testing
- Necessary environment
- What kind of testing should be done? Types of testing



STLC - Software Testing Life Cycle

Test Case development

- Test cases should be written for both automation and manual testing
- They should be done separately



STLC - Software Testing Life Cycle

Test Case

- Test case - a list of actions executed to verify a particular feature/functionality of the software.



Login

- Test case 1: check the result by entering the valid username and password
- Test case 2: Check the result by entering the invalid username and password
- Test case 3: Check the result by leaving field empty and trying to login

STLC - Software Testing Life Cycle

Login



- Test case 1: check the result by entering the valid username and password
 1. Go to site <https://cashwise.us>
 2. Enter valid username
 3. Enter valid password
 4. Click "Login" button

Expected result: successfully login to Cashwise platform

Actual result: -----

STLC - Software Testing Life Cycle



Write test cases for below

- Test case 2: Check the result by entering the invalid username and password
- Test case 3: Check the result by leaving the field empty and trying to login

STLC - Software Testing Life Cycle

Test Environment setup

- Making sure the software is ready for testing.
- If any issue, development team or DevOps of anyone who is the blocker should resolve



STLC - Software Testing Life Cycle

Test Execution(running)

- Both Automation and manual tests are performed
- If there are any issues then Bug is raised and assigned to developers to fix it



STLC - Software Testing Life Cycle

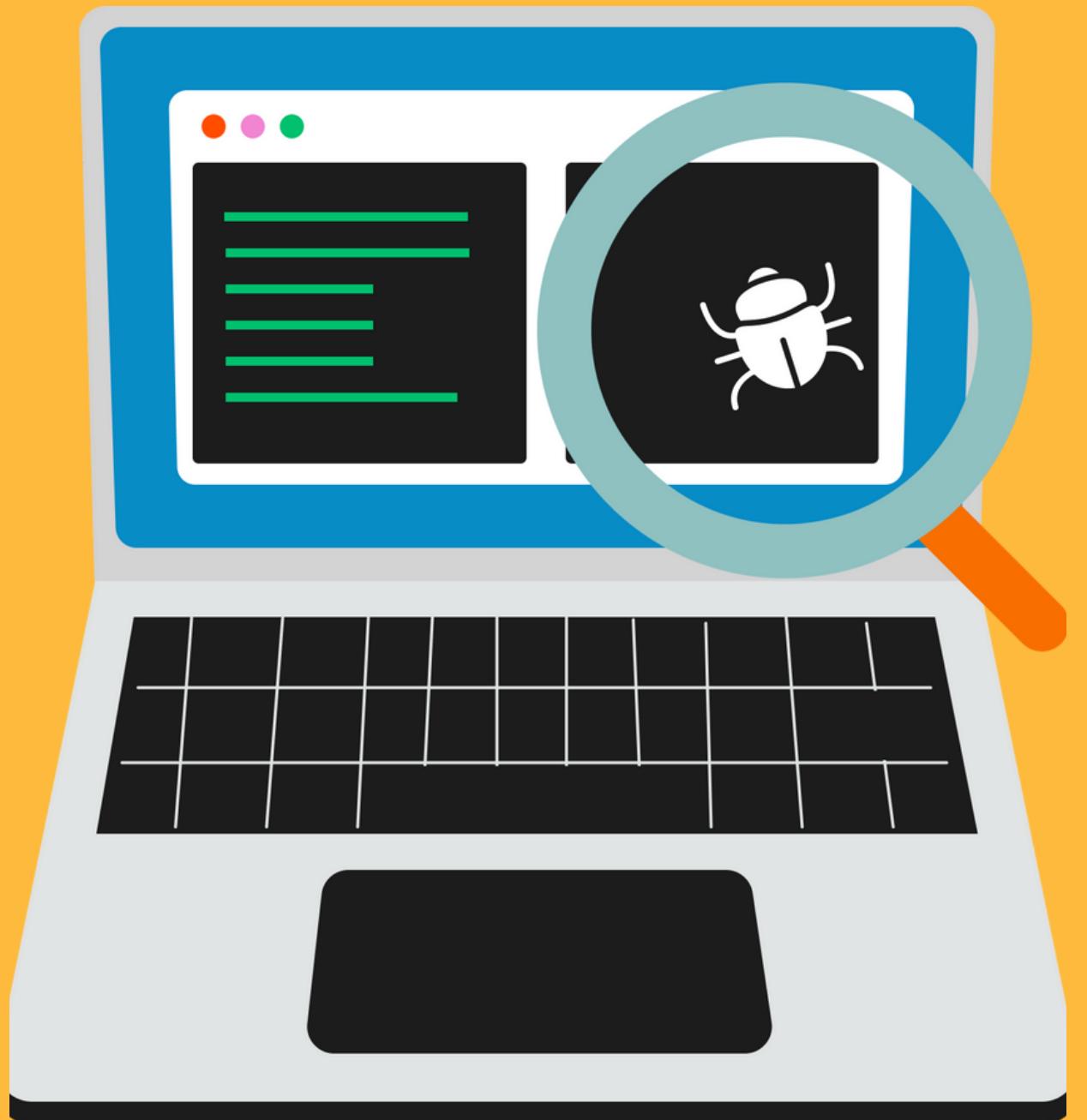
Test Cycle Closure

- Report
- how long it took?
- Number of bugs
- Fixed bugs
- Not fixed bugs



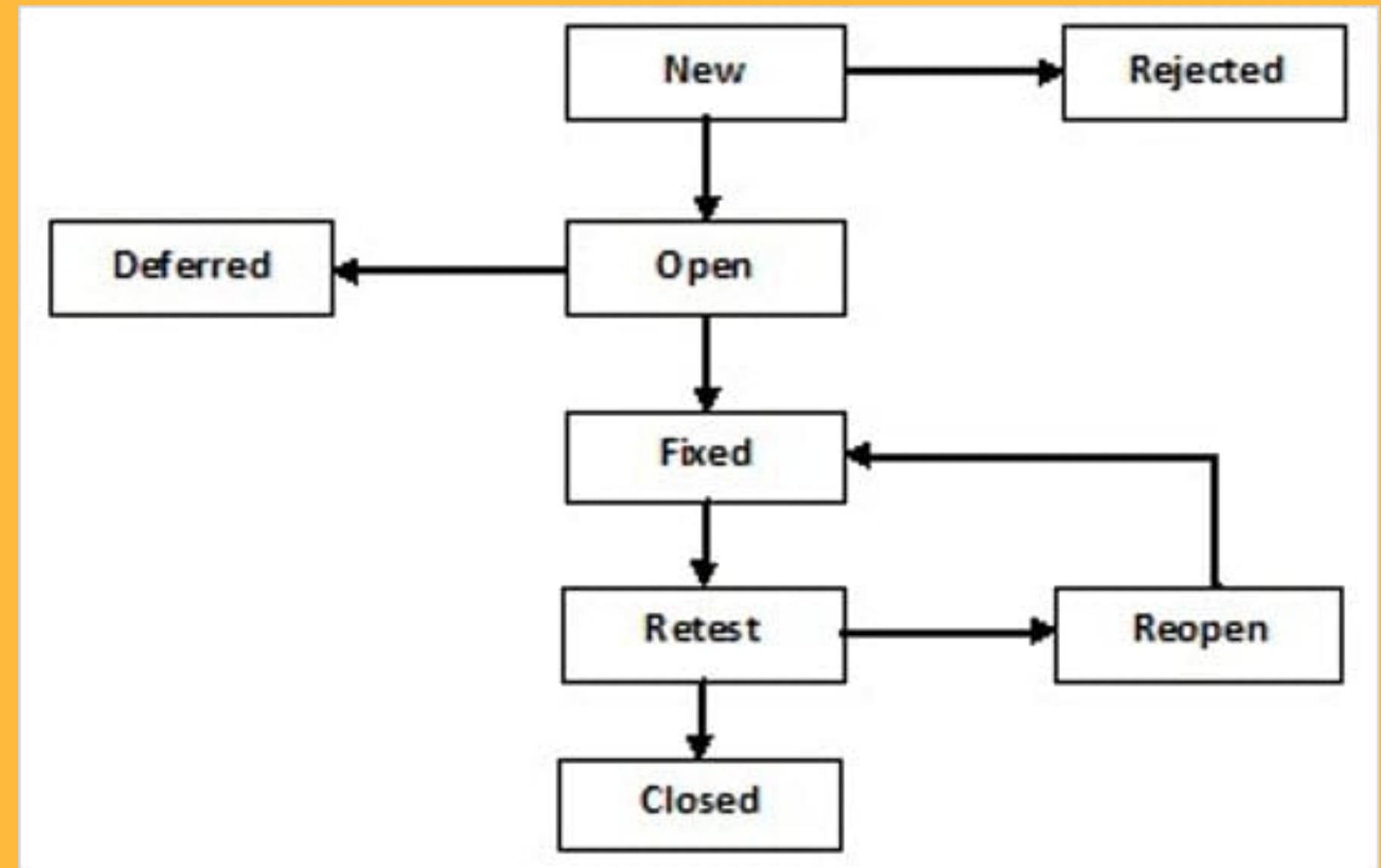
Bug/Defect life Cycle

The bug life cycle is the process of one bug getting created and until fixing to close the bug ticket



Bug/Defect life Cycle

- When both manual or automation testers find a bug the first step is reproducing it
- After assuring the bug gets created on Software development management tool



BUG/DEFECT/ISSUE

In software development, a bug refers to an error, flaw, or defect in a computer program that causes it to behave in an unintended or incorrect way



Bug/Defect life Cycle

- Reporting a bug

Summary: short summary of the bug ticket that is understanding what the bug is about

Description: more detailed information about the bug

Severity: how critical is this bug? impact? Critical, Major, Minor, Low

Priority: when it needs to be fixed? High, Medium, Low

Steps to reproduce: Give step-by-step instructions to reproduce

Screenshots

Bug/Defect life Cycle

- Reporting a bug - SRS541

Summary: Instructor course creation not working

Description: In the instructor role, course creation gives error message

Severity: Critical

Priority: Medium

Steps to reproduce:

1. Login into the Peaksoft LMS application
2. Navigate to courses
3. Click on create a new course

Expected result: on clicking create a new course, the application should allow the instructor to create a new course. (See screenshots)

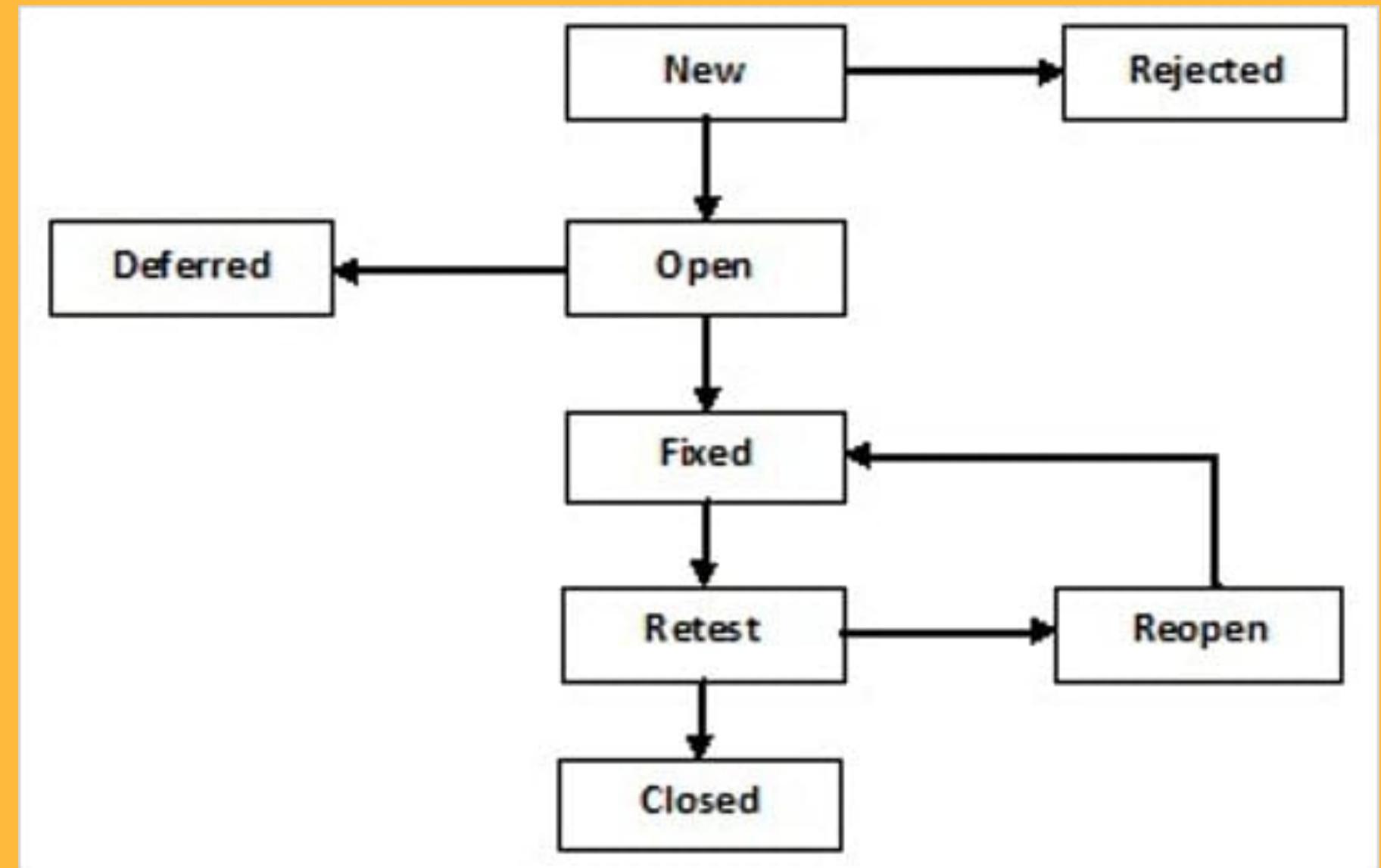
Bug/Defect life Cycle

Bug SRS541

New: when a bug newly gets created by the Tester

tip: First reproduce the issue multiple times.

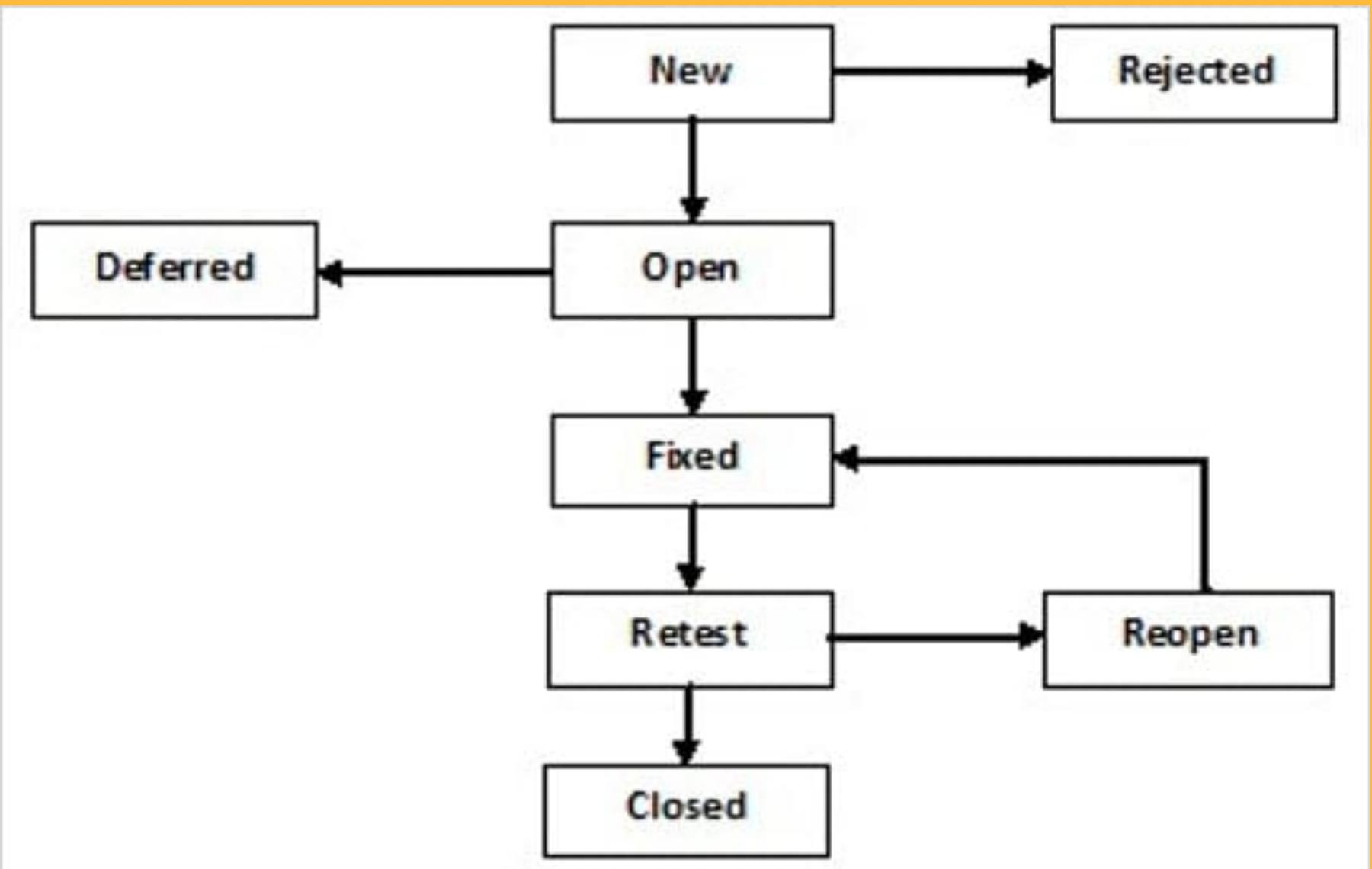
Then discuss with the developer.



Bug/Defect life Cycle

Bug SRS541

Open: when the developer accepts it and starts working
tip: have an idea when it will be done. Ask or pay attention during the meeting

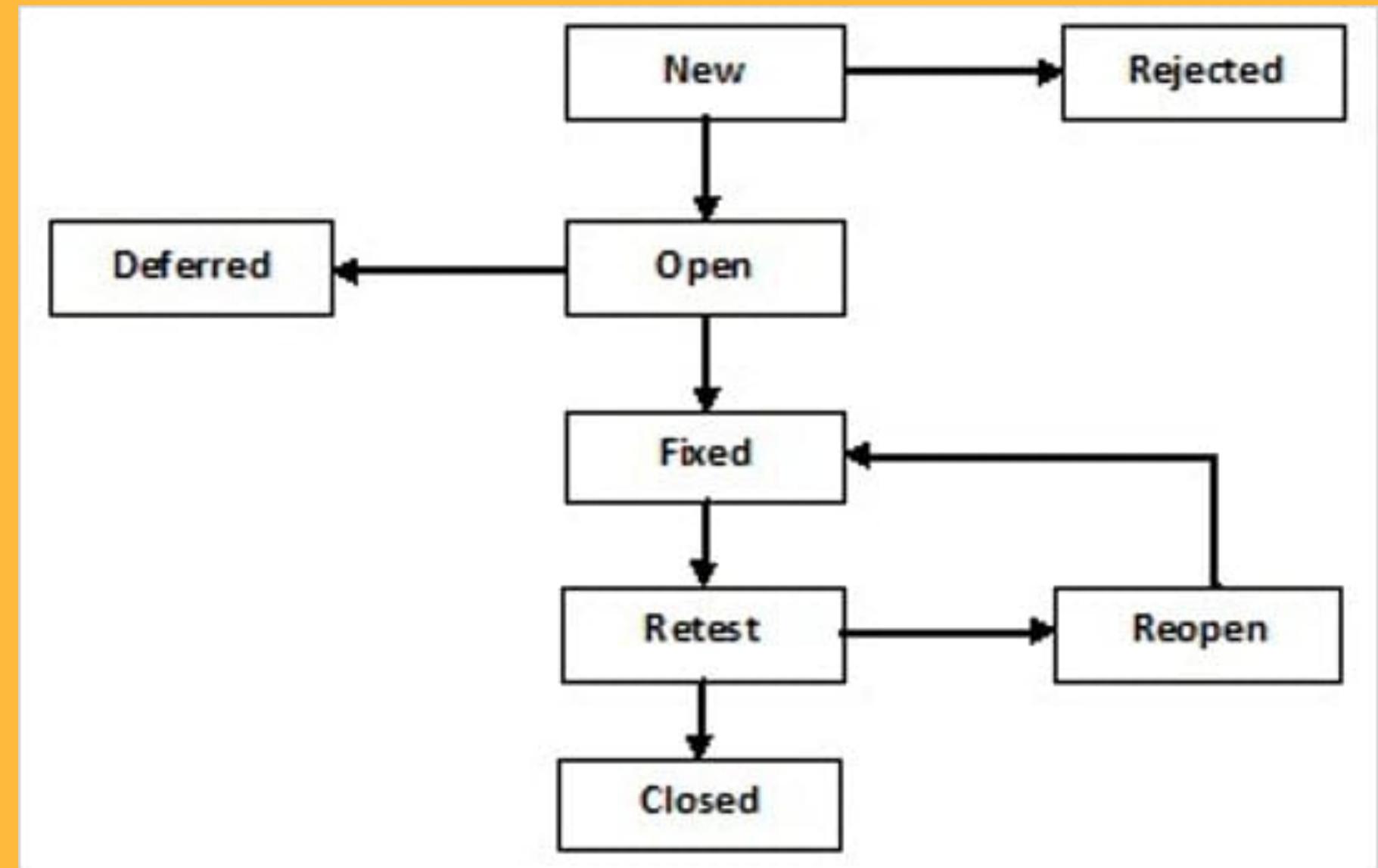


Bug/Defect life Cycle

Bug SRS541

Rejected: when the developer
doesn't agree with you regarding
this bug

tip: check with Business Analyst

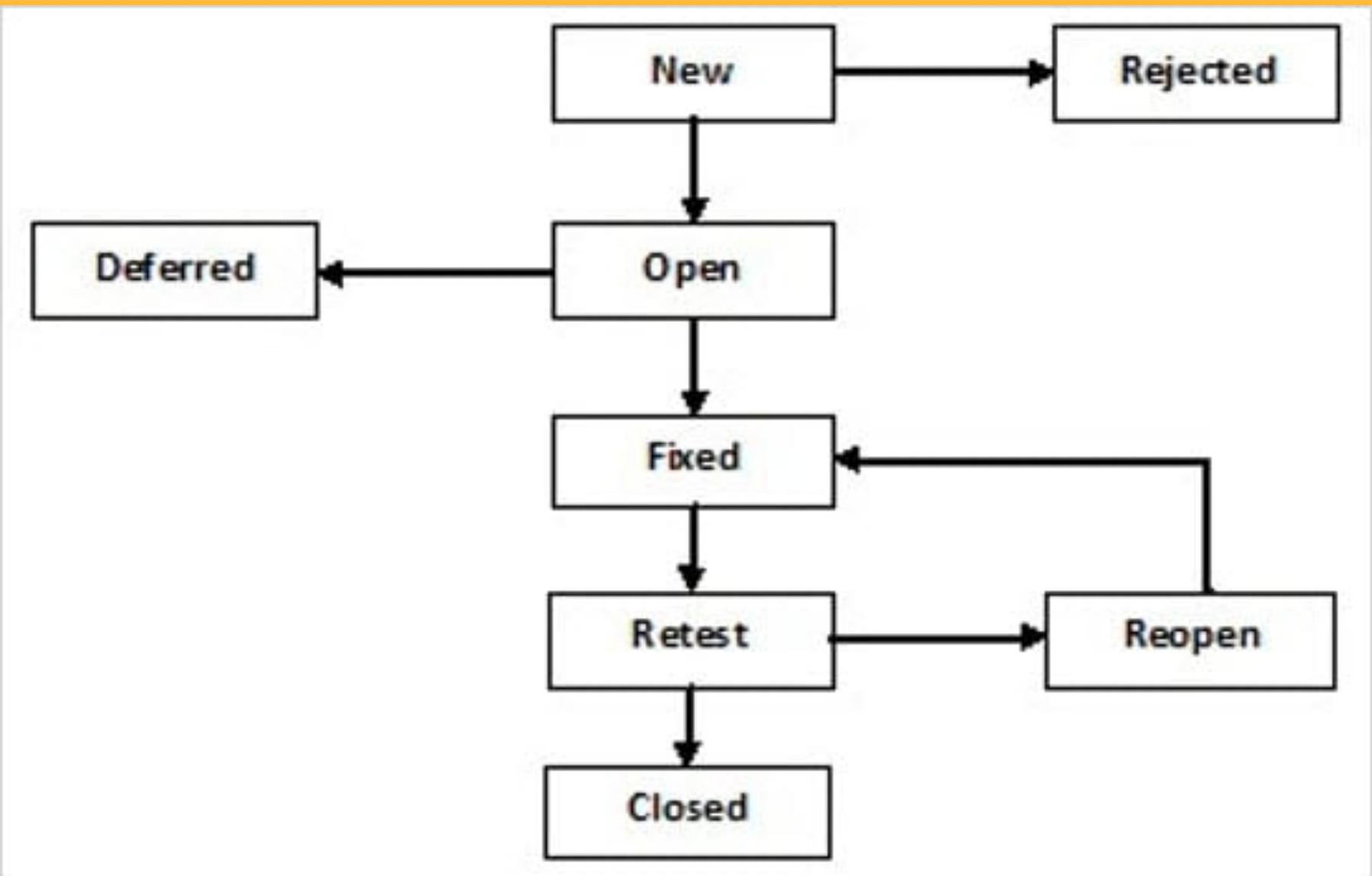


Bug/Defect life Cycle

Bug SRS541

Deferred: when the bug is not so important and blocker. The developer pushes to the next sprint or release

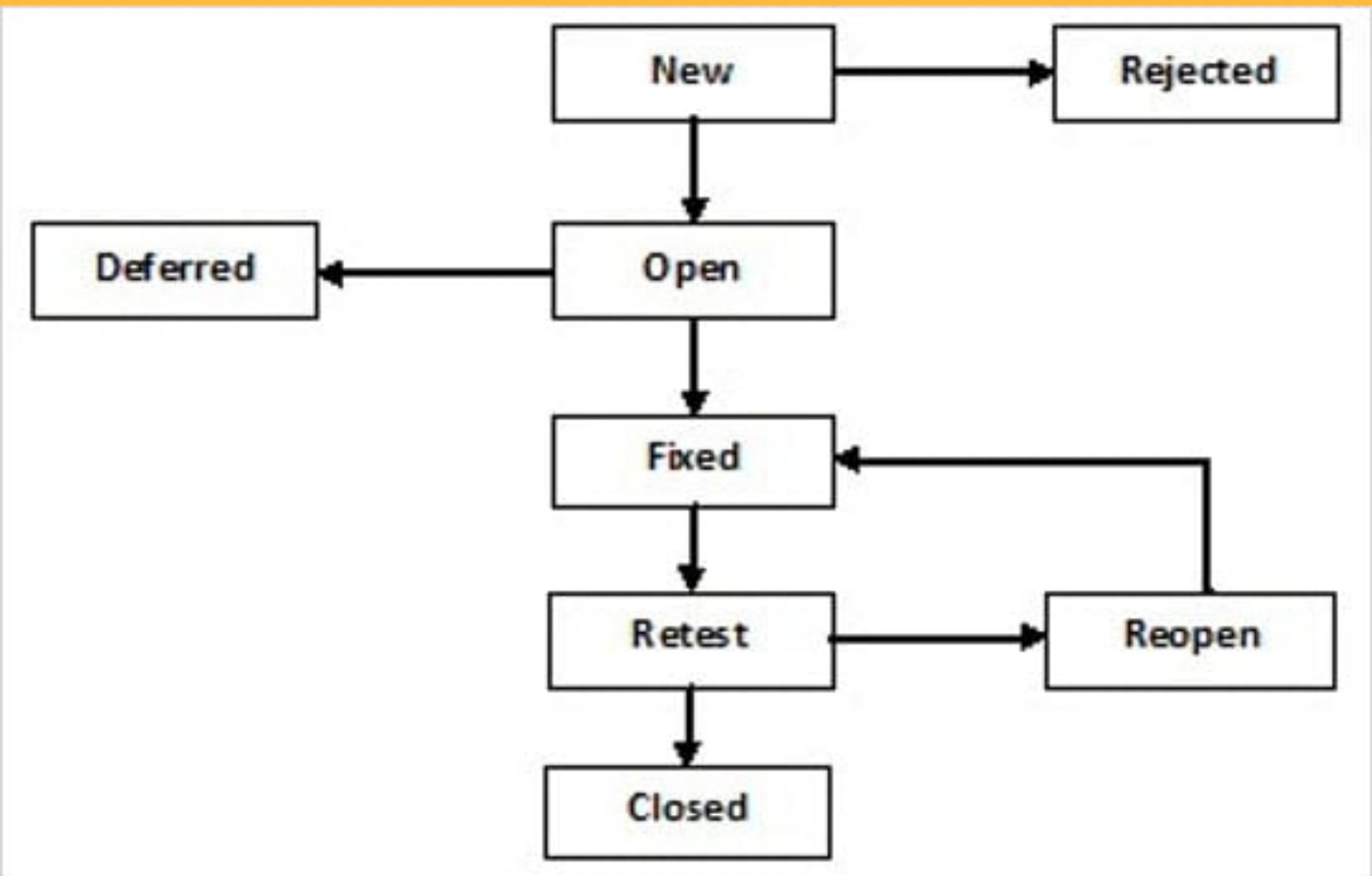
tip: let test lead or manager know about it



Bug/Defect life Cycle

Bug SRS541

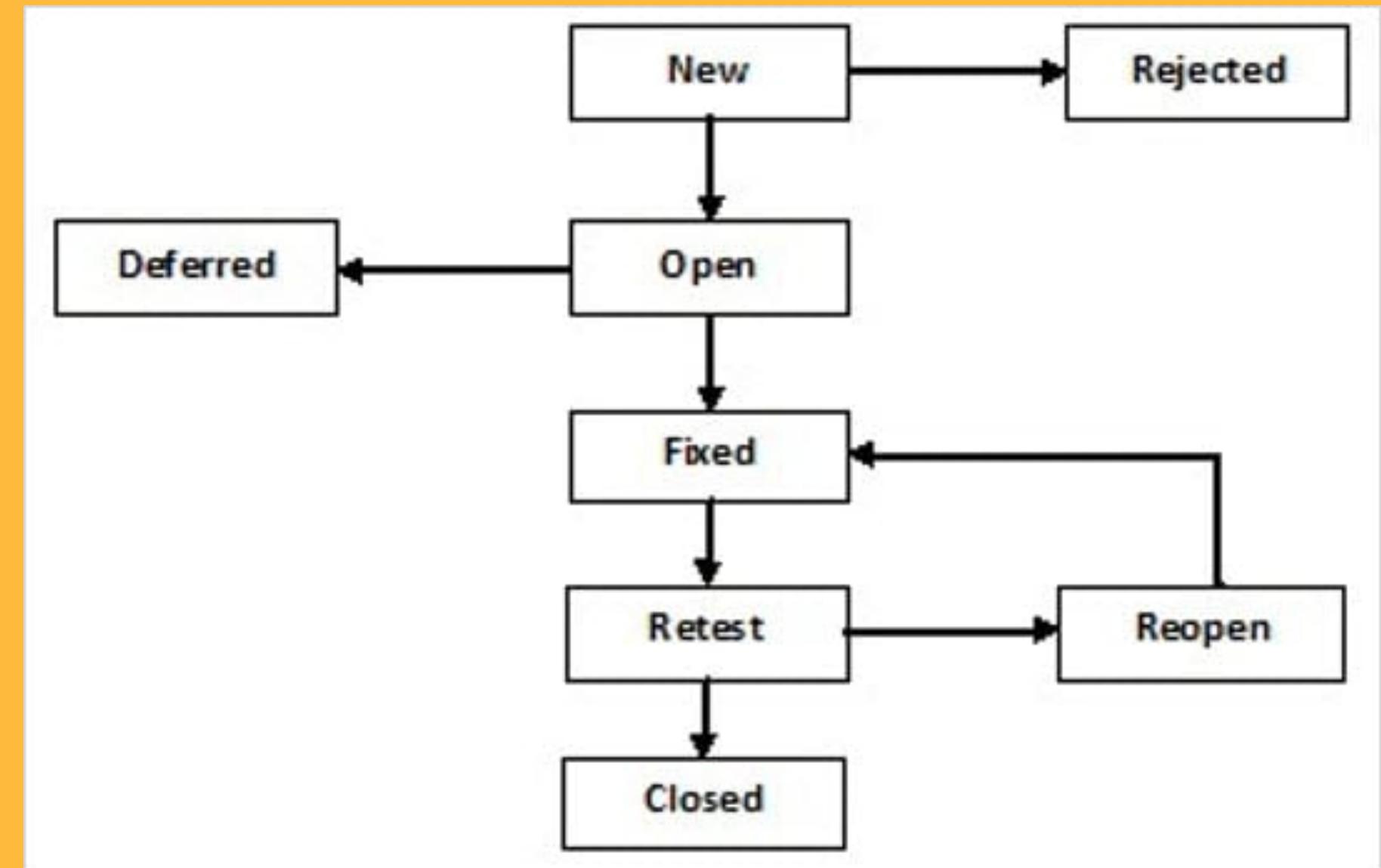
Fixed: when the bug is fixed by the developer, but not available for retesting yet.



Bug/Defect life Cycle

Bug SRS541

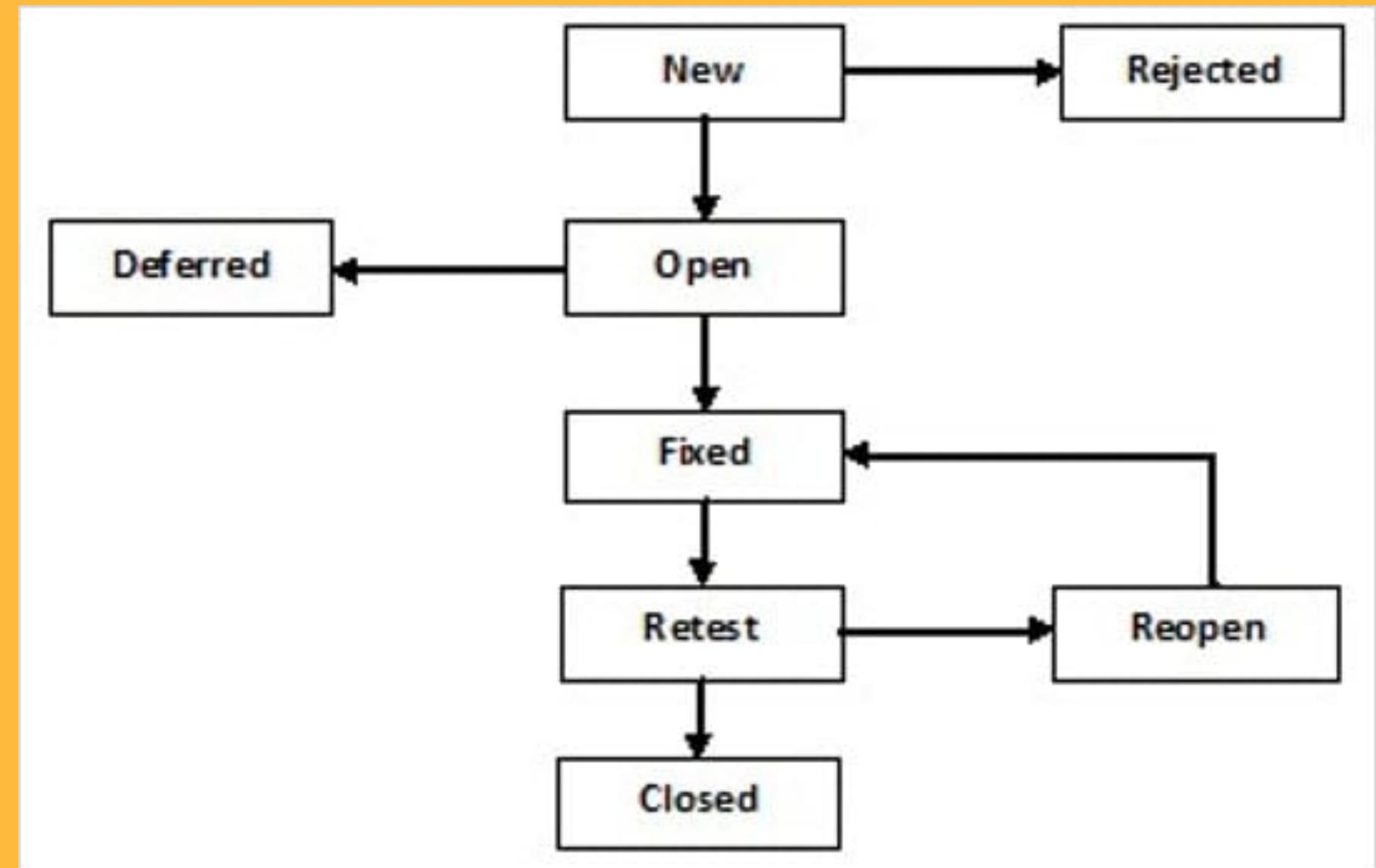
Retest/Ready for retest: the author of the bug must retest the fixed bug



Bug/Defect life Cycle

Bug SRS541

Reopen: when tester doesn't agree with the result and reopens and assigns back to the developer to fix it

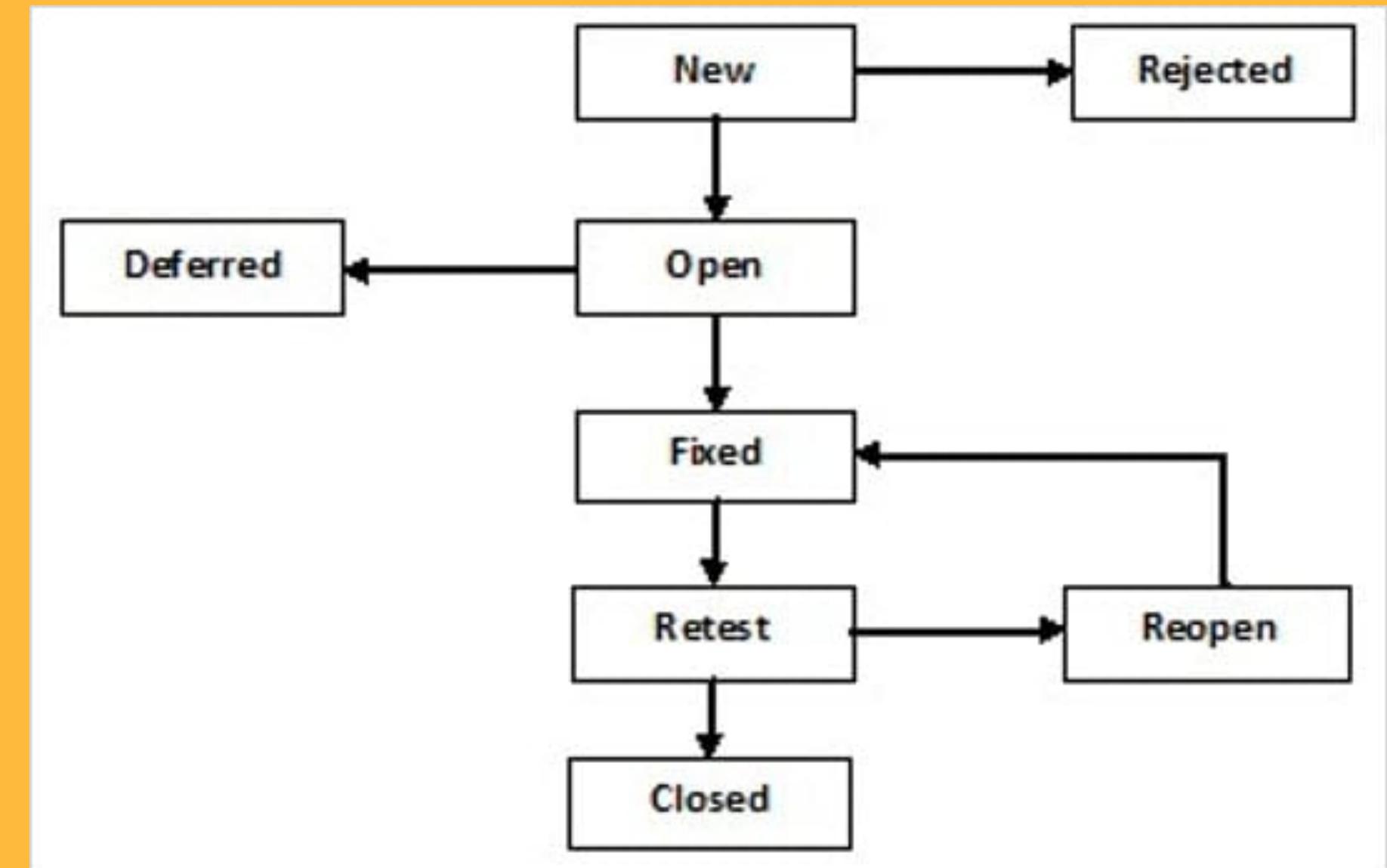


Bug/Defect life Cycle

Bug SRS541

Closed: the bug is successfully fixed and retested successfully.

At this point tester can close the bug



Testing types

1. Unit testing
2. Smoke testing
3. Integration testing
4. Regression testing
5. Functional testing
6. Performance testing



Testing types

1. Unit testing

Unit testing is normally done by the developers at the development stage of the software.

It is the lowest and earliest testing at the code level.

There are ready frameworks for developers to use to do unit testing.

Ex: Junit, TestNG



Testing types

2. Smoke testing

Smoke testing is testing only the main and basic functionalities of the Software just to make sure that the software is up and running.

This testing is done by the Tester(usually by automation tester) on a daily basis.

Also, this testing is scheduled early in the morning before everyone comes to work



Testing types

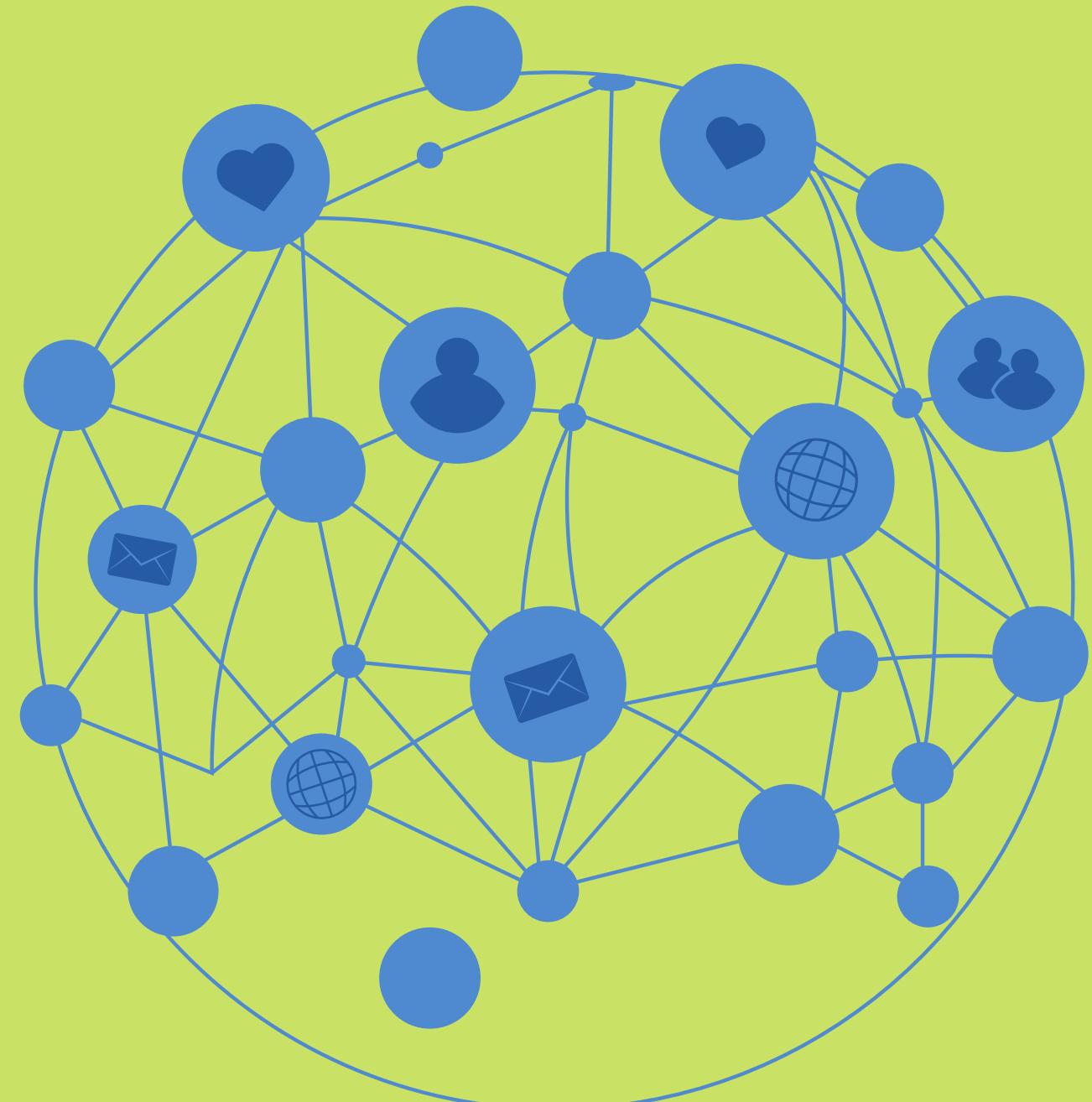
3. Integration testing

Integration testing is done by the QA team.

This type of testing is focused on checking the two or more modules working together.

One software/application is dependent on other services or departments. In this case,

QA should do integration testing to make sure integrated services and modules are working together fine



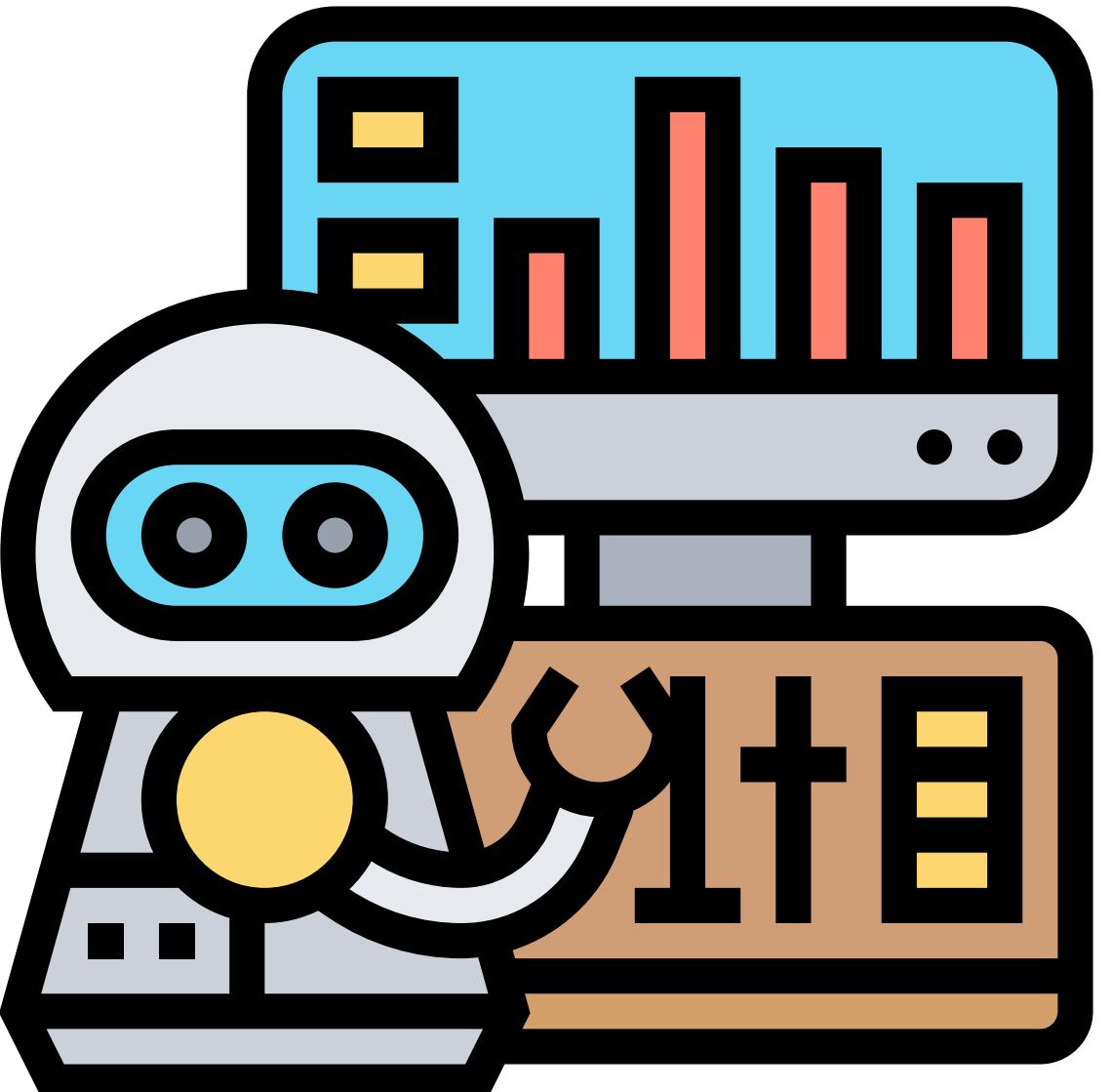
Testing types

4. Regression testing

Regression testing makes sure that newly created or added features/functionalities are not breaking the existing software.

Every time developers make new changes to the software's code, there is a possibility that it might break the other untouched part of the software since their code base is the same.

QA Automation code is very useful in this case



Testing types

5. Functional testing

Functional testing focuses on the Software requirement and tests the functionalities of the application.

When we do negative, positive, boundary testing. These are the examples of functional testing



Testing types

6. Performance testing

This type of testing is all about checking the algorithm, system, and servers. Because it focuses on checking the speed of the application, the way software handles when too many users use it at the same time, scalability, response time, reliability etc.

This test is normally done by the Performance team. Sometimes by devs, by QAs too



Testing types

Black box and white box testing

These are the testing grouping at a high level.

Black box testing means the tester doesn't know about the internal structure or the code of the functionality. We focus on the behavior of the software.

White box testing is focusing on the code.

Otherwise, unit testing. Testing the actual code, system, and logic of the internal structure

