1305308 Platform Development

Week 09: Platform Testing

SEC 01 SEC 02

By: Vittayasak Rujivorakul

Class Schedule – Before Midterm Exam

15-08-2023	Introduction to Platform Devlopment
22-08-2023	Platform Architecture and Design
28-08-2023	Get Requirements on Department
05-09-2023	Front-End Development
12-09-2023	Back-End Development
19-09-2023	Platform Security
26-09-2023	Project Progression
03-10-2023	Midterm Examination (In Class Exam)

09-15 Oct 2023 is Academic Midterm Examination

Class Schedule – Before Final Exam

17-10-2023	Platform Testing and Quality Assurance
24-10-2023	Platform Deployment and DevOps
31-10-2023	Platform Analytics and Insight
07-11-2023	Platform Maintenance and Support
14-11-2023	Workshop and Project
21-11-2023	Mini Project Present
28-11-2023	**Final Examination (In Class Exam)**

04-15 Dec 2023 is Academic Final Examination

Outline

- Importance of testing in platform development
- Type of testing: unit, integration, and end-to-end
- Automating test for continuous integration and deployment
- Ensuring platform reliability and performance

Importance of testing in platform development

Why testing the software

- Helps in saving money – cost-effectiveness
- Security – vulnerability test
- Quality of the product – compatibility
- Satisfaction of the customer
- Enhancing the development process – quality assurance
- Easy while adding new features
- Determining the performance of the software

7 Software testing principles

TESTING SHOWS THE PRESENCE OF DEFECTS: If there are no bugs found after testing, it is not proof that there is not any.

EXHAUSTIVE TESTING IS NOT POSSIBLE: To maximize testing efficiency, focus on your priorities rather than testing every possible combination.

EARLY TESTING SAVES TIME AND MONEY: Fixing bugs at their root is more cost- and time-effective than in the further development stages.

DEFECTS CLUSTER TOGETHER: Usually, most bugs are clustered in specific software areas, and testing should focus on them first.

BEWARE OF THE PESTICIDE PARADOX: Tests should be updated over time to remain effective.

TESTING IS CONTEXT DEPENDENT: Testing is conducted differently for specific software types.

ABSENCE-OF-ERRORS IS FALLACY: Fixing all defects does not guarantee the absence of bugs.

Software Testing

Functional vs Non-Functional

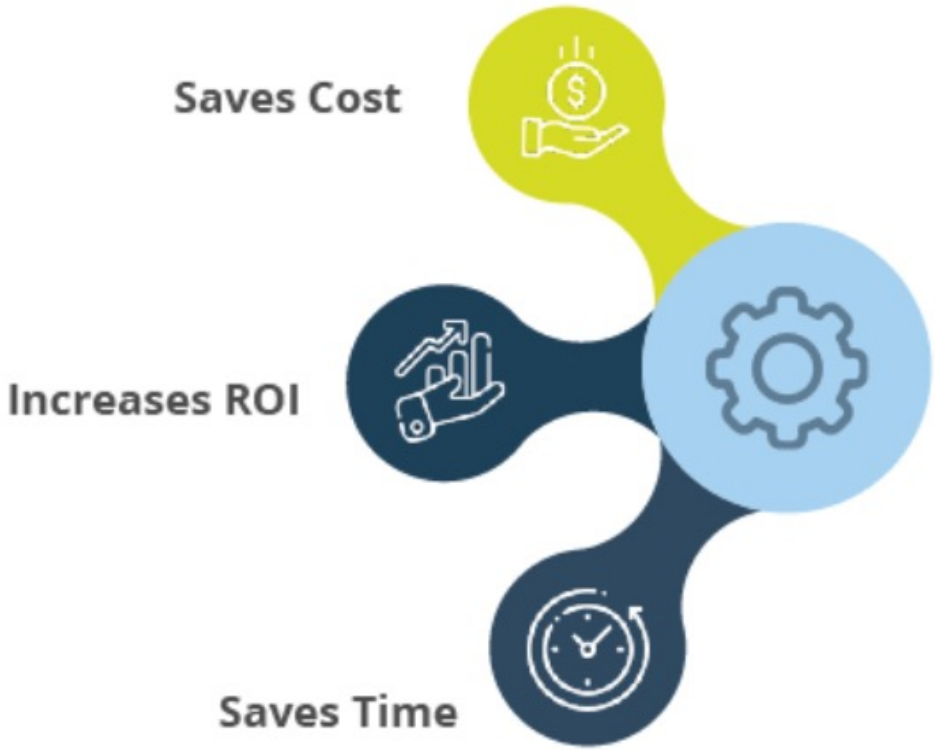
Functional	Non-functional
checks **apps operations and actions**	checks **apps**
based on **business requirements**	based on **customer expectations**
checks **actual vs expected results**	checks **response, speed, and other requirements**
manual	**automated**
tests the **functionality**	tests **how the functionality is performed**
defines **what** to test	defines **how** to test

Benefits of Software Testing

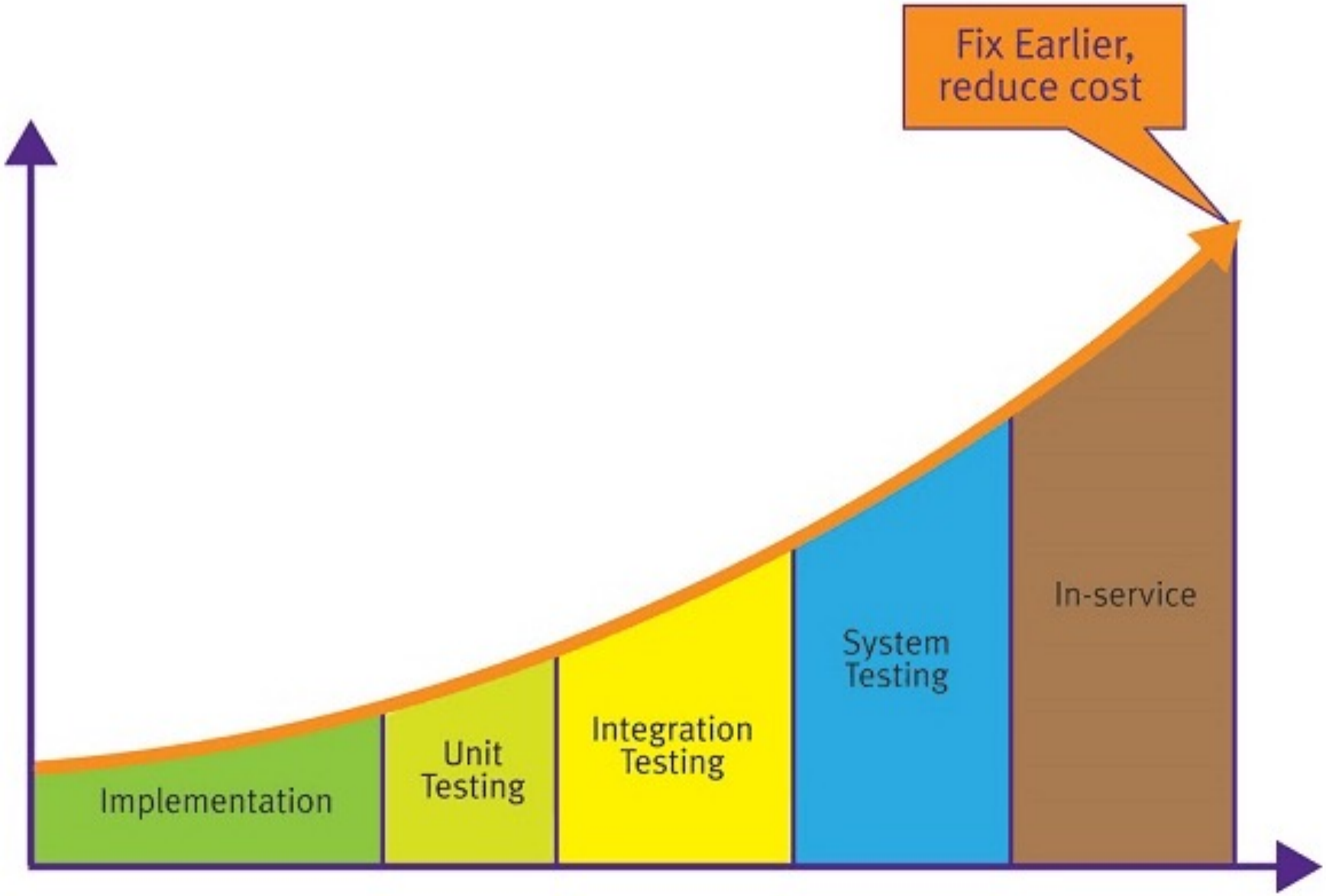
Application-Wise Benefits

- Increases test coverage
- Ensures test accuracy
- Eases regression testing time
- Facilitates with re-usable test scripts
- Helps to Validate Complex Scenarios effectively

Cost-Wise Benefits

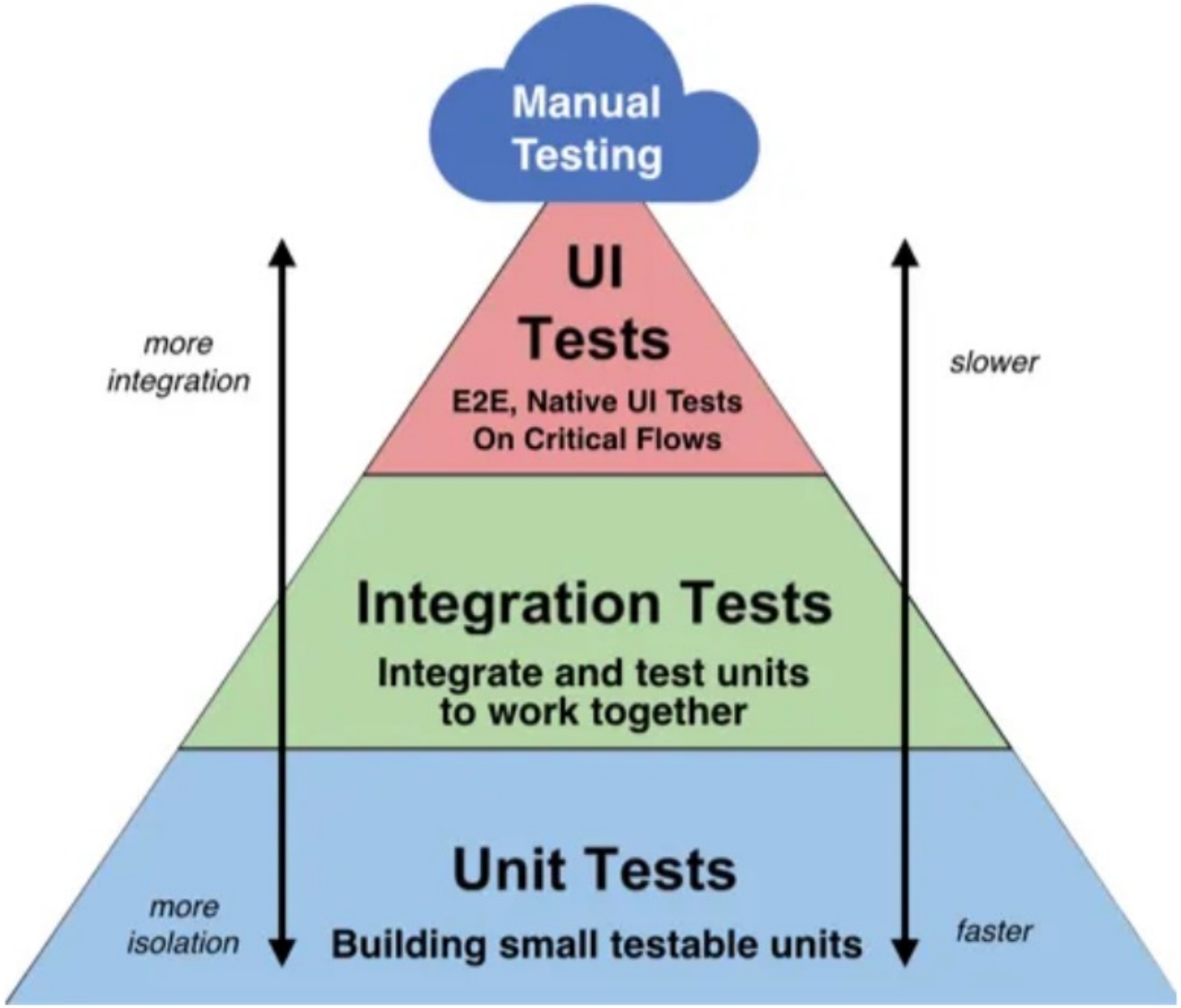


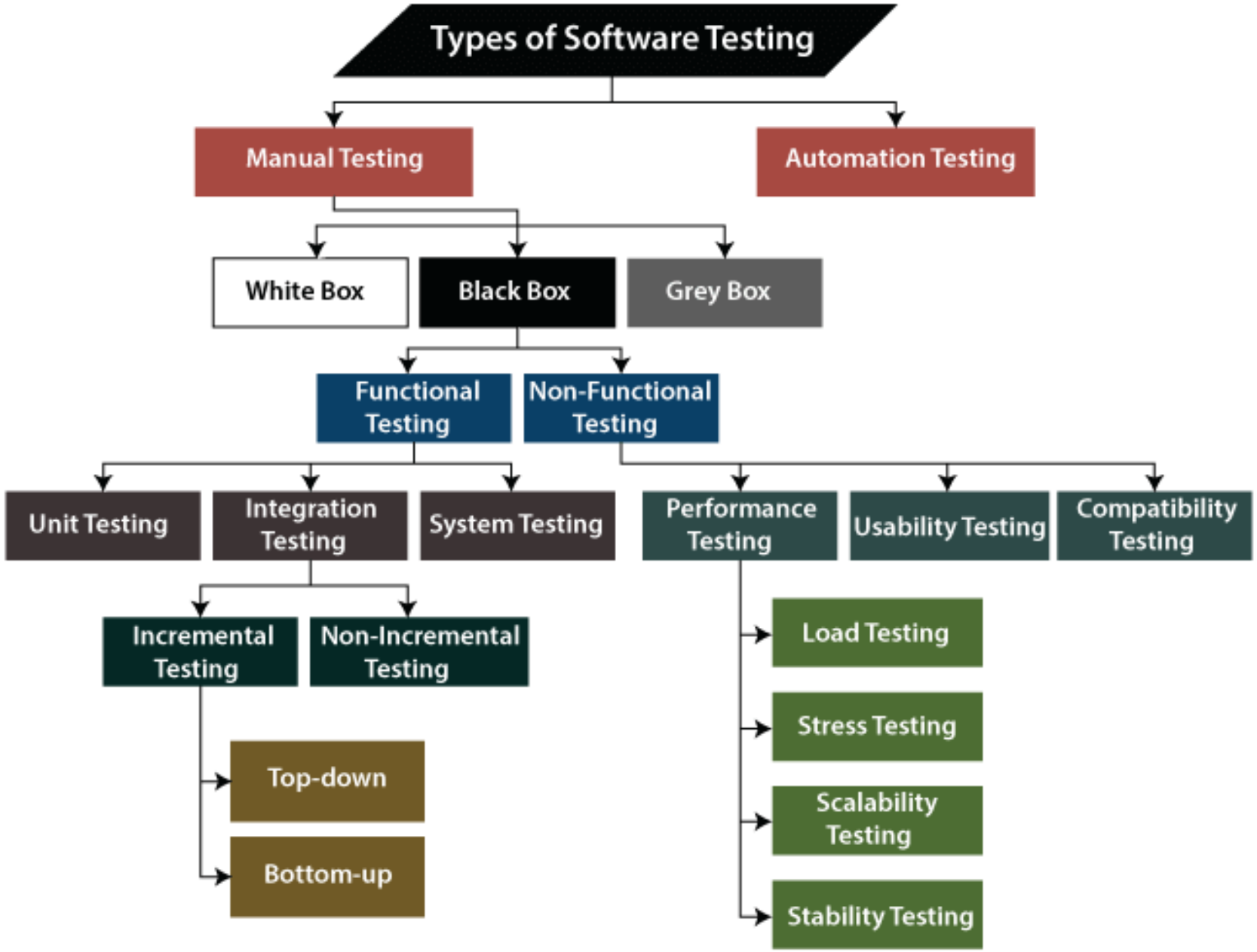
The Cost of Defects



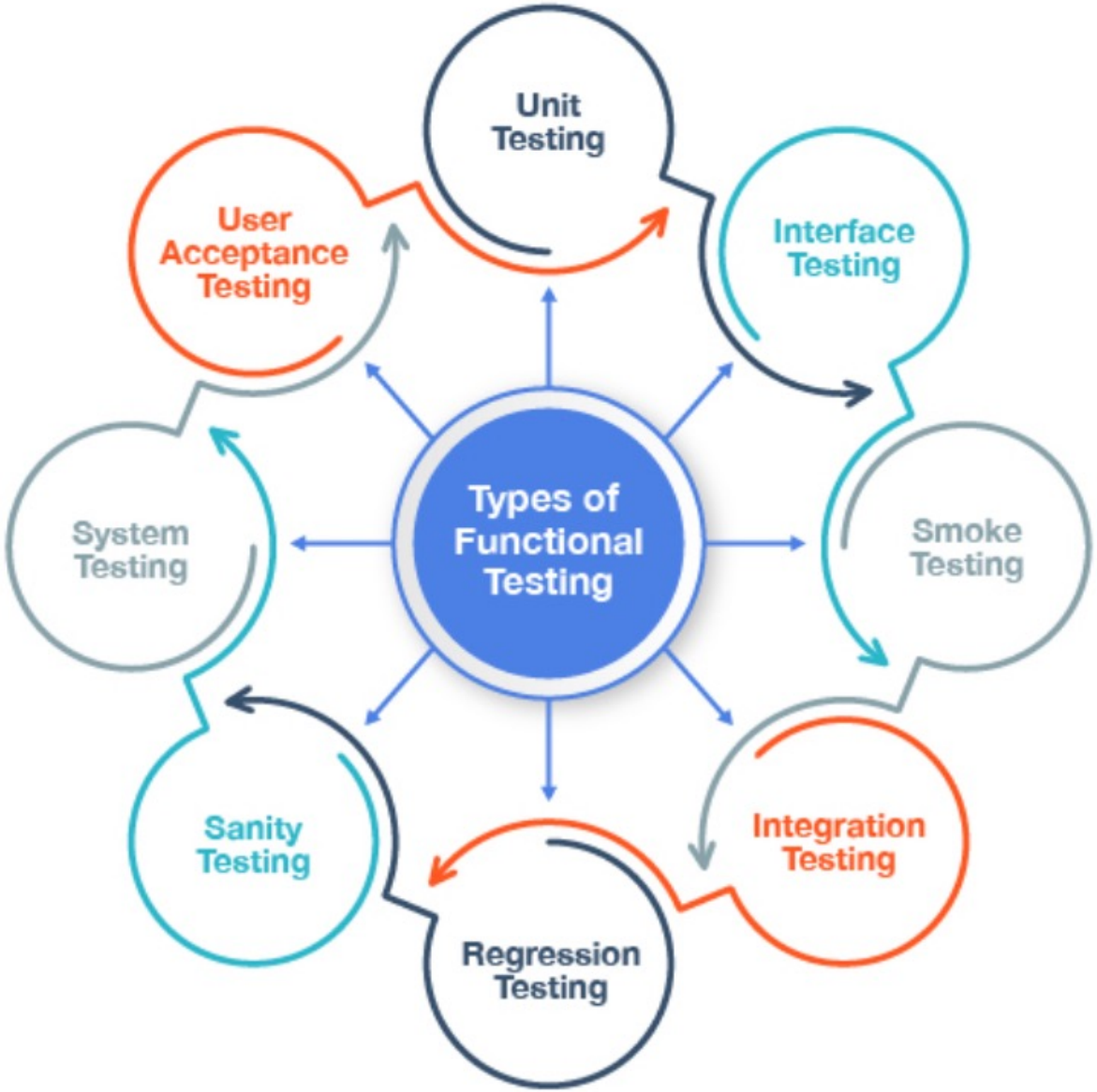
Type of testing: unit, integration, and end-to-end

Type of Testing

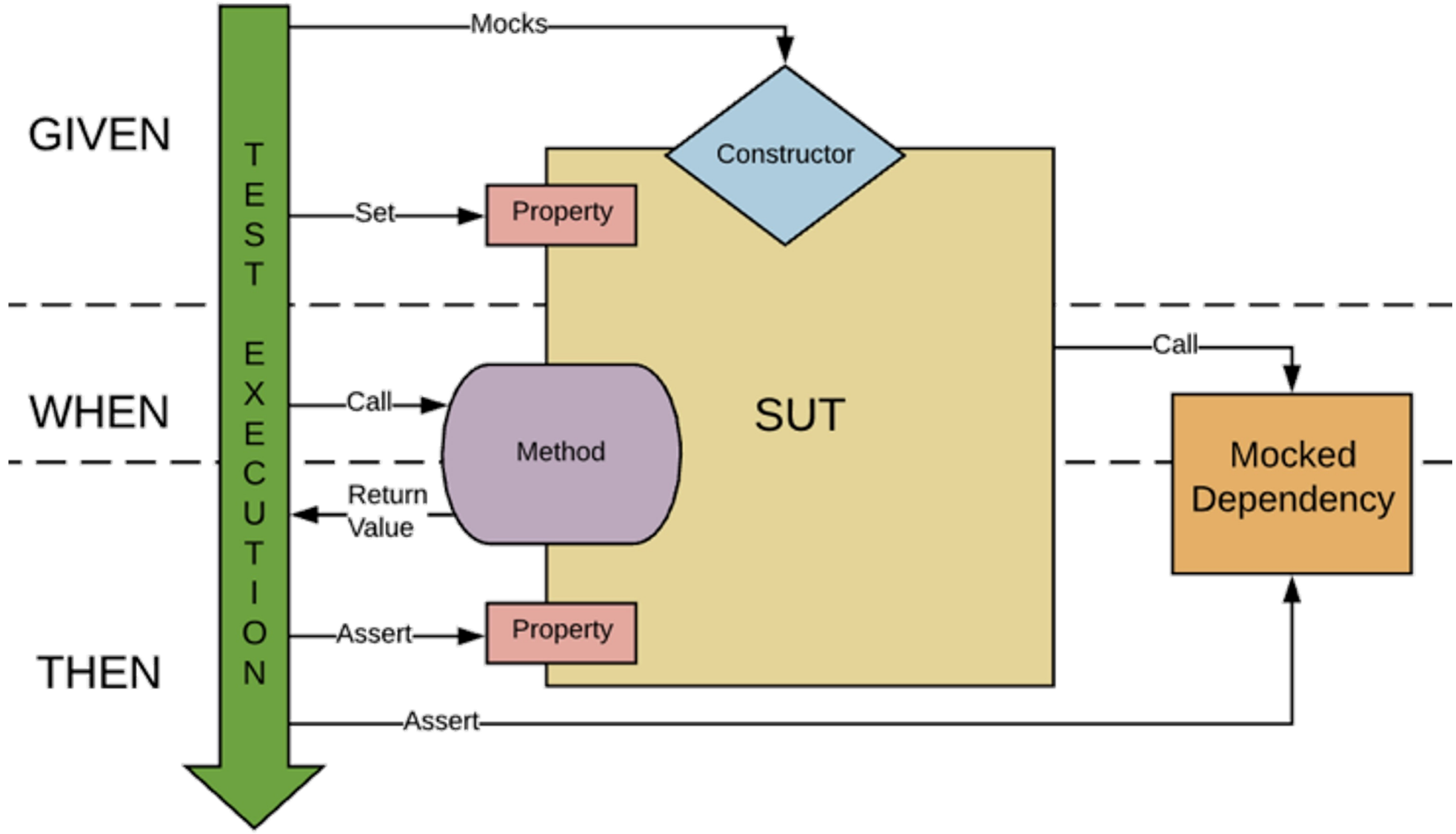



Types of Software Testing
Manual Testing
Automation Testing
White Box
Black Box
Grey Box
Functional Testing
Non-Functional Testing
Unit Testing
Integration Testing
System Testing
Performance Testing
Usability Testing
Compatibility Testing
Incremental Testing
Non-Incremental Testing
Top-down
Bottom-up
Load Testing
Stress Testing
Scalability Testing
Stability Testing

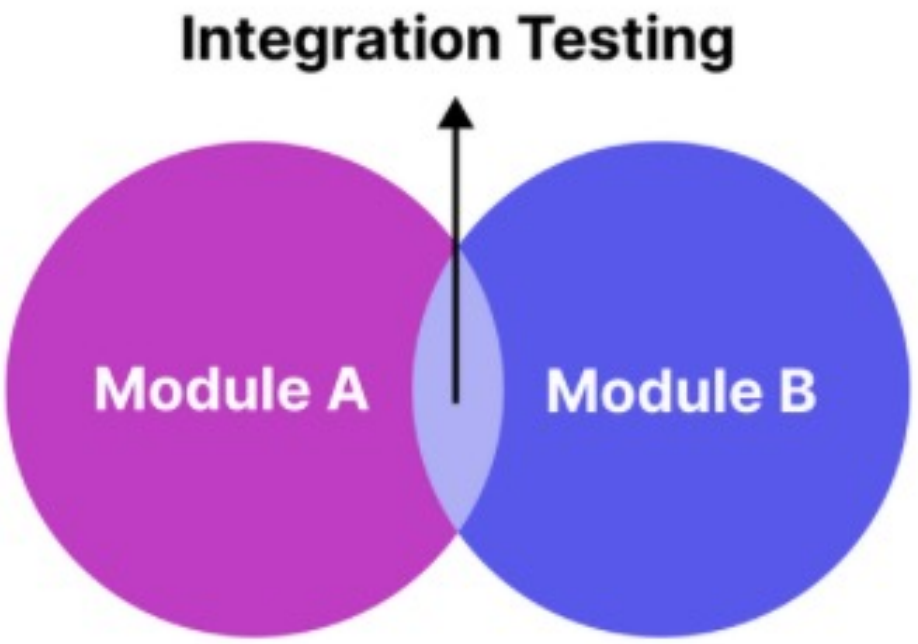
Functional Test vs Non-Functional Test

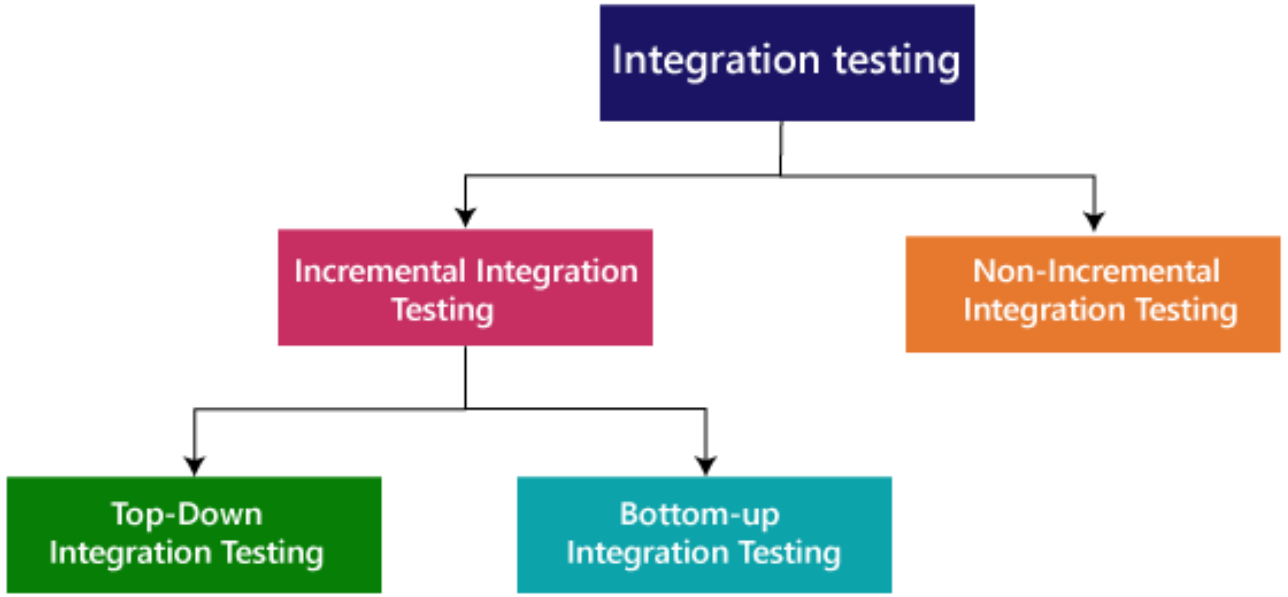


UNIT Test

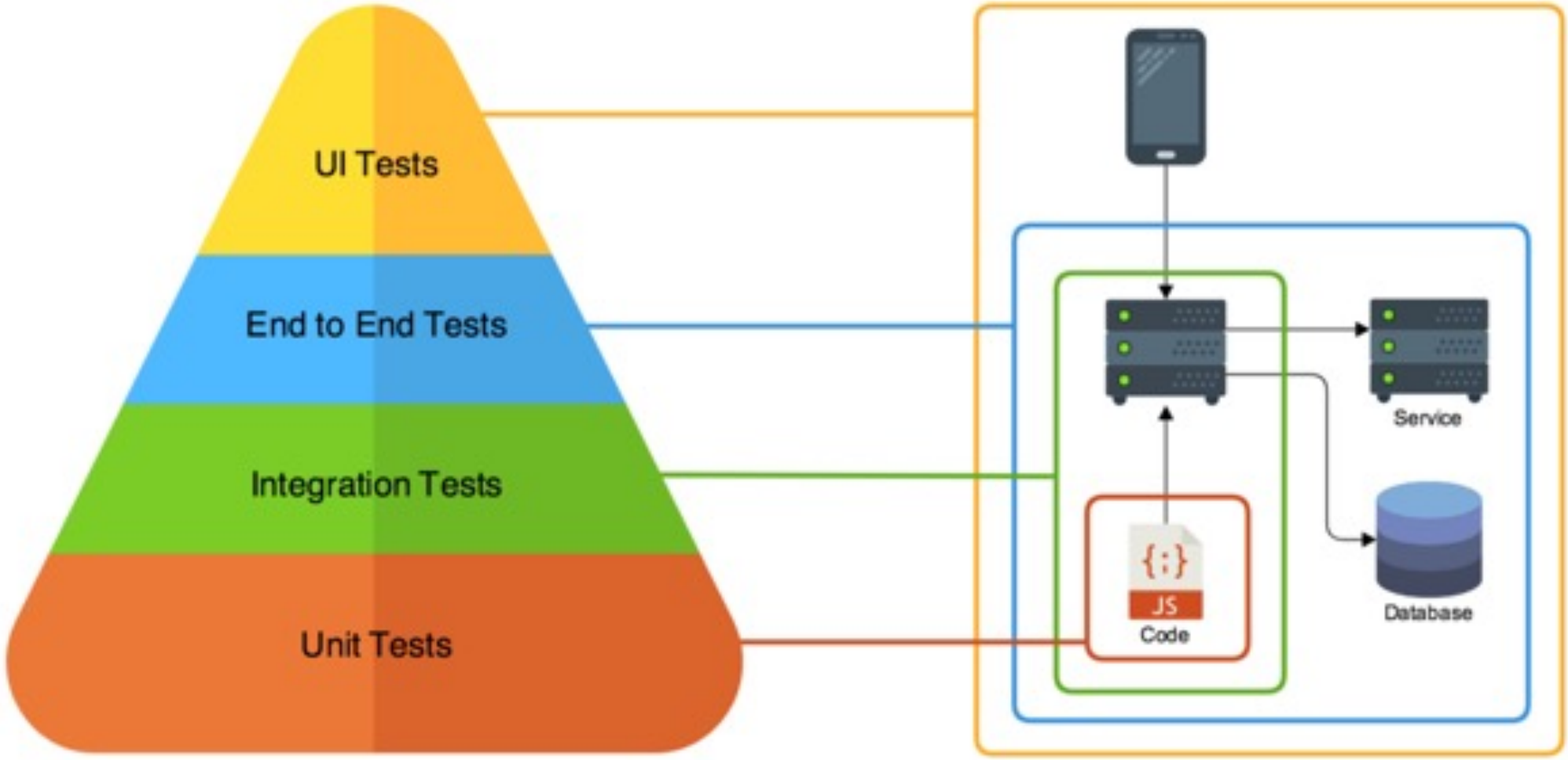

GIVEN
WHEN
THEN
TEST EXECUTION
Mocks
Constructor
Set
Property
Call
Method
Return Value
SUT
Assert
Property
Call
Mocked Dependency
Assert

Integration Test





End-to-End Testing



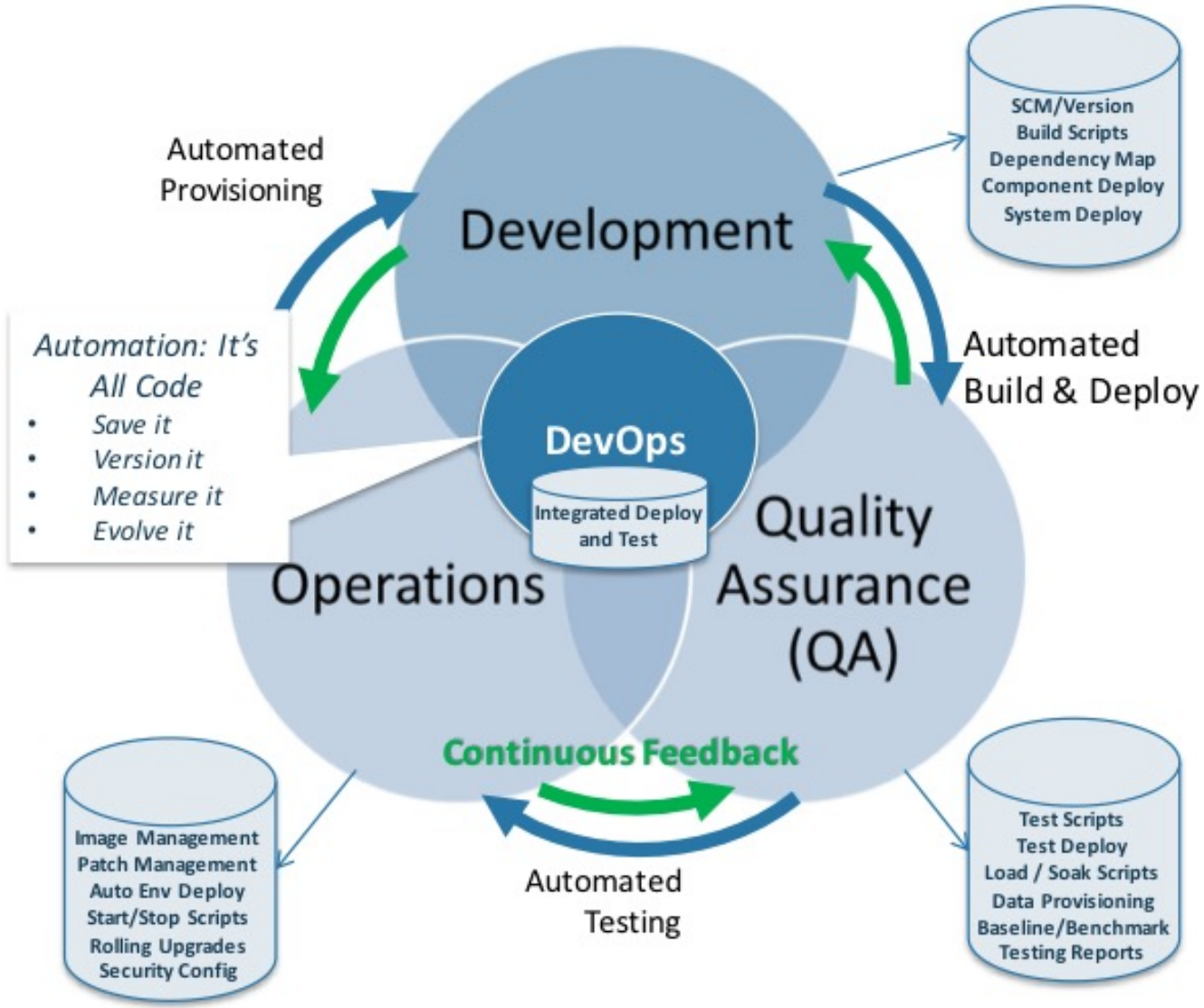
Automating test for continuous integration and deployment

Automation Testing

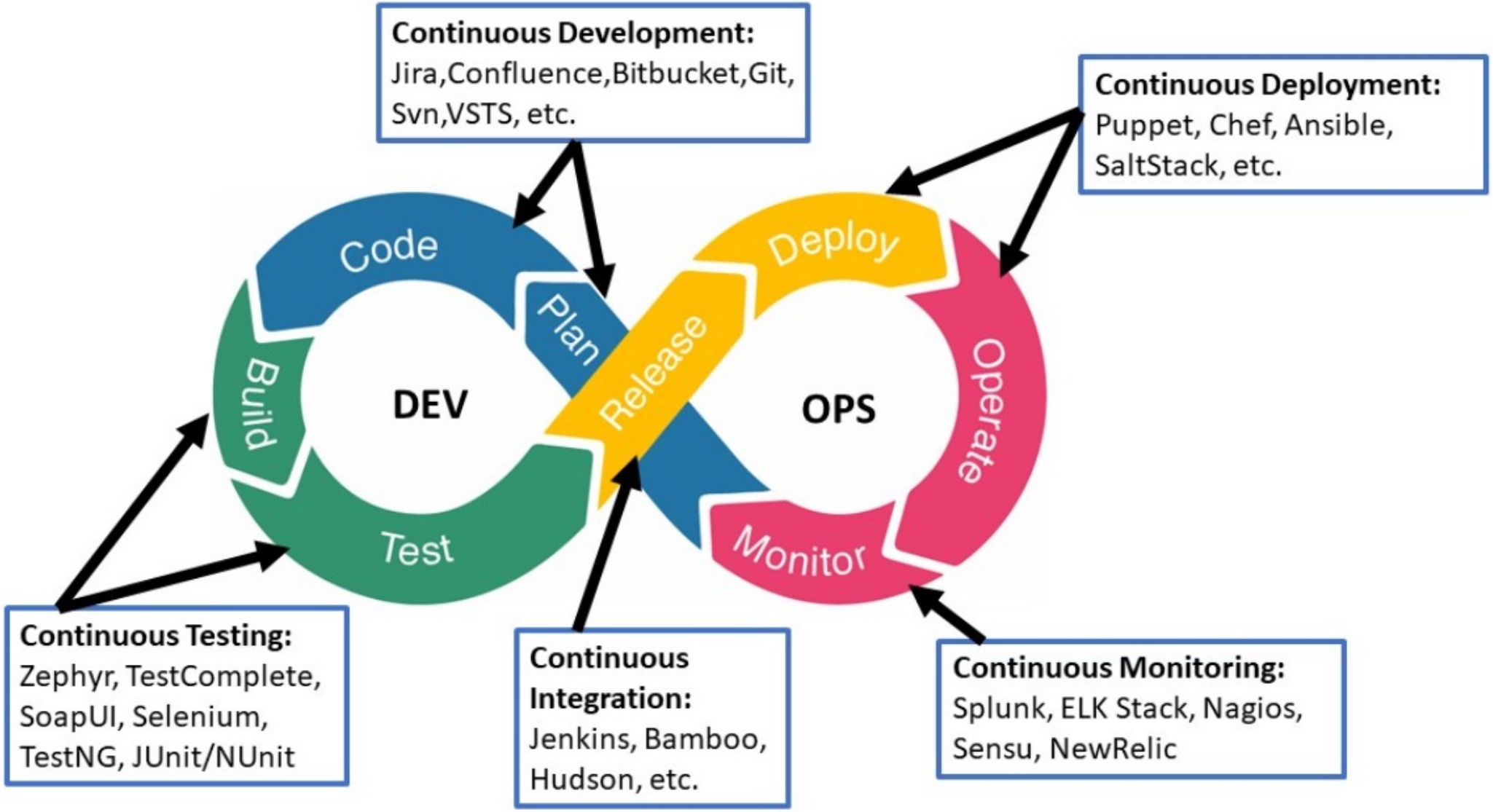
- Fast
- Reliable
- Reusable
- Improves Accuracy
- Saves time and money
- Reduces Human-generated error
- Supports the execution of repeated test cases

CI Testing Automation strategy

DevOps best practices



DevOps Tools



Choose the appropriate testing automate tool

Step 1	Understand your project needs thoroughly
Step 2	Check whether the tool will support the platform/technology on which your software/app is built
Step 3	Examine if the tool can execute cross-platform testing
Step 4	Verify if the tool will strike a balance between ease of operation and technology
Step 5	While shortlisting tools consider these vital parameters: » Licensing cost of the tool » Maintenance costs » Tool scalability » Training and support » Tool performance and stability

Set up your own cloud-based test environment

Step 1	Realizing the requirements of a test environment
Step 2	Planning, developing and provisioning of the test environment
Step 3	Embracing necessary technologies and tools with licenses
Step 4	Data masking and desensitization
Step 5	Controlling and observing the test environment proactively

Try Automate Test

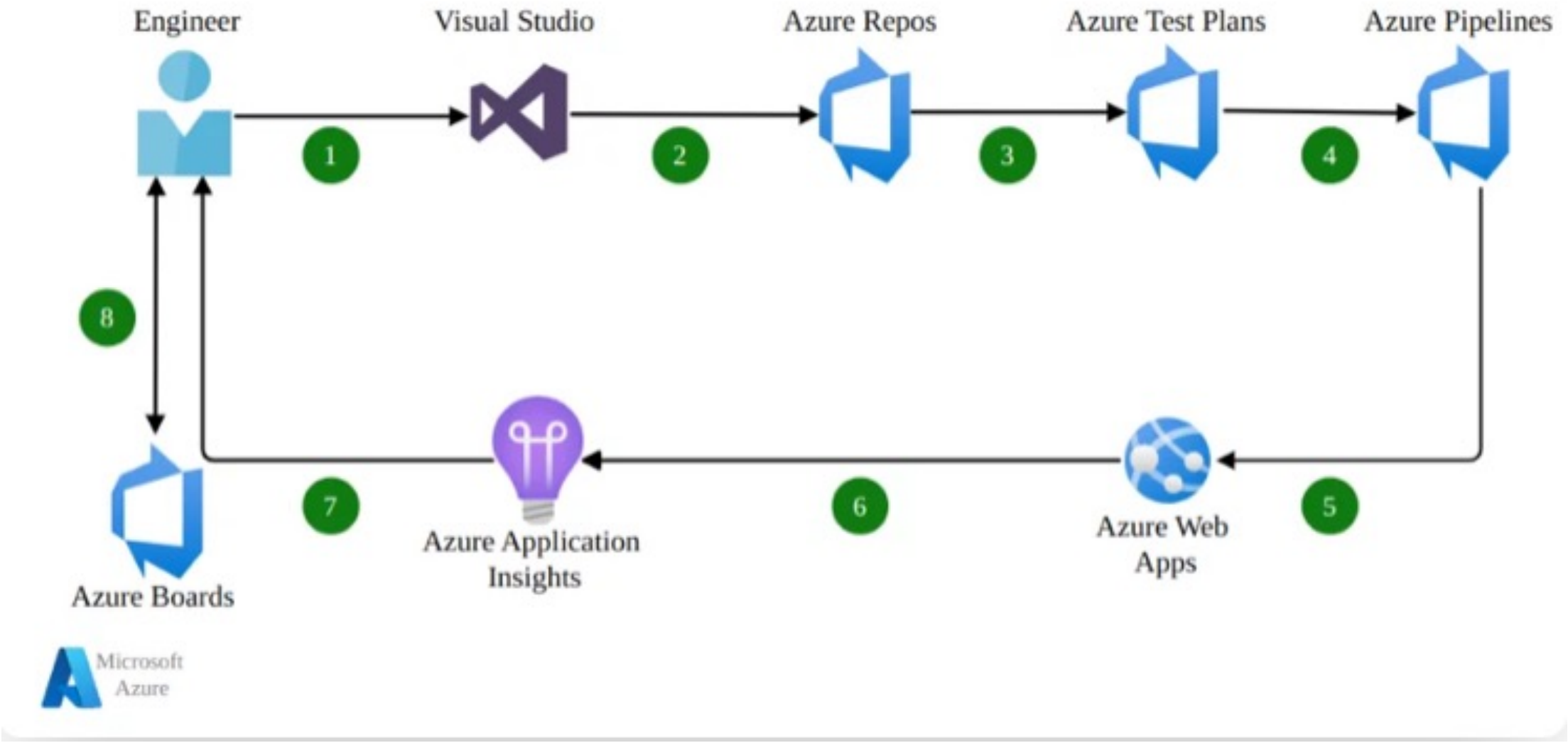
- https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-automation-testing

- https://opensource-demo.orangehrmlive.com/web/index.php/auth/login

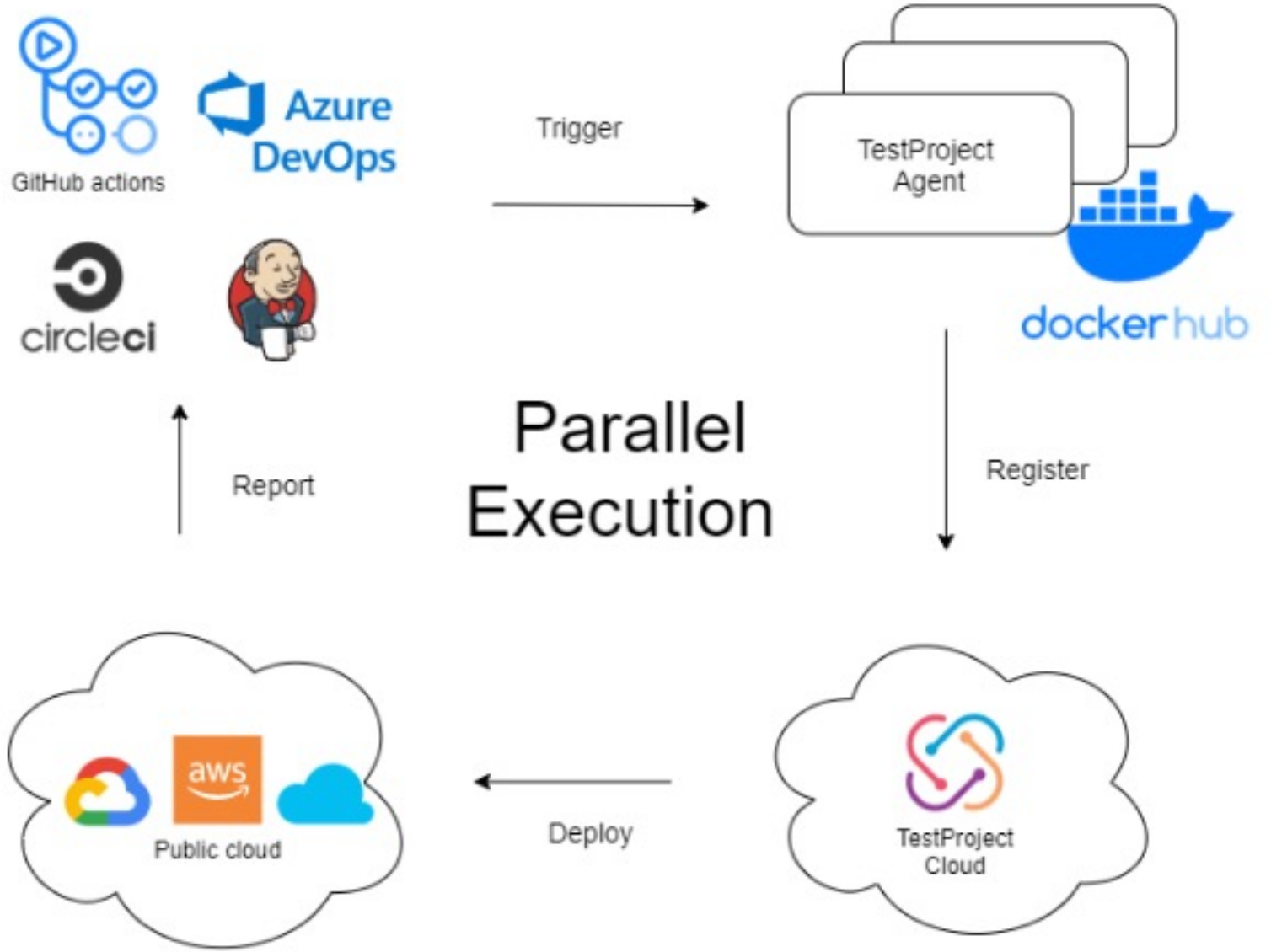
Ensuring platform reliability and performance

SRE-Site Reliability Engineering

Azure DevOps automate Testing



Test Automation Pipeline



Workshop

- Use AI to generate Test Case and Test Script
- Use Selenium Test Add-on

Thankyou