

Manual testing

Contents:

SDLC (Software development life cycle)

- i. Waterfall Model
- ii. Spiral model
- iii. V model
- iv. Prototype model
- v . Agile model

Software testing

i. White box testing

- Path testing
- Conditional testing
- Loop testing
- WBT from memory point of view
- WBT from performance point of view

ii. Grey box testing

iii. Black box testing

- Functionality testing
- Integration testing
- System testing
- Acceptance testing
- Smoke testing
- Ad-hoc testing
- Regression testing
- Performance testing
- Compatibility testing
- Globalization testing
- Exploratory testing

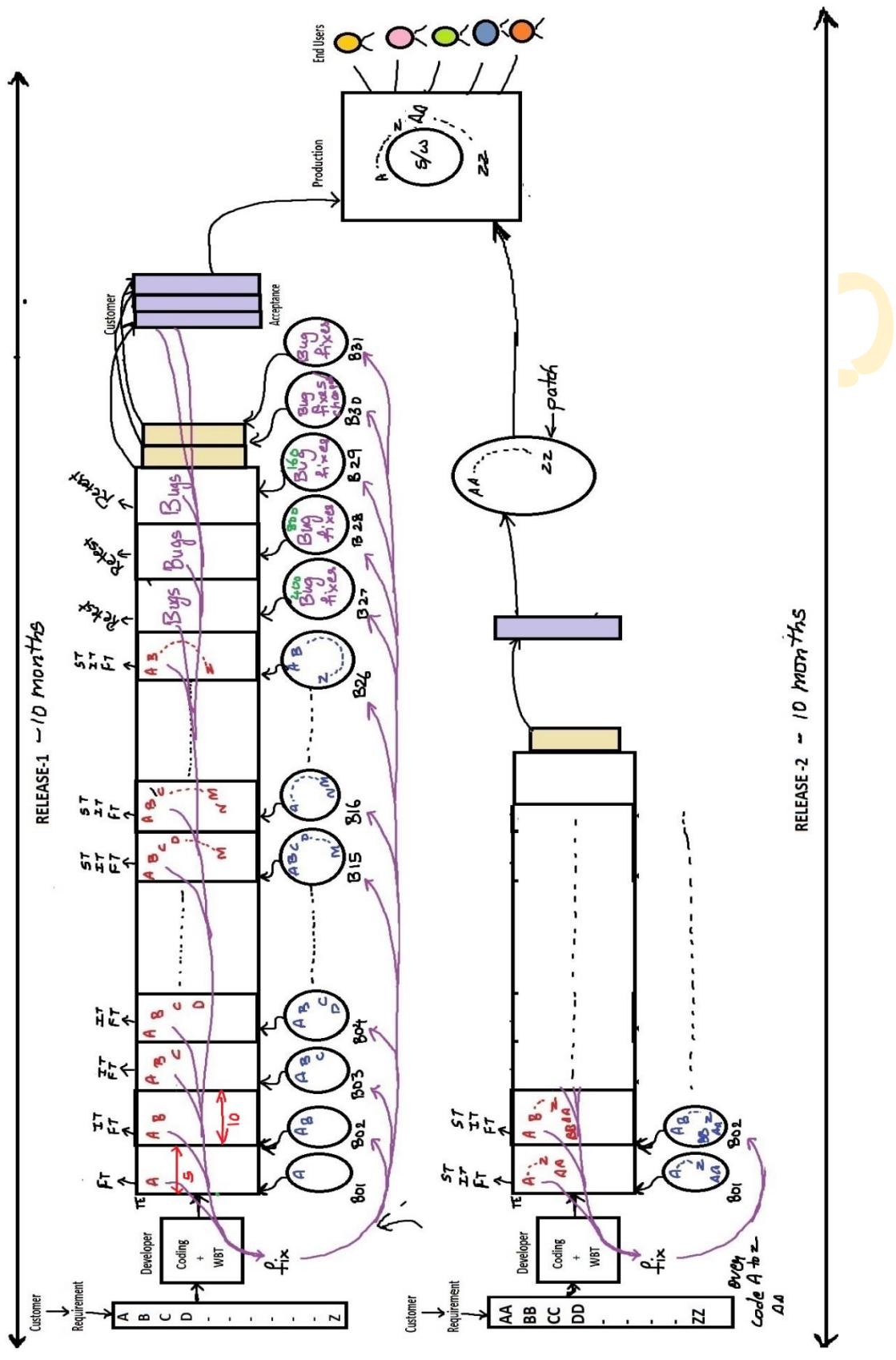
Test Plan

Defect tracking/ Defect life cycle

STLC (Software testing life cycle)

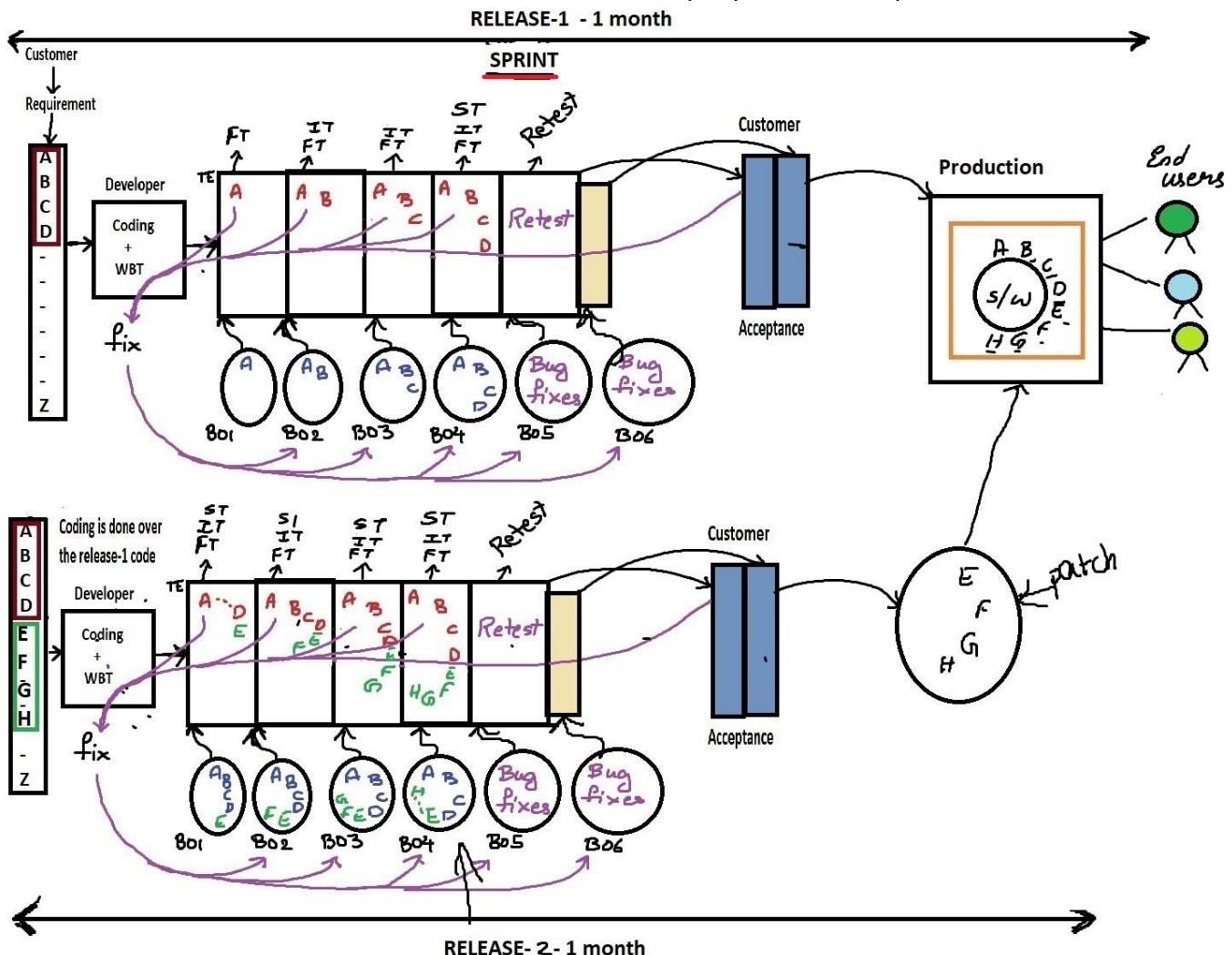
AGILE

Agile is a model where we develop software in an incremental and iterative process. They have come up with this model in order to overcome the drawbacks of the traditional model. Here we build large products in shorter cycles called as sprints.



Drawbacks of Traditional Model

1. The project might not meet the market standards due to longer tenure
2. The investment return takes more time where the customer takes long to start his business
3. No communication between the customer and the company once the requirement collection is done



Flavours of Agile Model

1. Scrum Process
2. Lean and Kanban
3. Xtreme Programming
4. Crystal Clear
5. Feature Driven Development (FDD) or Test Driven Development (TDD)
6. Dynamic System Development (DSD)
7. Adaptive System Development (ASD)

Scrum Process

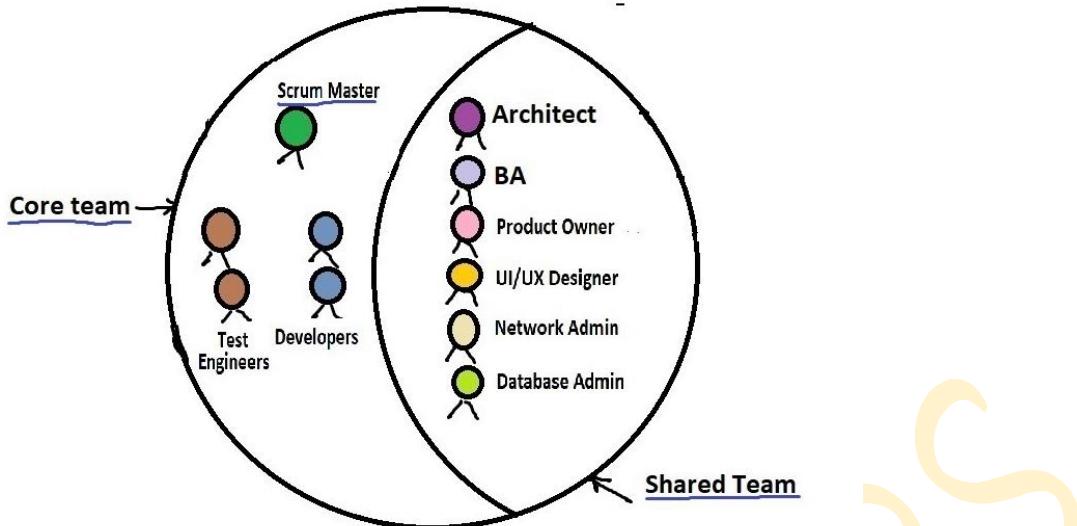
It is a process used to build an application in agile model

Maximum companies adapt to scrum process as it is a convenient and simple process.

Scrum Team

1. It is a group of engineers working towards completing the committed features and stories
2. Generally scrum team will have 5 to 12 members
3. It includes two teams, shared team and core team

4. Core team members include Scrum master, development engineer and test engineer
5. Shared team members include architect, product owner, database admin, network admin, UI/UX designer and BA
6. Scrum master leads this entire team and he facilitates everyone to complete their task



Product Owner:

1. He clarifies the questions from the development and testing team if any
2. He sets up the acceptance criteria for every story

What is acceptance criteria?

It is a criteria which should be present in order to move the product to the production or to the customer.

It is generally set by the product owner

This is done in the beginning of the sprint that is in the sprint planning meeting

EX: Create account -> Should go to pending approval page

Approve account -> User should be able to login. He should be listed in the RBI list. He should be listed in the income tax list

Development Engineer

1. He should derive the task for building every story
2. He should prioritize which story to build first
3. He should prioritize the tasks
4. He should derive the story points
5. Builds the story and performs white box testing

Test Engineer

He will derive the task to be completed to each feature or story

Ex: Create account:

1. Write test case
2. Review test case
3. Prepare traceability matrix
4. Execute the test cases
5. Defect tracking

Scrum Master:

1. The complete meetings are driven by the scrum master
 2. His prime role is to facilitate the all the members and co-ordinate between the stake holders
- EX:

- a) Getting architect support to the development engineers
- b) Getting product owner support to the TE to understand the stories or requirement

Product Backlog:

It is a prioritized list of stories or requirement that must be developed in the complete project

1. Generally product owner, customer, business analyst, architect, scrum master will be involved while building it

2. Generally stories in product backlog need not be in detail

Sprint backlog

1. It is the list of stories and the associated task committed by the scrum team that must be delivered with one sprint

2. It is actually committed by the scrum team that is development engineers and test engineers

Sprint Planning Meeting

1. Here entire scrum team sits together and pull the stories from the product backlog
2. Scrum master assign each stories to the development and testing team
3. Now each engineer derives the task to be completed to build the stories
4. Each engineer will prioritize the tasks to be completed
5. Each engineer estimate the time taken to complete their task, that is, they will be deriving the story points
6. Generally Sprint planning meeting should be completed within 8 hours (1 week sprint -> 2 hours)

Stand Up meeting/ Scrum meeting/ Daily meeting/ Status call/Role call meeting

1. Here entire scrum team meets. This meeting is completely done by the scrum master. Here every engineer should explain the following

- a) What they have done yesterday?
- b) What were the impediments/hurdles faced yesterday?
- c) What are the activities that are planned to do today?
- d) What are the impediments that are expected in order to complete today's job?

2. Scrum master tries to solve certain impediments right there in the meeting. If it is taking too much time, then scrum master notes it down in impediments backlog and solves it later

3. Generally this meeting should be completed within 10 to 15 mins

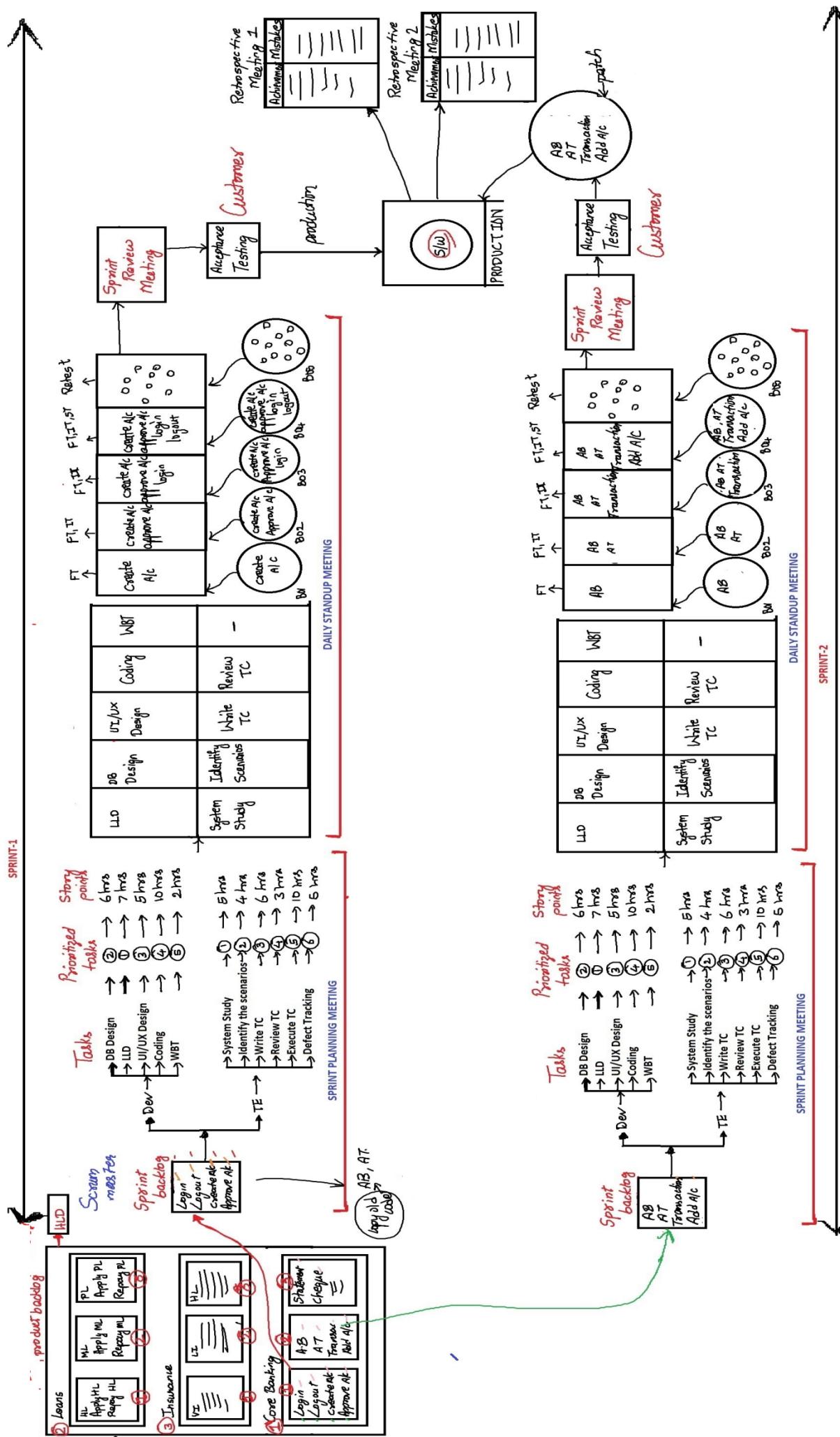
4. This meeting should be conducted in the beginning of the day

5. Here everyone should stand up in the meeting, so that people can talk to the point.

Sprint Review Meeting

1. Sprint Review meeting should be done at the end of every sprint where they give a demo on how the product is working and there are any missing features or stories, they will be corrected and discussed how to plan the next sprint

2. Generally product owner, developers, test engineers and the scrum master will be part of this meeting

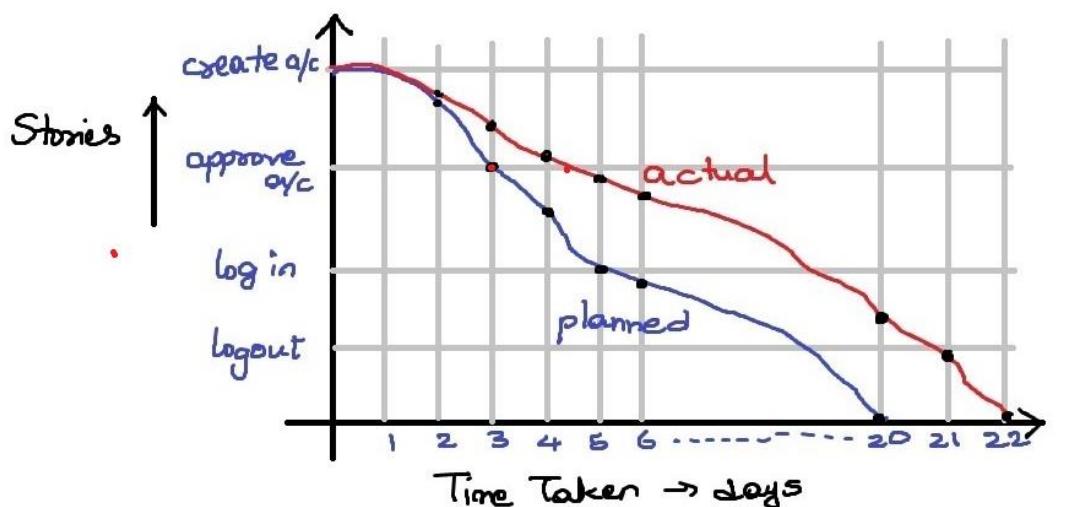


Retrospect meeting

1. Here the entire scrum team meets , sometimes customer may also join the meeting and discuss about all the achievements (good processes to follow and mistakes (wrong activities to correct)
2. These achievements and the mistakes will be documented as Retrospect document
3. When the next sprint planning meeting starts, we refer to this document and plan in such a way that old mistakes should not be repeated and good processes should be adopted

Burn Down Chart

It is a pictorial representation of the work pending/ left versus the time taken
Usually the scrum master updates the chart



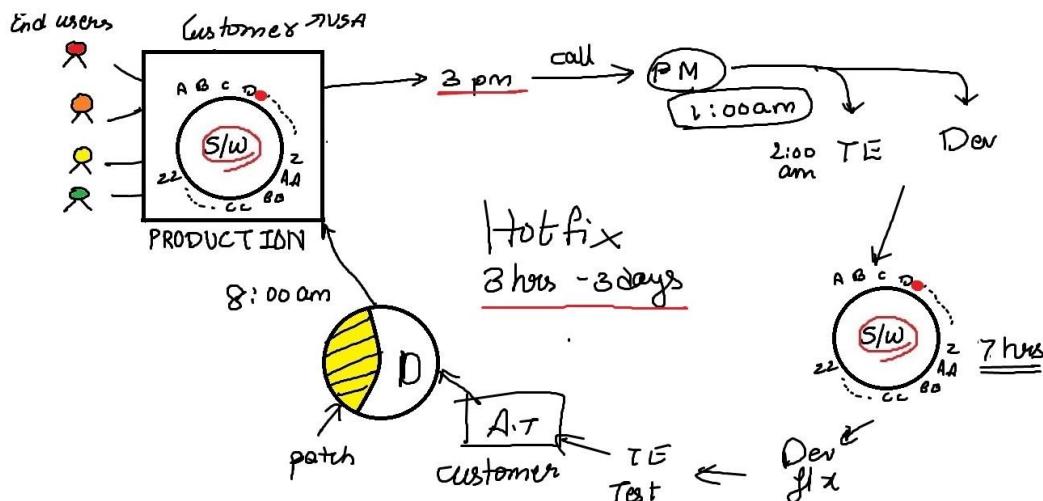
White board/ Story board

It is the board which contains the list of pending tasks, tasks in progress and completed tasks

Employees	Pending Tasks	Tasks in Progress	Completed Tasks
Employee-1	□ □ □ □ □	□ □	□
Employee-2	□ □	□ □ □ □	□ □ □ □
Employee-3	□ □ □ □ □ □ □	□ □	
Employee-100			

Hotfix

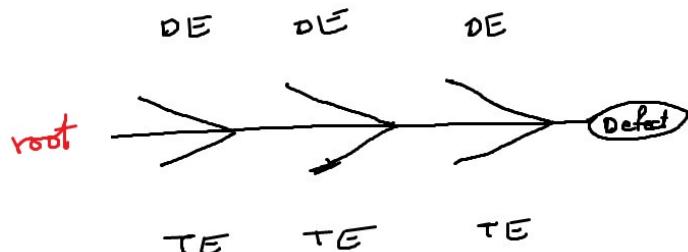
When ever a blocker/critical bug is found in the production, immediately the developer will fix the defect and TE will test it, customer will do a quick acceptance testing and a patch will be moved to the production. This is called as hotfix



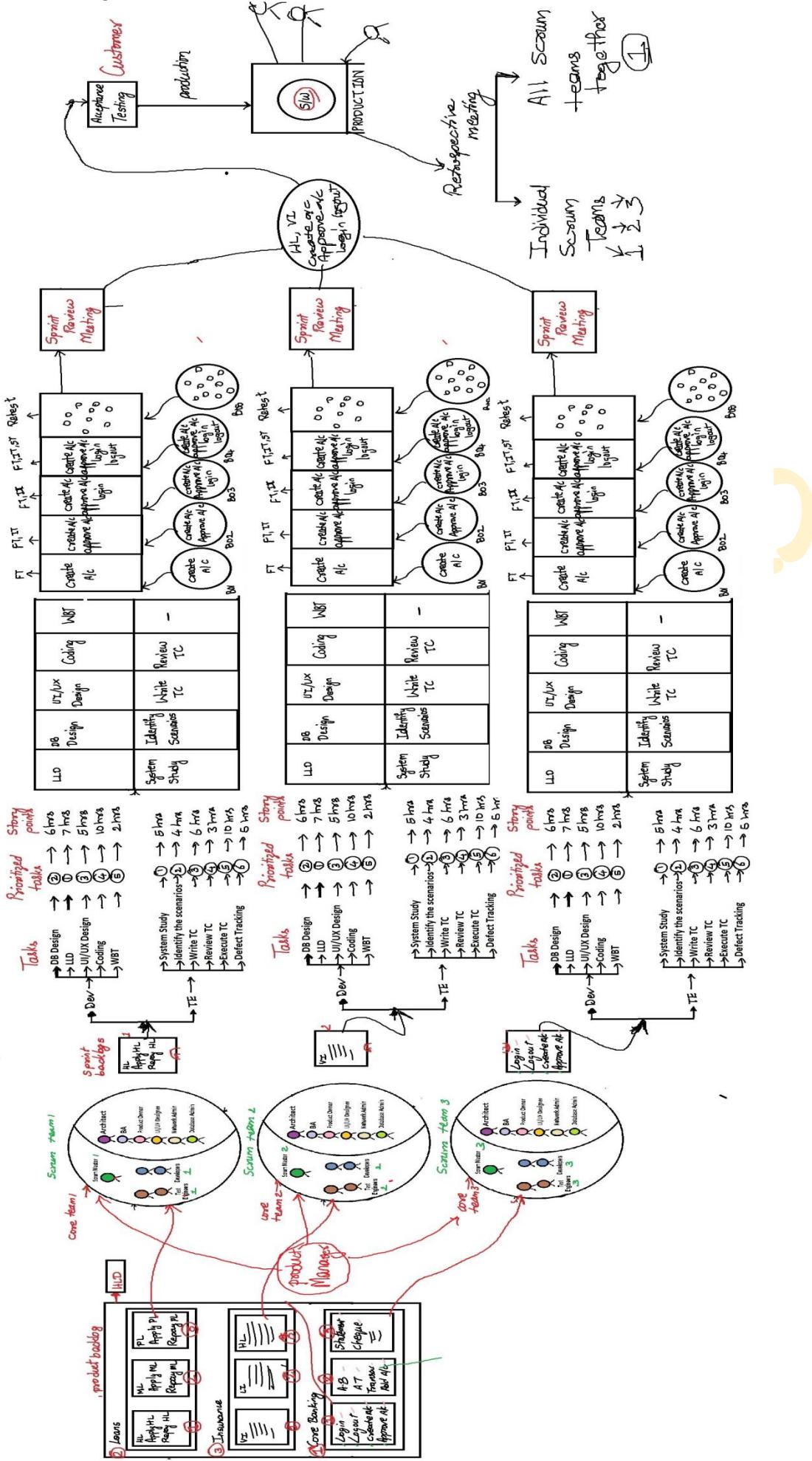
Root cause analysis

Here the entire team will discuss about the reasons/causes for getting the defect. All the reasons/causes are recorded using the fish bone diagram

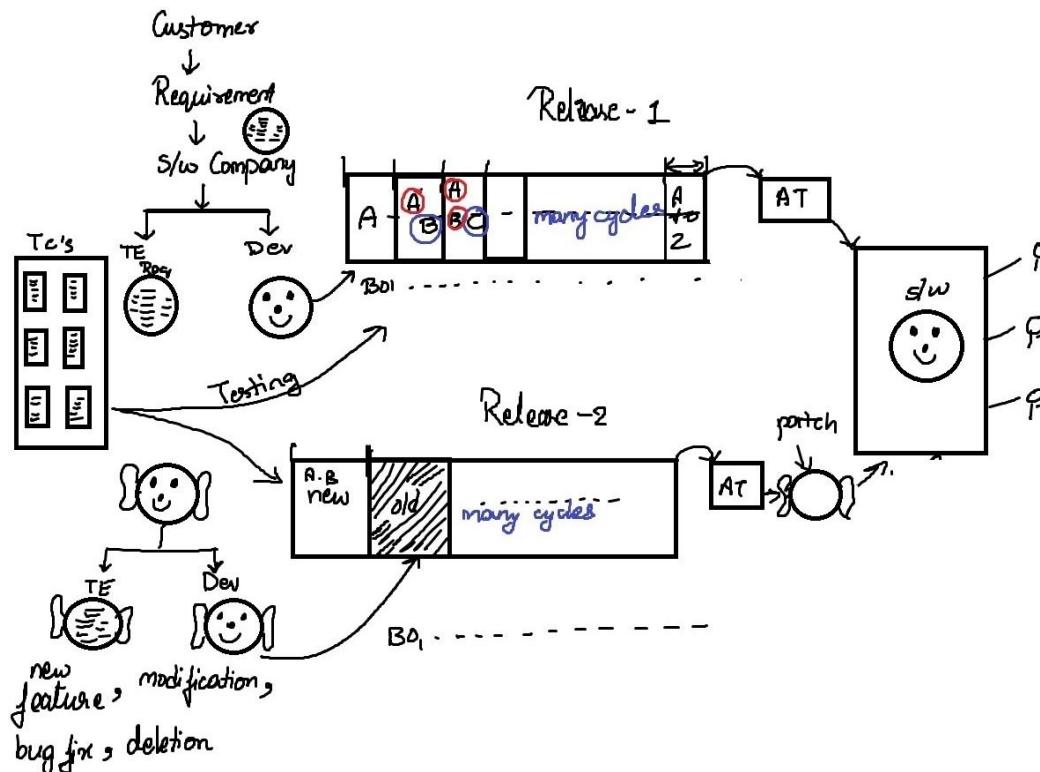
It is also called as Ishikawa method because it was found by Kaoru Ishikawa.



Agile with multiple scrum teams



REGRESSION TESTING



Testing the unchanged features to make sure that it is not broken because of the changes is called as regression testing. Here the changes can be addition, modification or removal of the features or bug fixes

OR

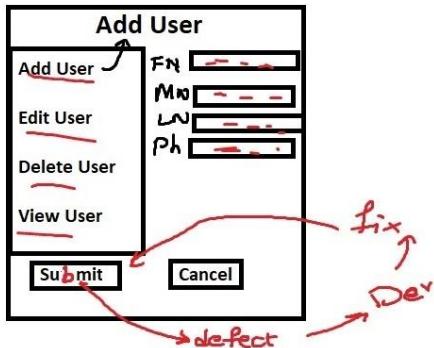
Re-execution of the same test cases in different test cycles or sprints or releases to make sure that the changes (addition, modification or removal of the features or bug fixes) does not impact the unchanged features.

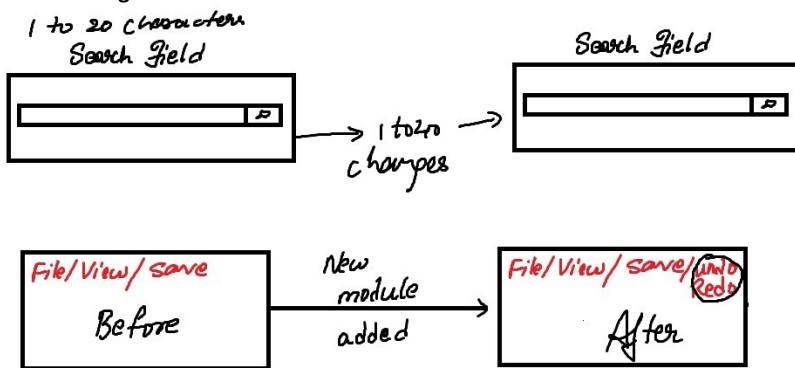
Types of Regression testing

- 1) Unit Regression testing
- 2) Regional Regression testing
- 3) Full Regression Testing

Unit Regression testing

Testing the changes or only the bug which is fixed is called unit regression testing.



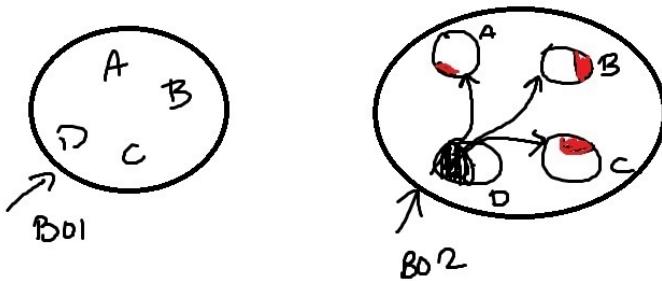


Regional Regression testing

Testing the changes and the impacted features of the changes is called regional regression testing.

OR

Testing the changes and only the impacted regions is called as the regional regression testing.



How will you identify the impacted regions while doing regression testing?

OR

How to do impact analysis?

-> Based on the product knowledge

As a TE we will be knowing how each module works in depth and also we will be knowing how all modules are related based on that knowledge, we will be able to identify the impacted areas

-> By preparing impact matrix

changed features	Sign up	Login	Logout	Compose SI	Inbox	All mails	Search	Drafts	Trash	Spam	Contact	Starred Imp..	Social
Attachment	X	X	X	✓	✓	✓	✓	✓	✓	✓	X	✓	X
Forgot Password	✓	✓	X	X	X	X	X	X	X	X	X	X	X

Here we list the changes and also all the features, then mark the impacted areas

-> By conducting impact analysis meeting

As soon as the new build comes, entire testing team meets and discuss about the list of bugs fixed and the impacted areas. List of features added, modified or removed and the impacted areas

-> By Interacting with the customer, BA, developer, testing team, we should gather the impacted areas and create an impact list based on that we should do regression testing

Advantages of Regional Regression testing

- 1) By not testing certain features we are saving TE time which in turn reduces the testing time
- 2) Test cycle duration reduces because of which the turn around time taken to deliver the product to the customer reduces

Disadvantages of Regional Regression testing

Chances are there we might miss to identify the impacted area because of which we might miss the bug

Full Regression Testing

Testing the changes and all the remaining features is called as full regression testing

Why and when we do full regression testing?

- 1) When ever too many changes are done in the product, better to do full regression testing
- 2) If the changes are made in the core features
- 3) Every 4 or 5 test cycles once, we should do full regression testing and last few cycles we should do full regression testing because we are about to launch the product- to the production and hence to not to take risk.

What is the difference between regression and retesting?

Regression testing	Retesting
Fixing the bug or doing changes might have impacted on other features to make sure that it is not broken because of the changes is called as regression testing	Testing only the changes made or the bugs that are fixed is called as Retesting

Note: repetition of FT/IT/ST on an old module itself is called as Regression testing

Progression testing

Testing the newly added feature is called as Progression testing.

Adding the new feature might affect the old features, testing the old features is called as progression testing

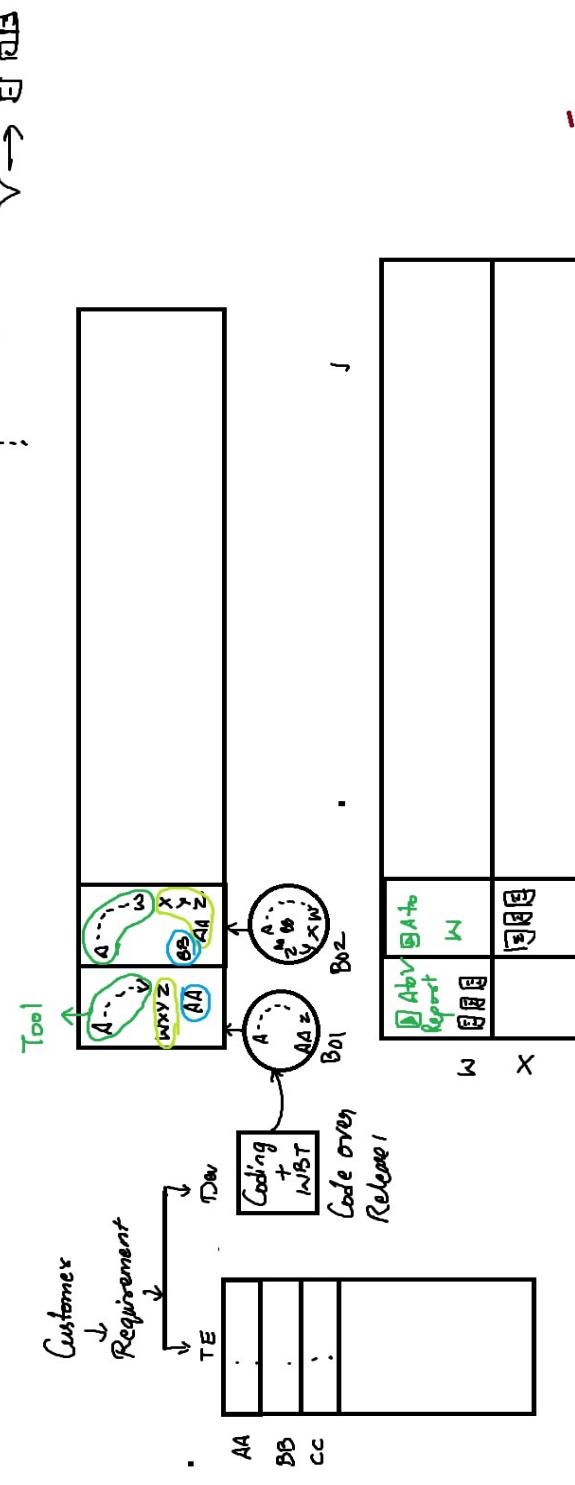
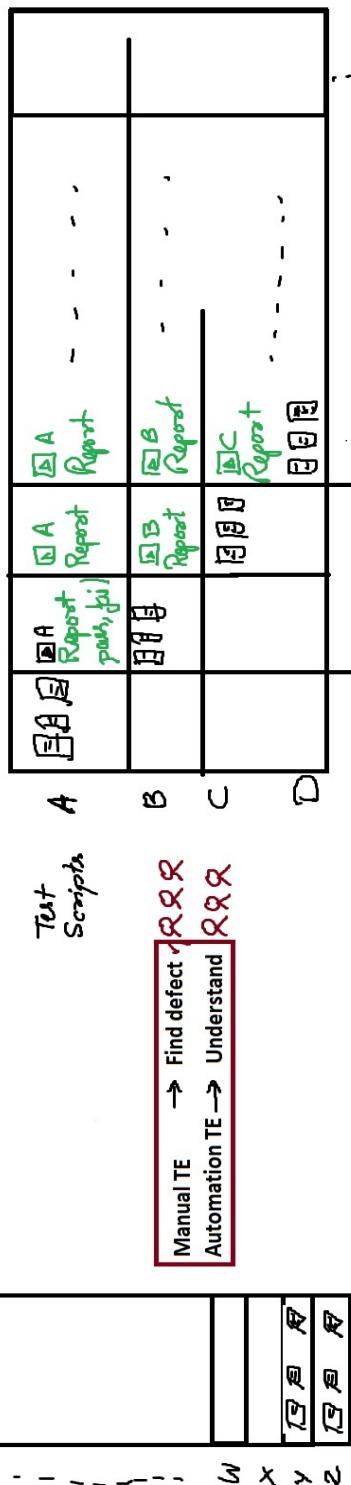
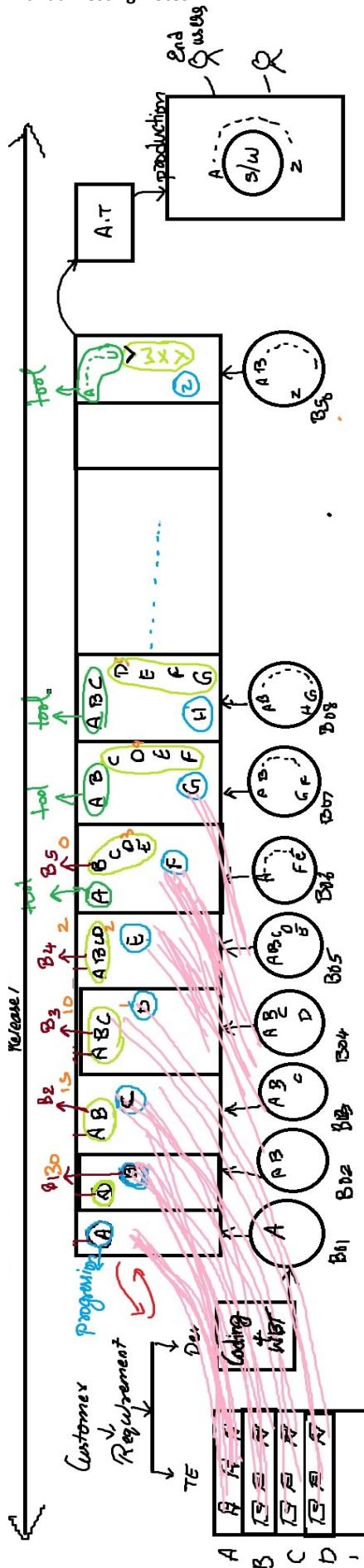
When do we do regression testing?

- 1) For Release-1 we do regression only from build -2
- 2) For Release-2 we do regression from build -1 only

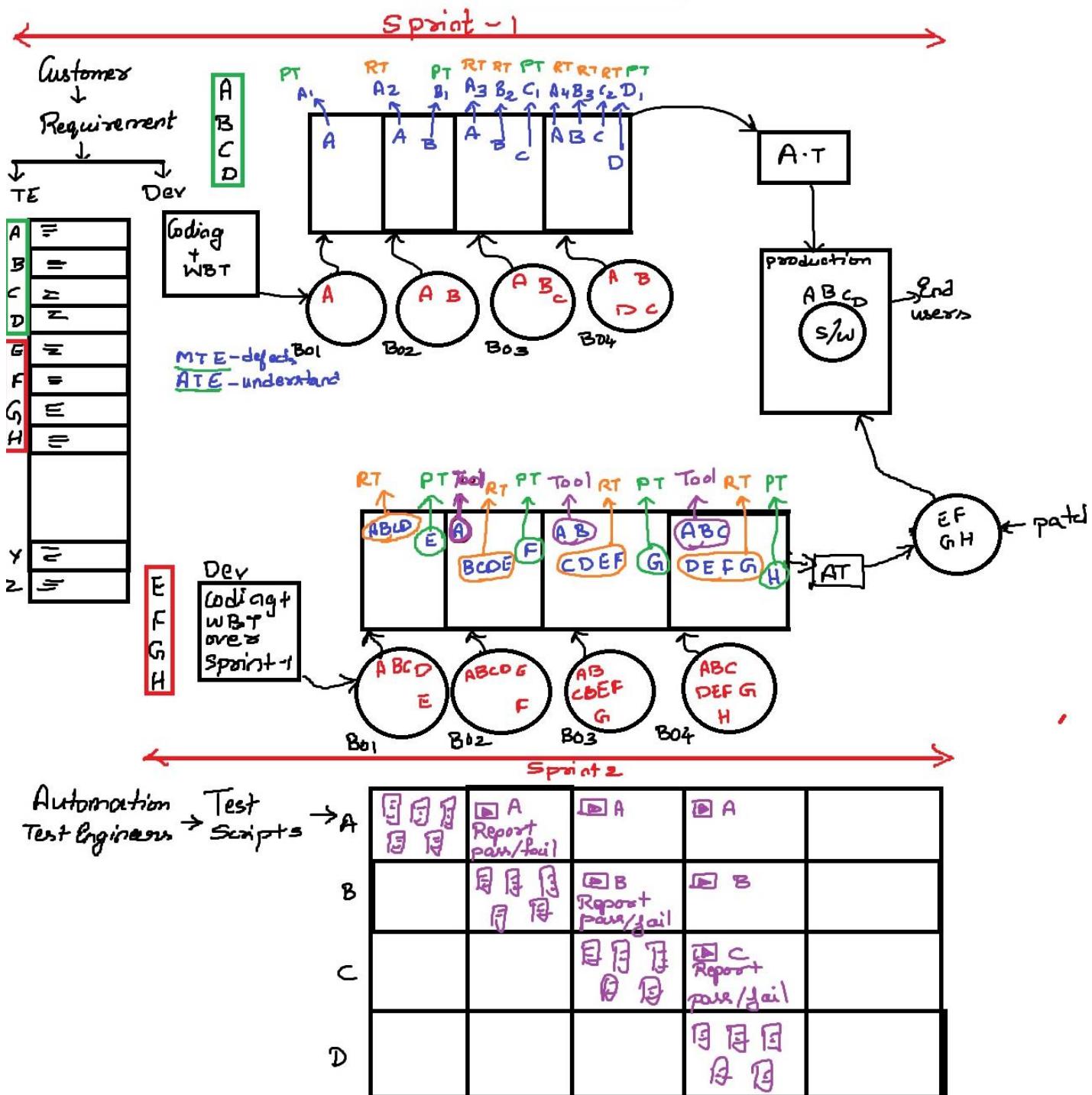
What are the drawbacks of Manual Regression testing?

- 1) Manual testing is repetitive in nature . Over the period of time, it becomes monotonous because of that TE may not be effective in testing
- 2) As the size of the application increases test cycle duration also increases because of which the turn around time taken to deliver the product to customer increases.
- 3) Man power needed is more and hence it is expensive
- 4) More effort id needed
- 5) More costly

Automation in Regression Testing



Automation in Agile



What is the role of Manual test engineer?

- 1) Test new features manually
- 2) Test modified features, bug fixes and removed features manually
- 3) Test the features which are not automated
- 4) Find the bugs and communicate to developers
- 5) Write and update manual test cases

What is the role of automation test engineer?

- 1) Understand the feature
- 2) Understand the test cases

- 3) Convert the manual test cases in to automation test scripts (Manual test case of stable features)
- 4) Execute the automation test scripts
- 5) Maintain the automation scripts
 - a) When ever the requirement changes test cases should be changed
 - b) When ever the test cases change, the automation test scripts should be changed
 - c) Also if any problems come in old scripts, we should fix it.

Why we go for test automation?

- 1) We do test automation to reduce manual repeated regression testing effort.
- 2) To reduce test cycle duration
- 3) To reduce the turn around time taken to deliver the product to customer
- 4) To reduce number of test engineers
- 5) To reduce the cost of testing
- 6) To improve the test efficiency
- 7) To have consistency in quality of test execution

We go for automation only when the product is stable and not in the early stages . Why?

- 1) In early stage of product development, our aim is to catch more number of defects, so we should do manual testing because automation cannot think itself
- 2) In early stages of product development, lot of bugs will be predicted , if we go for automation, we will not be able to complete automation script execution
- 3) In early stages of product development, customers will be doing lot of requirement changes, so changing the manual test cases will be easy, that's the reason, here we don't go for automation. If we go for automation, TE will be spending time in modifying the script rather than testing the software

When to go for automation?

- 1) When the product is stable
- 2) When there are no blockers or critical defects
- 3) Once after the manual testing is completed for 3 to 5 test cycles
- 4) When the product is manually tested for 1 or 2 releases
- 5) When there are no major requirement changes done by the customers

Q-SPIDERS