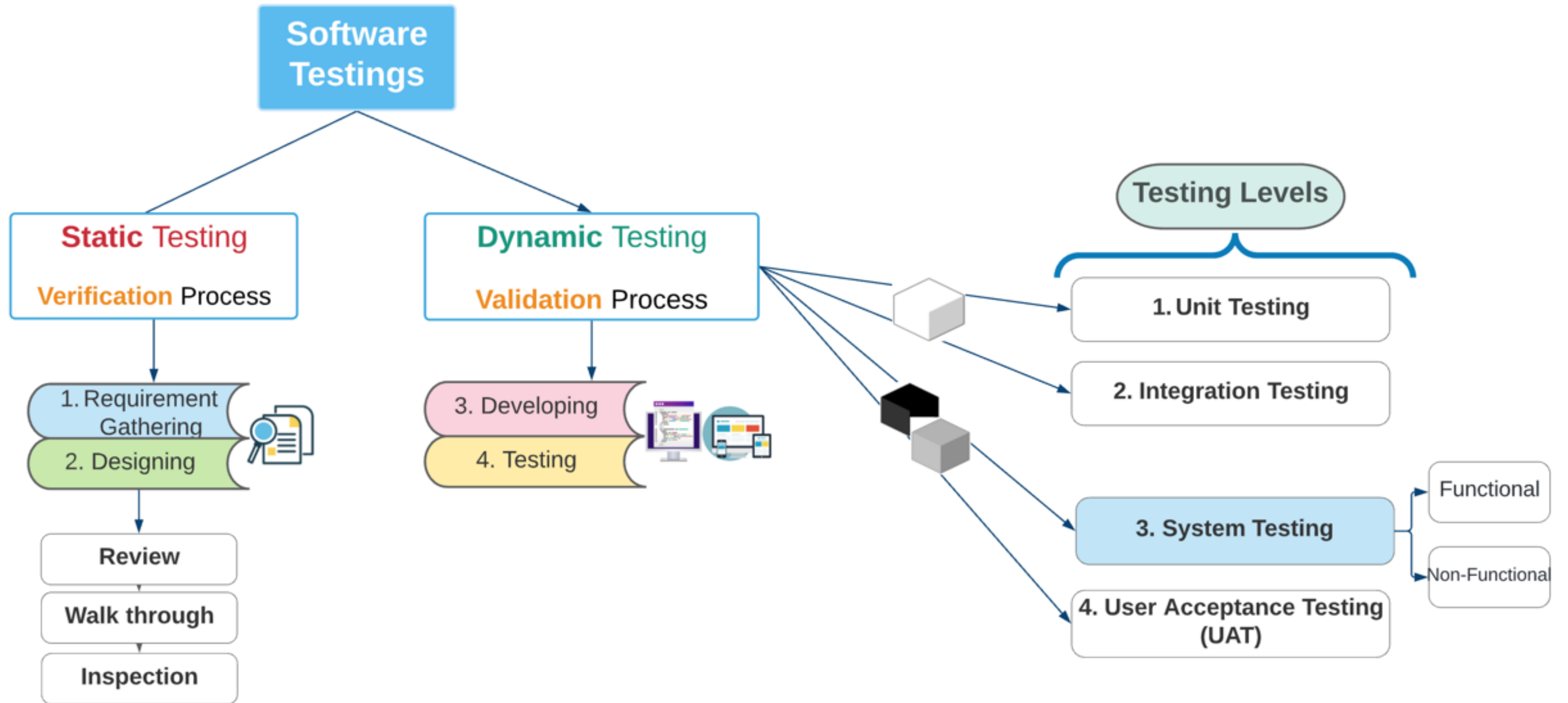
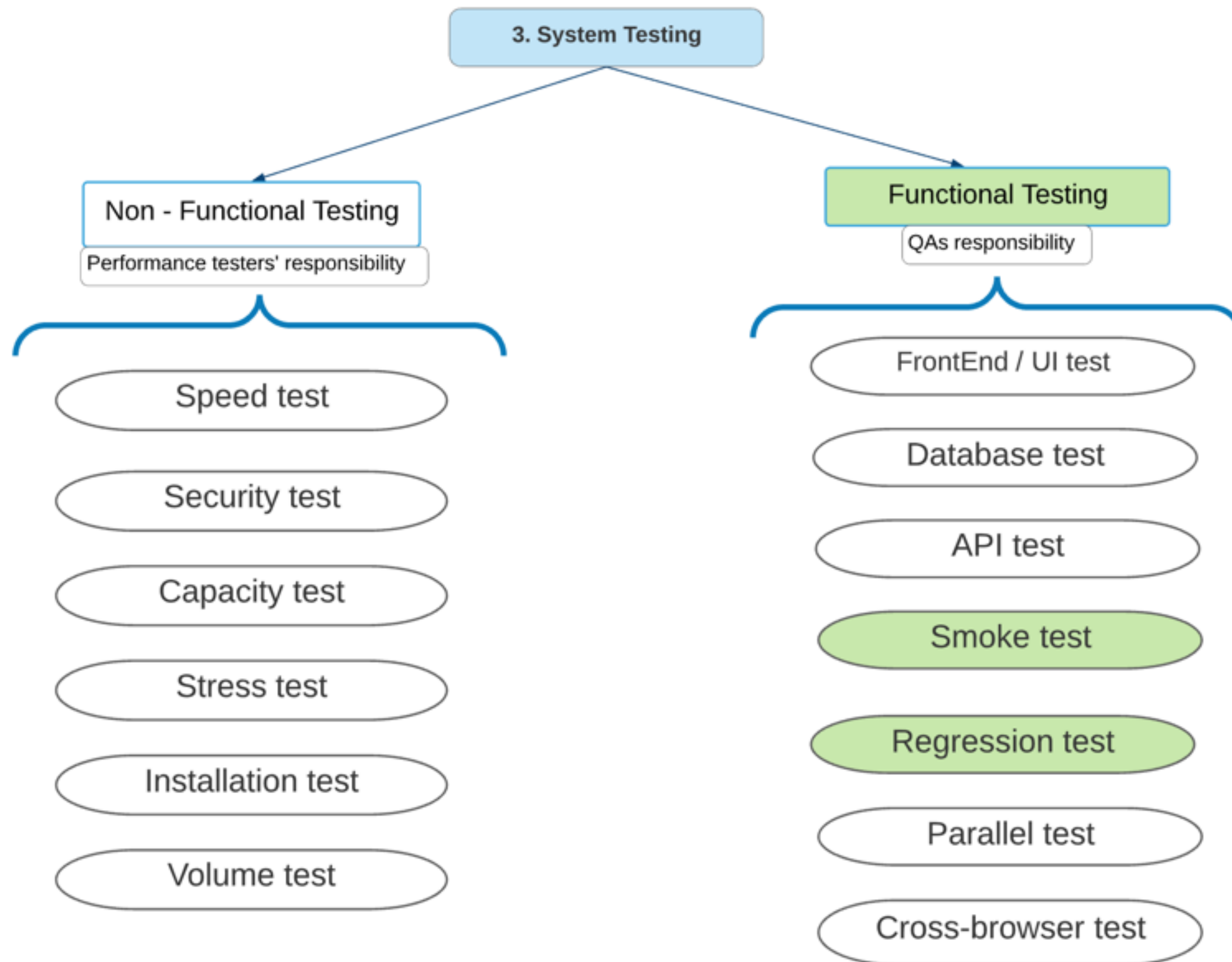


REGRESSION TESTING



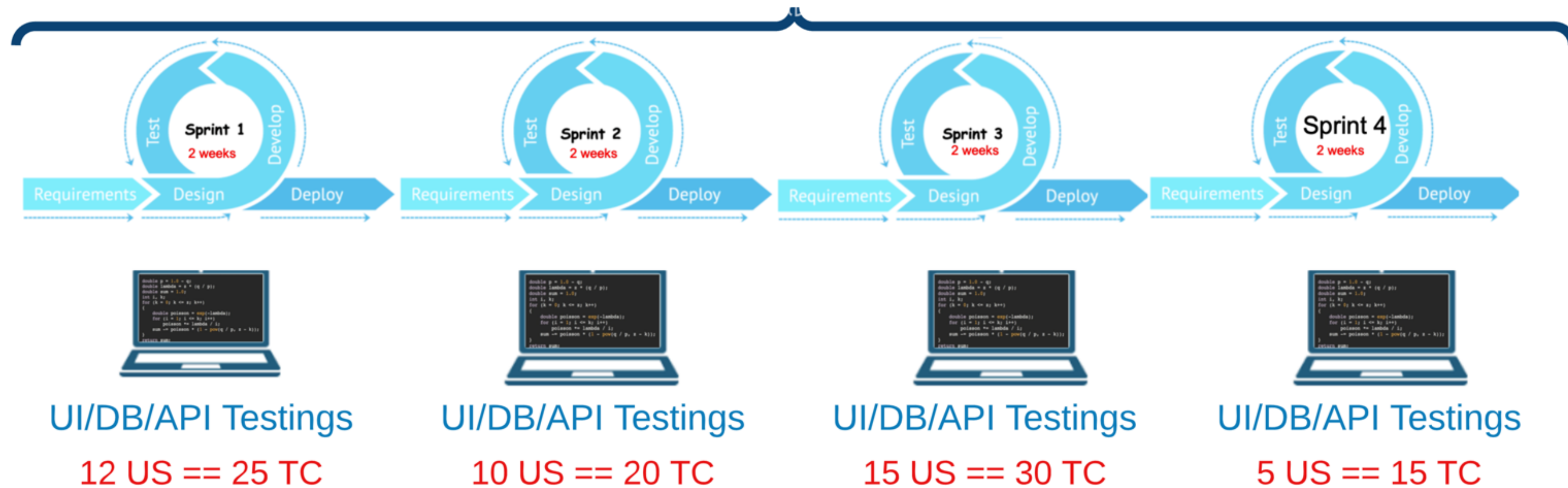




What is Regression Testing ?

- Regression testing **confirms** that software **previously** developed **features** are still **working well with** the **new added feature**.
- The **main goal** is to **make sure** that a **product functions as expected after any changes** made to the application.
- Regression testing is preformed when:
 - Before release (not last testing before the release date, UAT is the last testing)
 - New requirements are added - delete, update, add, etc
 - New features are added
 - A previous defect or bug is fixed
 - Performance issues are fixed

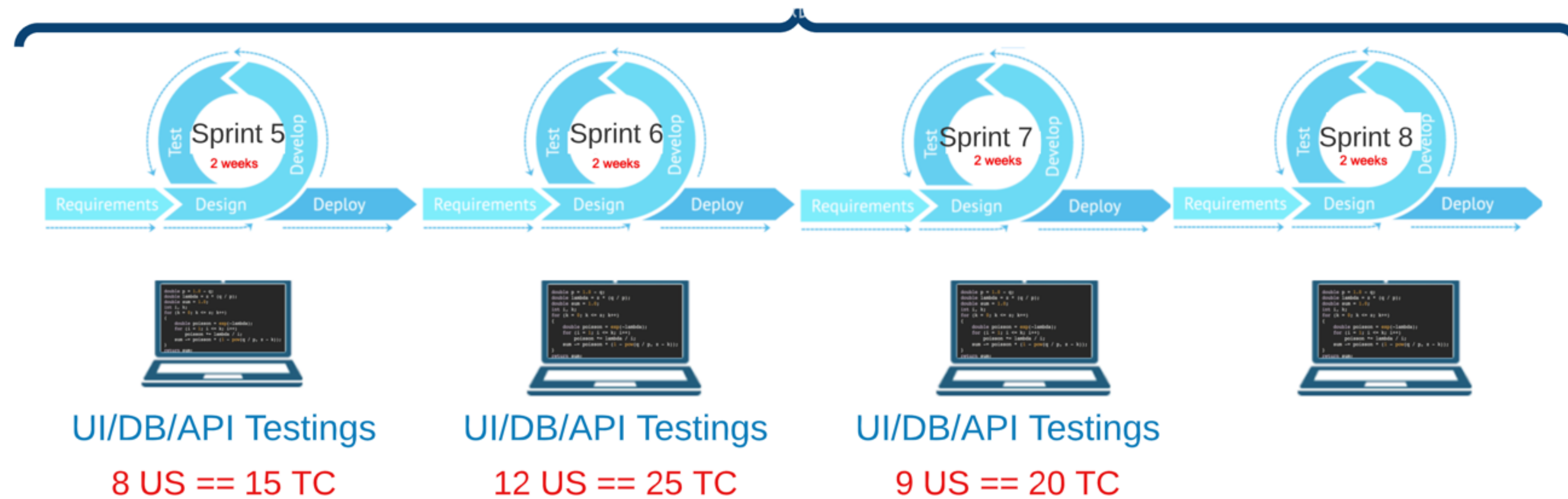
Release 1



Created & Tested a **total** of **90 TC** during the Release v1.0

Regression testing -> one week of Sprint4 is for Regression Testing
Test ALL 90 TC manually & automatically

Release 2



Created & Tested a **total** of **60 TC** during the Release v2.0

Regression testing -> Whole Sprint8 is for Regression Testing

90 + 60 = 150 TC in total

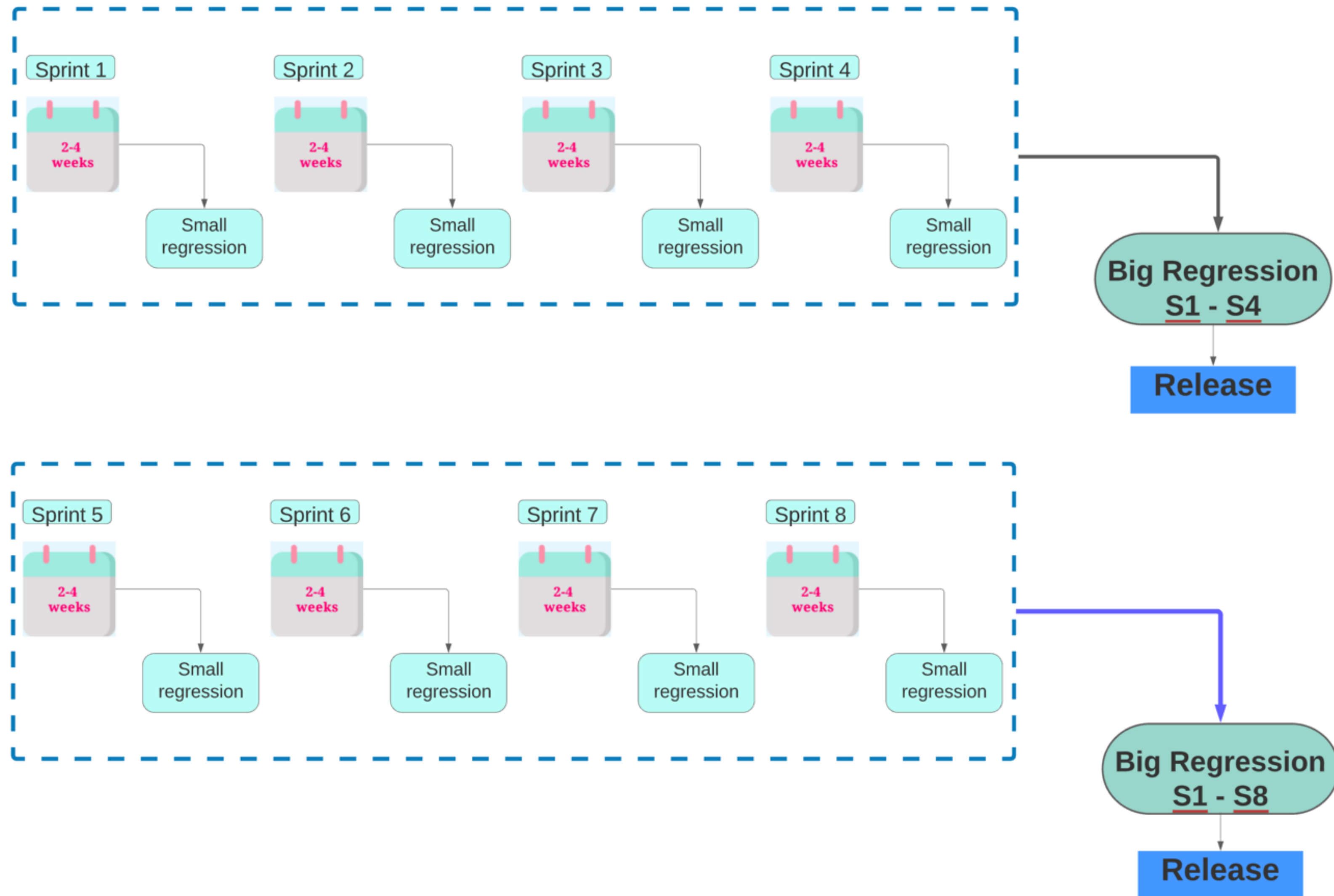
Automatically test the TC that has code (80-85% of total)

Manually test the TC that is not automated (20-25%)



Regression testing -> Entire Sprint 49 is for Regression Testing
Approximately 700 TC in total over the 2 years

Automatically test the TC that has code (80-85% of total)
Manually test the TC that is not automated (20-25%)

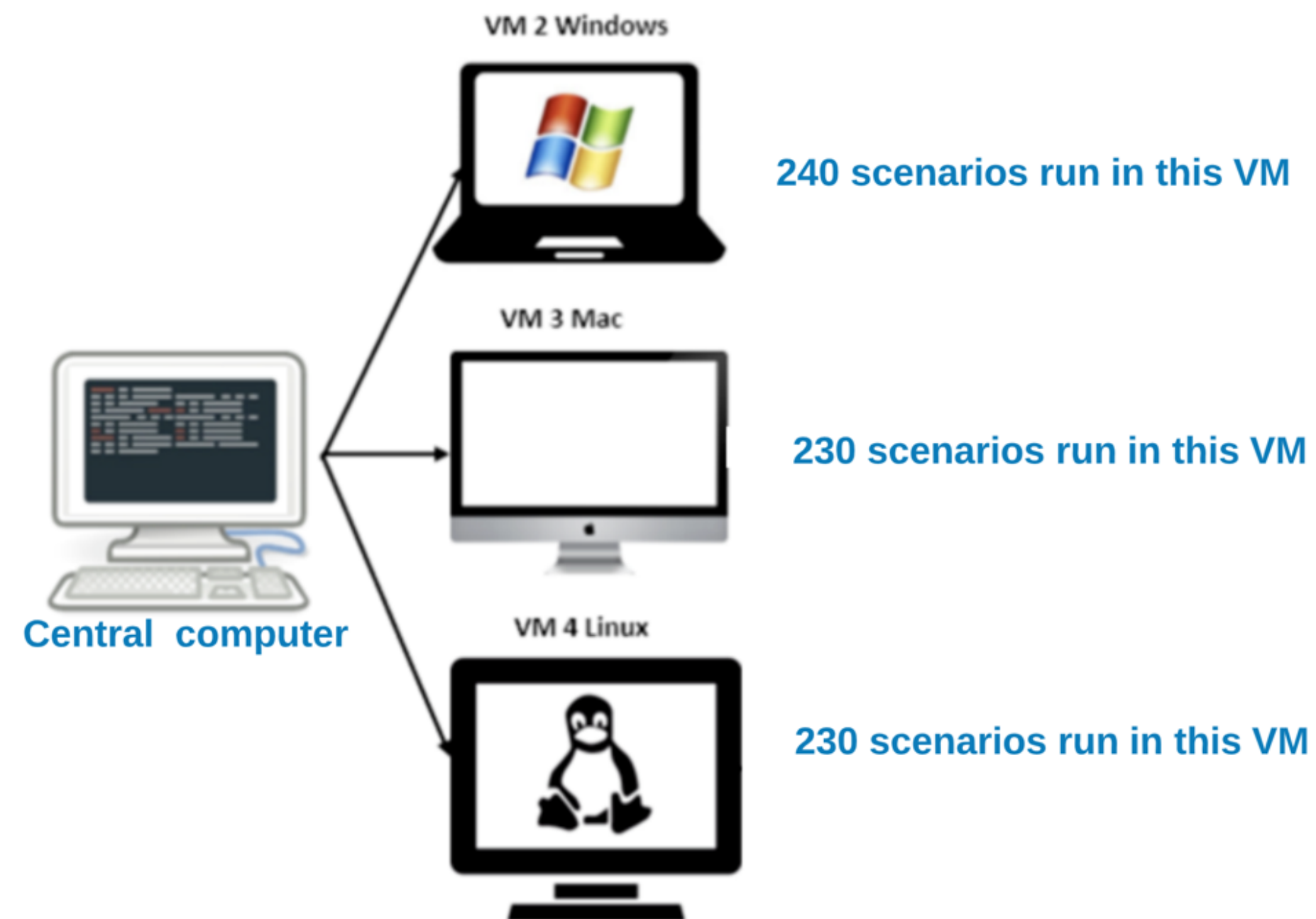


Major & Minor Regression						
	Release	Sprints	Test case/Scenario #	Minor/Small Regression	Major/Big Regression	Time to run
1	Release V1.0	Sprint 1	20 TC	20 TC	-	
2		Sprint 2	15 TC	15 TC	-	
3		Sprint 3	15 TC	15 TC	-	
4		Sprint 4	5 TC	5 TC	55 TC	Automation: 55 TC - 30m manual: 55TC - 2-3 days
5				The app is released to the Production		
6	Release V2.0	Sprint 5	15 TC	15 TC	-	
7		Sprint 6	20 TC	20 TC	-	
8		Sprint 7	20 TC	20 TC	-	
9		Sprint 8			110 TC	automation -> 110 TC - 1 hour manual -> 110 TC - 5-7 days
10				The app's new feature is released to the Production		
11	Release <u>V3.0</u>	Sprint 9	20 TC	20 TC		
12		Sprint 10	25 TC	25 TC		
13		Sprint 11	15 TC	15 TC		
14		Sprint 12			170 TC	Automation -> 170 TC - <u>2hours</u> manual -> 100 TC - 5-7 days
15				The app's new feature is released to the Production		

Selecting Test Cases for Regression Testing:

1. Major functions test cases,
2. Recently code or function changed
3. Visible functions to the users
4. The business team prioritize some functions
5. Test cases that have more steps
6. requires to test with more test data
7. Repetitive, high-frequency test cases

Parallel testing -> Parallel testing is an automated testing process in that QAs run Regression testing multiple TC simultaneously, against different browsers and VMs to reduce testing times.



700 TC -> run in 1 computer -> 5-6 hours

700 TC -> run with 3 VM -> 1 hour

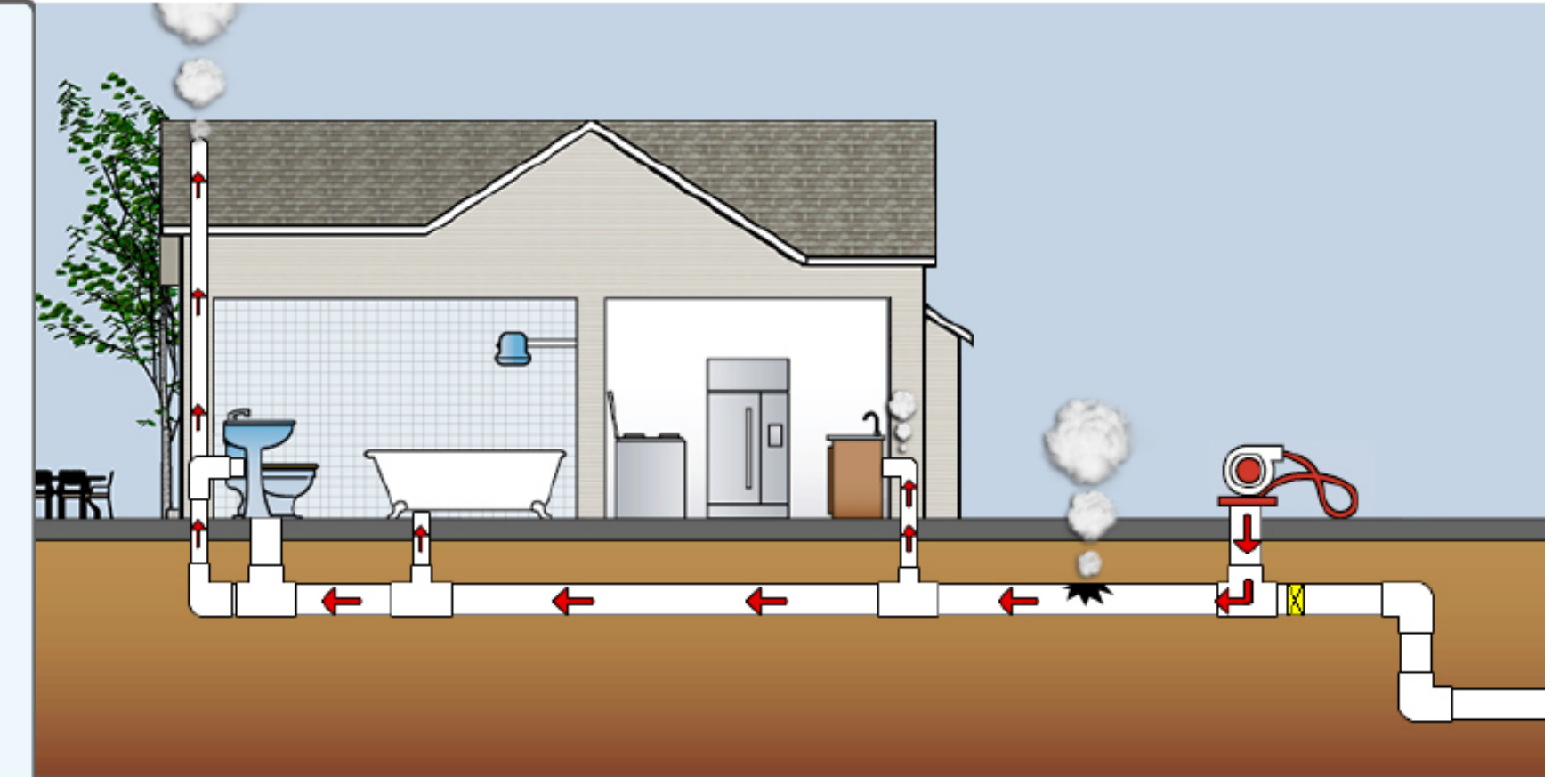
Running the Regression Testing in parallel

How do you run your Regression test?

- Regression testing is done to check if the new functionality works with the old ones, also to make sure that a change hasn't broken any existing functionality.
- we run big regression every 2 months, before release.
- I have around **700 scenarios** in my regression test suite.
- The **automated scenarios** run with **3 virtual machines** in **parallel** that takes around **1 hour** to run.
- The **non-automated** scenarios are manually executed by all the testers which takes a week of time.



The term “**Smoke Testing**” is likely originated in the plumbing industry. **Plumbing Smoke Testing** makes **discovering leaks** and cracks in a project entire pipe. Then software development industry used "Smoke test" title.



Plumbing pipe line system

What is Smoke Testing ?

- Smoke testing is **performed everyday** to determine **whether the application is stable or not**, and the qa environment is up and running .
- The Smoke test is **meant to be quick to execute**, and **its goal is** to assure that the major features of your system are working as expected.
- It consists of a minimal set of tests run.

Is smoke testing performed automatically or manually?

- It is done **automatically** through Jenkins, Github, and virtual machines.
- **If there is any bug/error** found from smoke test, **then QA team** need to **test manually** to confirm the bug and identify which major function has the bug.

How often do you perform smoke test?

- Our Smoke test runs once every day morning.

(Note: There are some companies run smoke test twice a day - one before working, one after working)

How many scenarios do you have in your smoke test?

- Ideal answer can be between 50 - 90 scenarios.

(In real work, the number is as little as 5 scenarios, or big as 200 scenarios. However, in the interview, you want to answer the most common situation so you won't be questioned more detail about it.)

How long does it take to run smoke test?

- Smoke test should takes around 15~30 minutes to run.

Who select scenarios to smoke test list? How often add more scenarios to smoke test?

- Smoke test scenarios are **selected** in a meeting with all the testers.
- **More scenarios** are added to run as smoke test, when there is a major function is developed to the app.

How to get smoke test result?

- Since smoke test is automatically tested, the **test result** is also automatically prepared and sent to each tester's **email**.



17 lines (15 sloc) | 411 Bytes

```
1 @logout
2 Feature: Log out feature
3   @smoke ←
4   Scenario: Driver
5     Given I login as a "driver"
6     When I logout
7     Then the page title should be "Login"
8   @smoke ←
9   Scenario: Sales manager
10    Given I login as a "sales manager"
11    When I logout
12    Then the page title should be "Loginww"
13 # @smoke ←
14 # Scenario: Store manager
15 #   Given I login as a "store manager"
16 #   When I logout
17 #   Then the page title should be "Login"
```



Jenkins

Project name: Some Test

Code URL: www.GitHub.com.companyname/projectname

Goal : tags="smoke"

Time : 5 AM , Mon-Sun

Report type: Cucumber Report

Send-to:

tester1@companyName.com

tester2@companyName.com

tester3@companyName.com

Can be added developers' emails:

developer1@companyName.com

developer2@companyName.com

developer3@companyName.com

Project	Number	Date
Smoke test	1078	03 Jun 2021, 5:25

Scenarios Statistics

The following graphs show passing and failing statistics for features

1078th time that this
smoke test has been

Scenarios



Scenarios	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Smoke test	75	0	0	0	0	75	75	0	4	24.906	Passed

Can you tell me about your Smoke Test?

- In my current project, we perform the Smoke test to check if the application is up and running by checking the basic functionalities. They are meant to be quick to execute just to assurance the major features of the app are working as expected.
- Our smoke test runs every morning at 7 AM through Jenkins in the QA env.
- There are 85 scenarios in my Smoke test; it takes around 25m to run
- I start my work by checking my email every day; Jenkins sends me the Smoke test result.



Req collection

Static testing - SRS



Design app & SDS

Static testing - SDS



Develop the app - code

Dynamic Testing

White-box testing

1. Unit-testing

2. Integration-testing

Integration testing is passed?



SOFTWARE TESTING

Test the software

Dynamic Testing

Black & Gray box testing

System-testing

3. Functional-testing

4. Non-Functional-testing

The client wants to add more feature?



SOFTWARE MAINTENANCE

Any bugs from the production
will be fixed

Regression testing is passed?



UAT

Alpha - testing -> automation
engineers/QAs/SDET

Beta - testing -> the client



APP RELEASE

The client is happy?