



*an initiative of RV EDUCATIONAL INSTITUTIONS*

# NEW-AGE GLOBAL UNIVERSITY FOR LIBERAL EDUCATION



*an initiative of RV EDUCATIONAL INSTITUTIONS*

# Agile Software Engineering

## CS 2004

### Unit 5

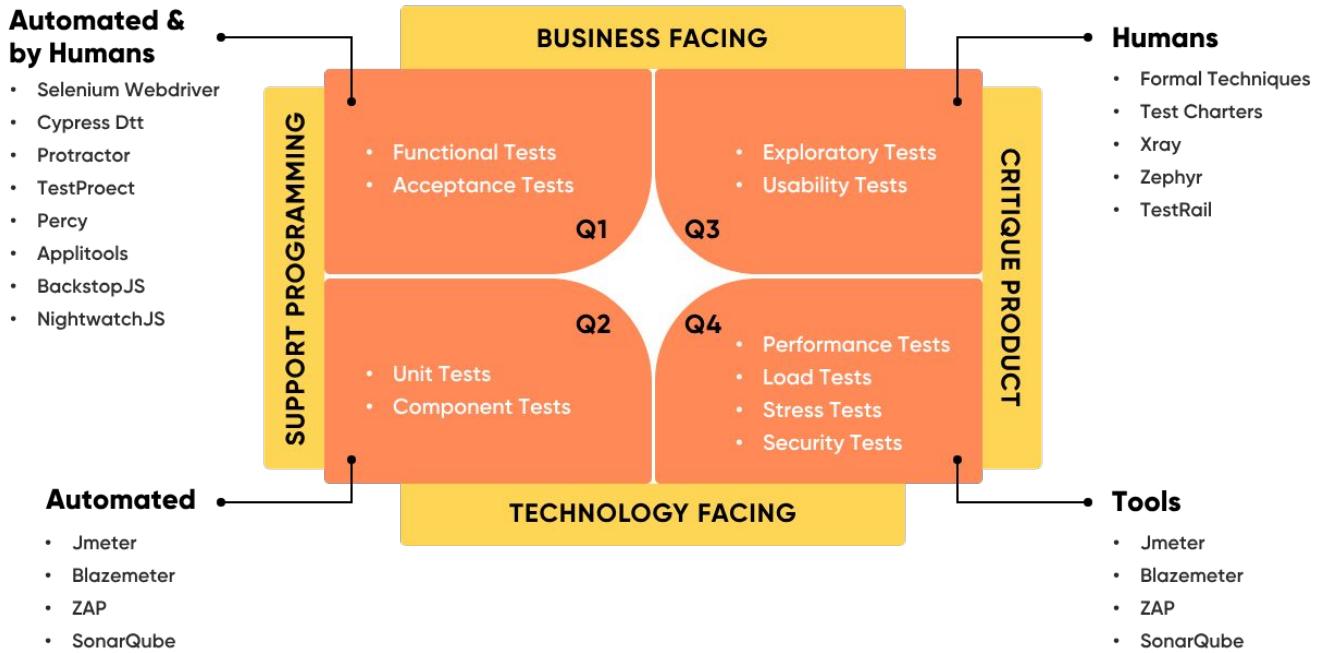
#### Agile Testing & Automation

# Agenda

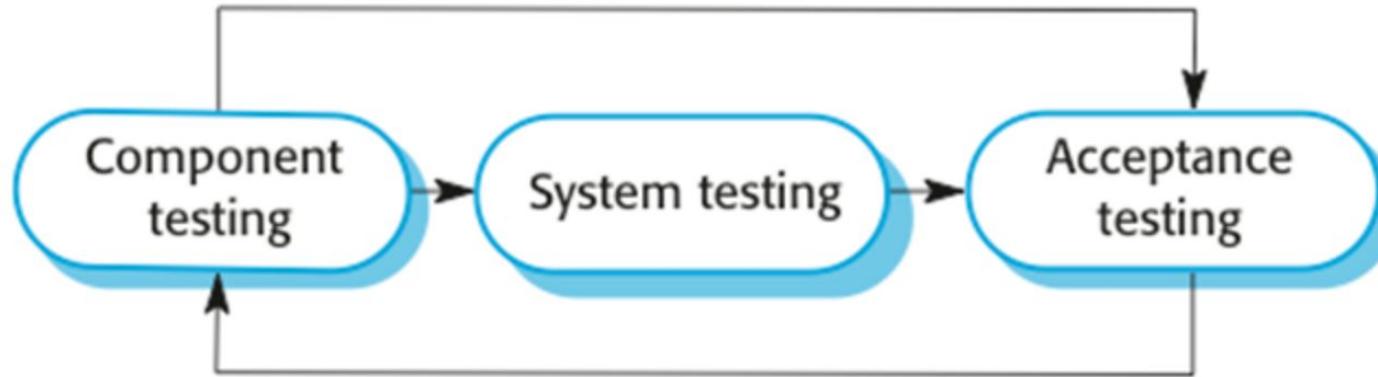


- Testing: Functionality Testing, UI Testing(Junit, PyUnit, Sonar), Performance Testing, Security Testing, A/B testing;
- Agile Testing: Principles of agile testers; The agile testing quadrants, Agile automation, Test automation pyramid;
- Test Automation Tools - Selenium, RPA(UiPath) Traceability matrix

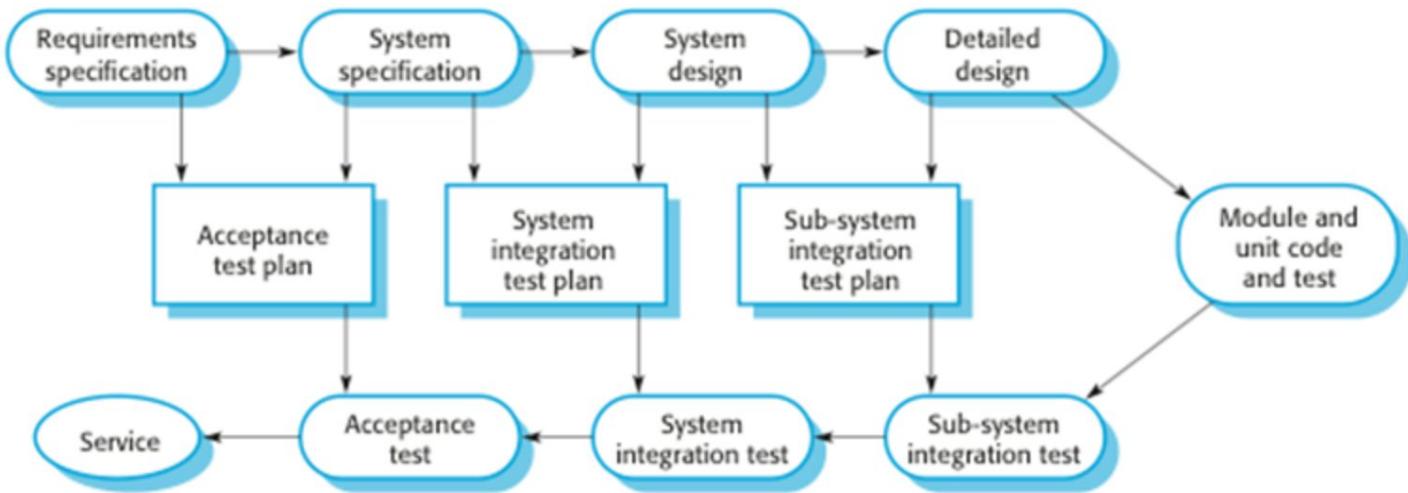
# Testing types



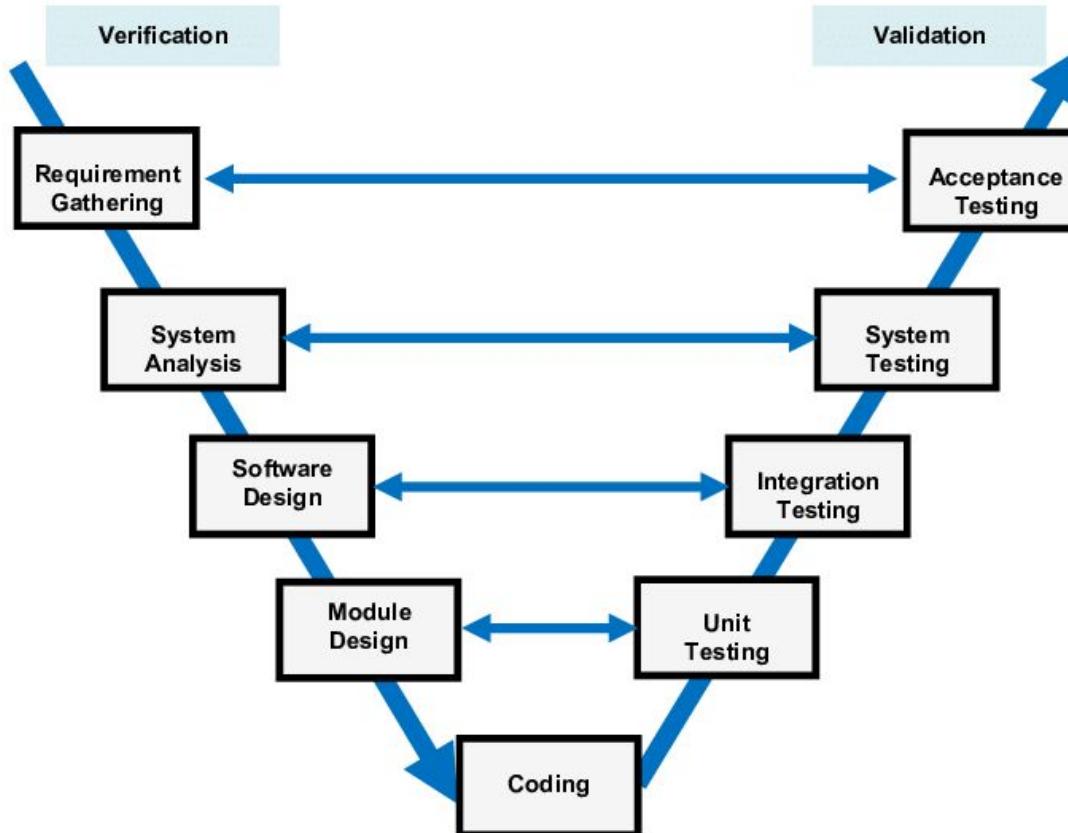
# Stages of testing



# Testing phases in a plan-driven software process



# 2 PHASES- Verification & Validation



# Testing



- Unit Testing / Component Testing
- Integration Testing
- Interface Testing
- Smoke Testing
- Sanity Testing
- Regression Testing
- System Testing
- Functional
- Non Functional Testing

# Testing (2 of 2)



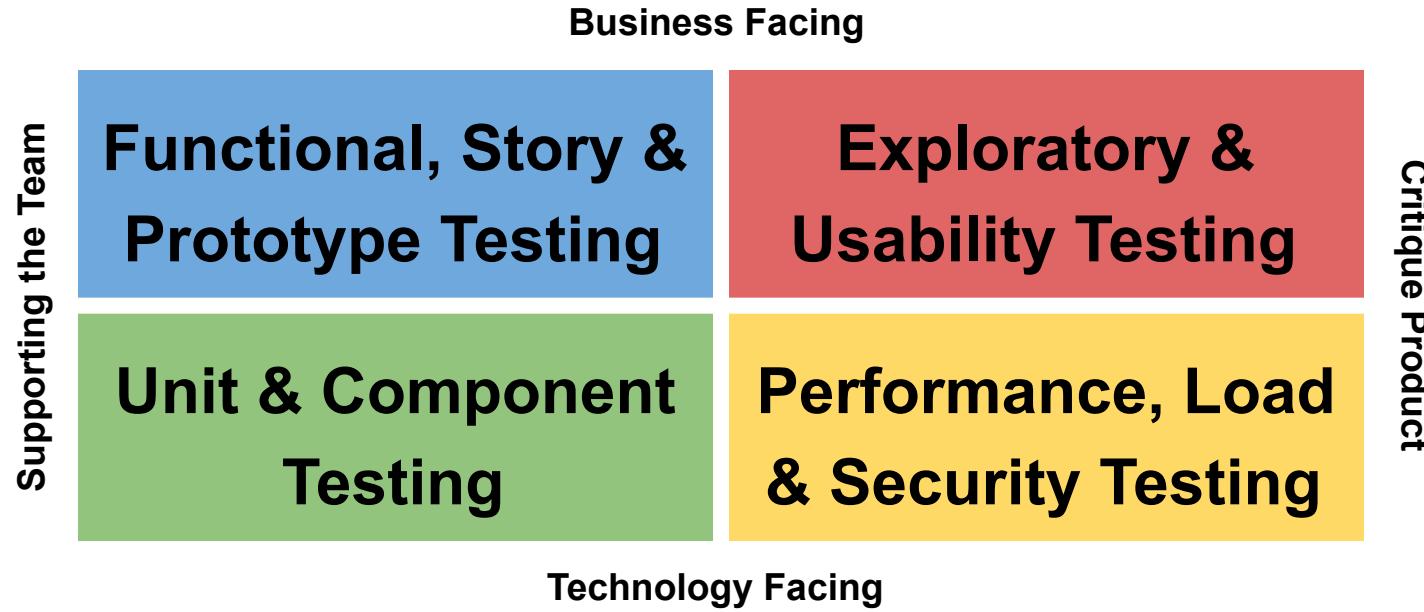
- Performance Testing
- Stress Testing
- Security Testing
- A/B Testing
- Field Testing
- Inter Operability Testing
- Exploratory Testing
- Alpha Testing
- Beta Testing
- Usability Testing
- Acceptance Testing

# Agile Testing

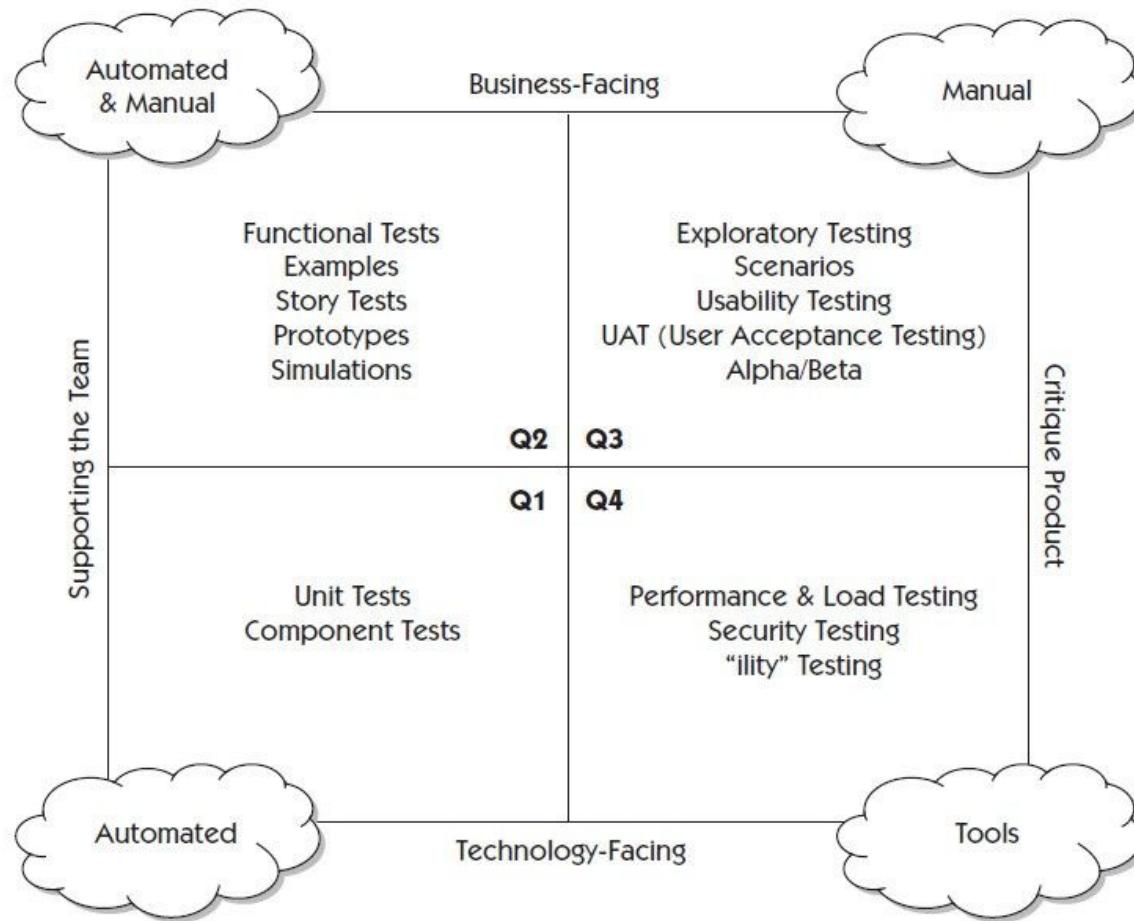
- Agile testing is a software testing practice that follows the principles of Agile development.
- It emphasises
  - collaboration,
  - flexibility, and
  - customer satisfaction,
- focusing on delivering high-quality software products.



# Agile Testing Quadrant



# Agile Testing Quadrants



## Q2

With a combination of manual and automated tests, these 'business-facing' tests are more customer-focused, but they support your application build as well. These include functional tests, which ensure the product does what it is meant to do.

## Q1

Unit and component tests are performed throughout your application's development. They provide feedback to your developers on the quality of their code on an ongoing basis, usually through repeated, automated processes.

## Q3

User Acceptance Tests (UAT), usability and exploratory testing all belong in this quadrant. These involve manual testing by experienced QA engineers and end-user testing. Your aim here is to gain feedback and improve the quality of the product, ensuring it is fit for the designed purpose.

## Q4

These are technology-facing performance tests, like load testing and checking the data security of your software application. There are many tools available to automate this type of testing, such as Selenium used alongside JMeter.

# Unit Testing



- Unit Testing involves testing individual units or components of a software application in isolation to verify that each unit functions as expected.
- A "unit" is typically the smallest testable part of the application, such as a function, method, or class.
- Unit tests are usually automated and written by developers as part of the development process.

# Why should you do Unit testing

## ?



- 1. Early Detection of Bugs:** Unit tests help catch bugs and issues early in the development cycle, reducing the cost and effort required to fix them later.
- 2. Continuous Integration and Delivery:** Unit tests are integral to Continuous Integration (CI) and Continuous Delivery (CD) pipelines. They ensure that new changes do not break existing functionality.
- 3. Refactoring with Confidence:** Unit tests provide a safety net for developers when refactoring code, ensuring that changes do not introduce new bugs.
- 4. Documentation:** Unit tests serve as documentation for the code, providing examples of how different units are expected to behave.
- 5. Improved Code Quality:** Writing unit tests encourages developers to write modular, testable, and maintainable code.

# Integration Testing



- Integration Testing is a crucial practice that focuses on testing the interactions and interfaces between different components or systems to ensure that they work together as expected.
- Integration Testing involves combining individual units or components of an application and testing them as a group.
- The goal is to identify any issues that arise from the interaction between these integrated components.
- This type of testing is essential to verify that the different parts of the system work together correctly.

# Importance of Integration Testing



- 1. Early Detection of Integration Issues:** Integration testing helps catch issues that occur when different components interact, reducing the risk of discovering these problems later in the development cycle.
- 2. Continuous Integration:** In Agile/Scrum, continuous integration practices involve frequently merging code changes into a shared repository. Integration tests are run as part of the CI pipeline to ensure that newly integrated code does not break existing functionality.
- 3. Improved Collaboration:** Integration testing fosters collaboration among development teams, as it requires coordination and communication to ensure that components developed by different team members work seamlessly together.
- 4. Increased System Reliability:** By testing interactions between components, integration testing enhances the overall reliability and stability of the system.

# System Testing

- System Testing involves testing the entire application or system as a single entity to validate its end-to-end functionality.
- It ensures that the integrated components of the system interact correctly and that the system meets the specified requirements, including functional and non-functional aspects.

# Importance of System Testing



- 1. Validation of Requirements:** System Testing verifies that the system meets the specified requirements and behaves as expected, ensuring that the software fulfills the needs of the stakeholders.
- 2. End-to-End Testing:** It ensures that all integrated components of the system work together as intended and that the entire application functions correctly from start to finish.
- 3. Detection of Defects:** System Testing helps identify defects and issues that may not have been detected during earlier testing phases (e.g., unit testing or integration testing).
- 4. User Experience:** It validates the user experience by testing the system from the end-user's perspective, ensuring that the application is intuitive and meets user expectations.
- 5. Quality Assurance:** System Testing contributes to the overall quality assurance process by ensuring that the system is reliable, stable, and ready for deployment.

# Agile Quality

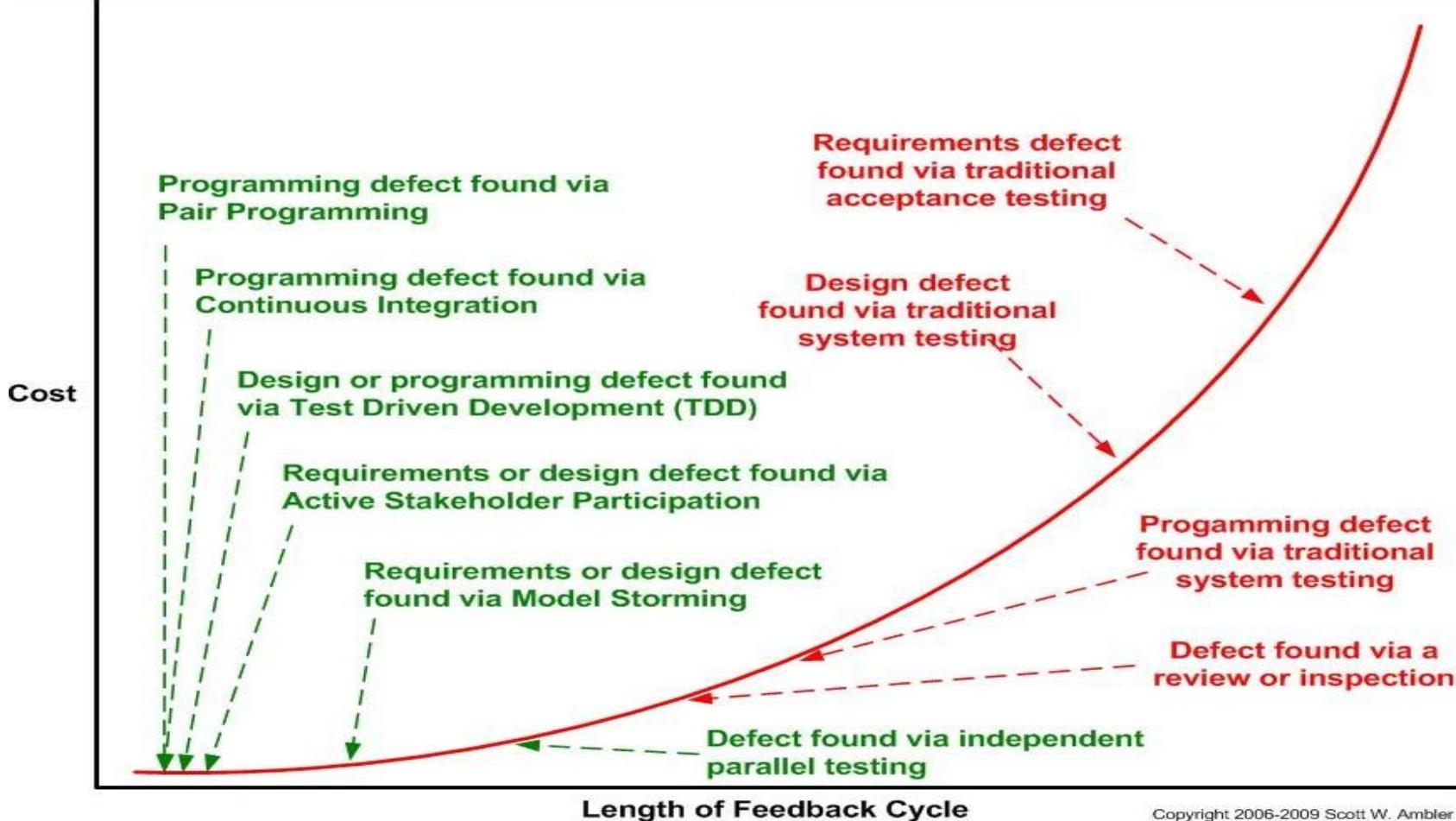


## QA is no longer the Sole Quality Gatekeeper!

- QA may play new role
- Agile Quality requires new skills
- **Focuses on Prevention instead of Detection**
- Testing role may be done by others
- Testing done upfront
- Automation plays a bigger role
- Focuses on xUnit testing

**QUALITY**  
**IT'S**  
**EVERYONE'S**  
**RESPONSIBILITY**

# Up front testing leads to lower costs and better quality



# The whole team is responsible for quality



Idea:  
Testing in collaboration

## Business

- Product Owners / Product Manager
- Subject Matter Experts

## Technology

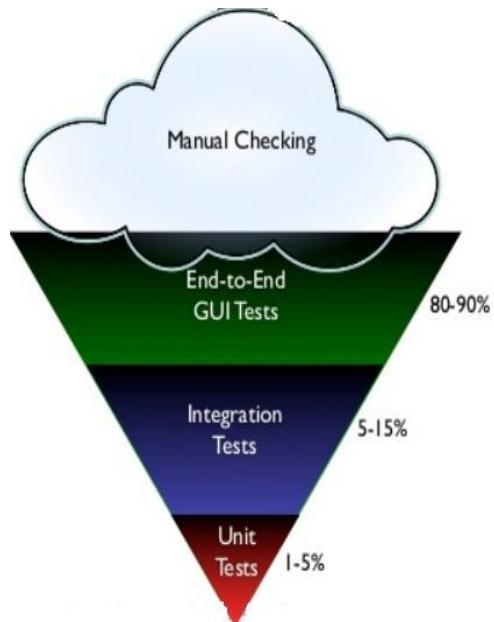
- Architects
- Database Administrators
- User Experience Designers
- Operations/Support team members

## Team

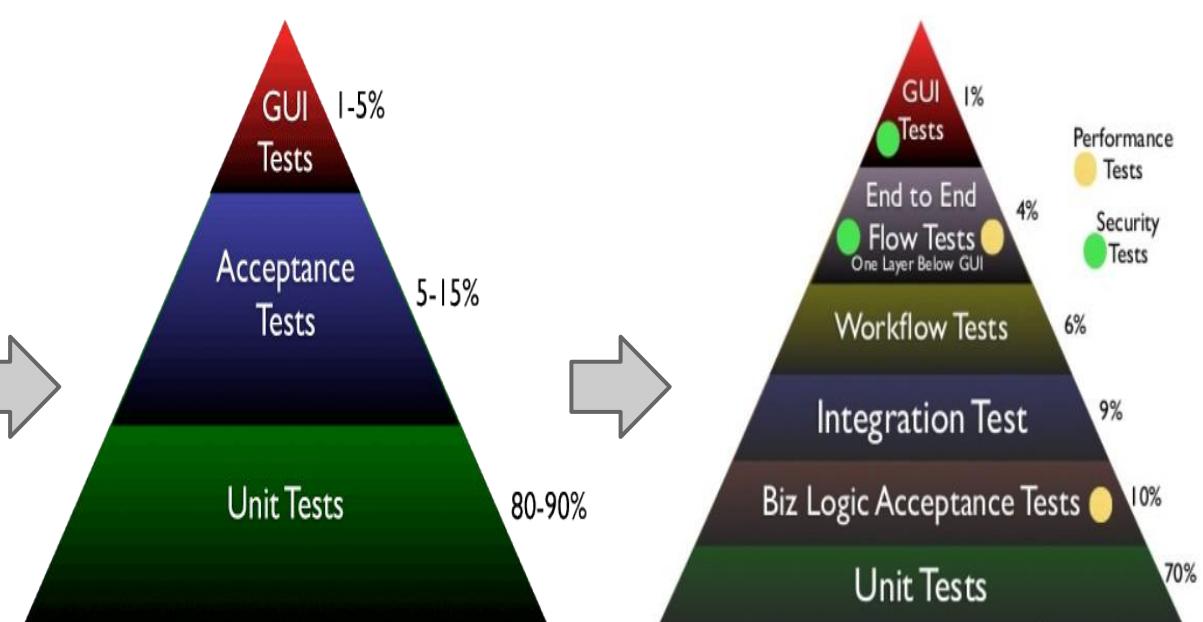
- Developers
- Testers
- Business Analysts

# Agile Test Automation Pyramid

**Traditional**

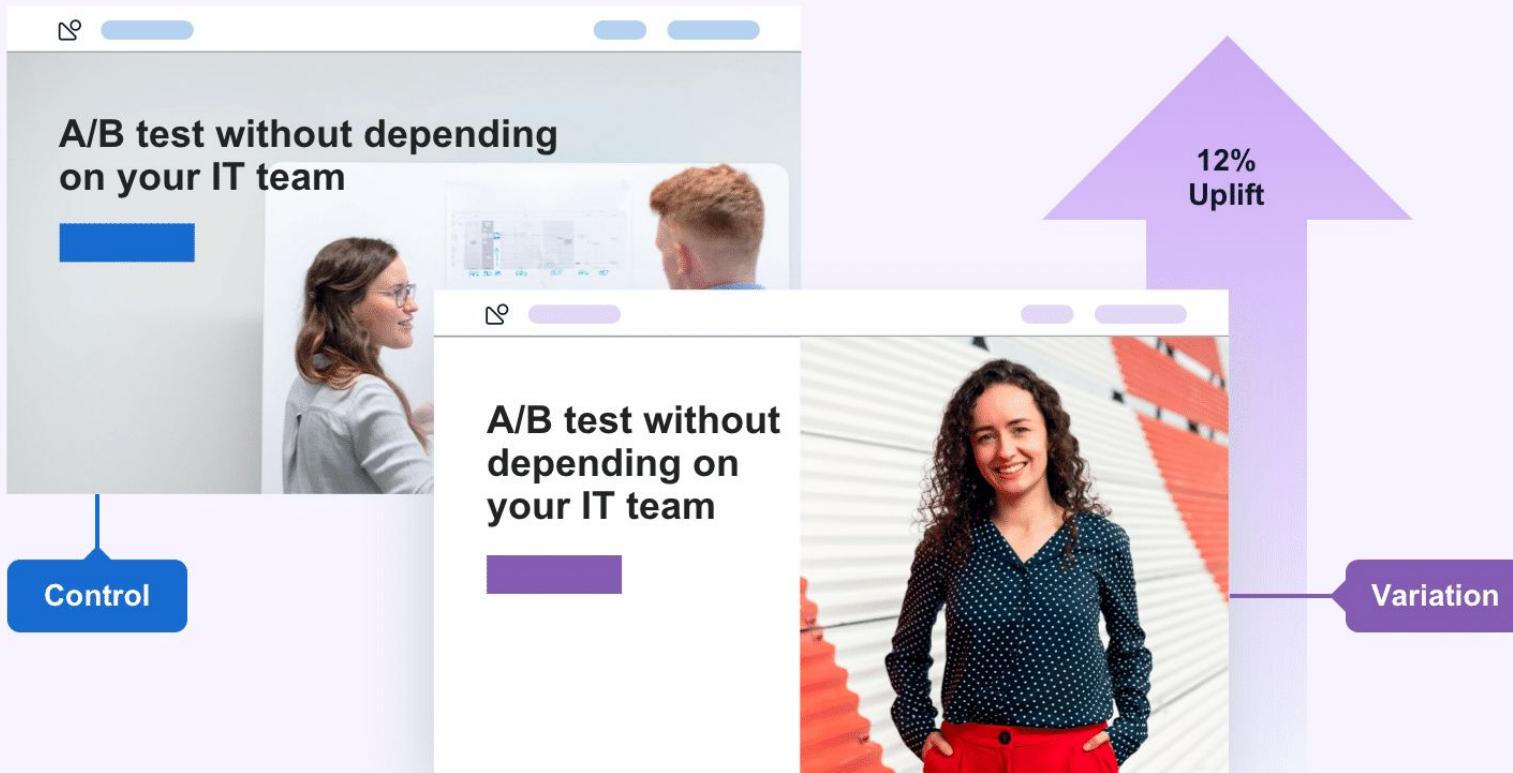


**Agile**



Source: <http://www.slideshare.net/nashjain/inverting-the-testing-pyramid>

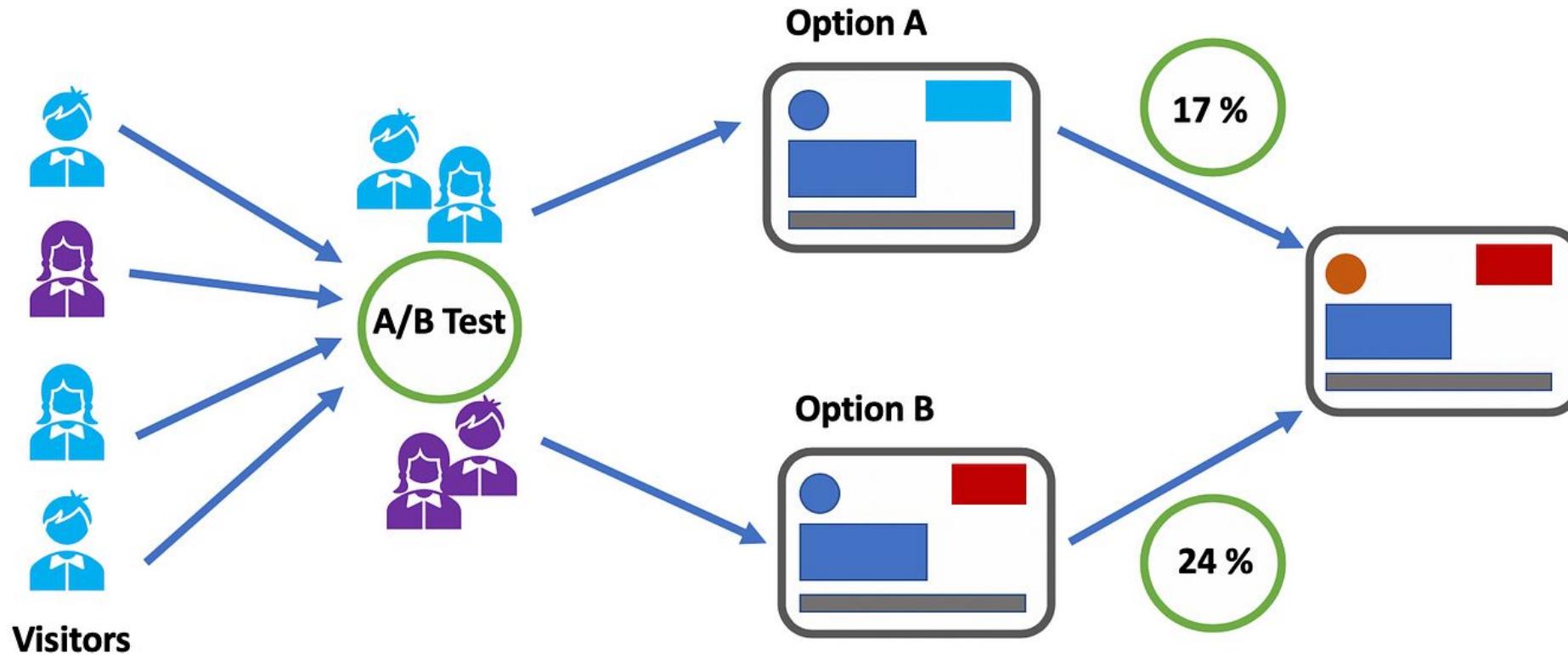
# What is A/B Testing



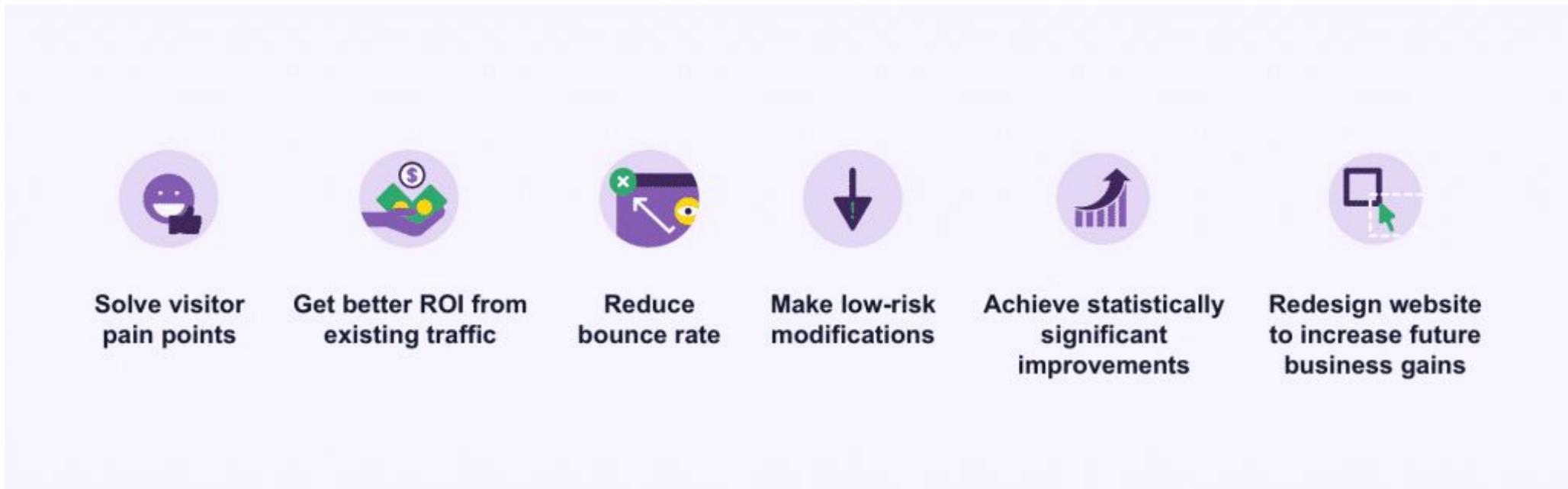
# A/B Testing

- A/B testing, also known as split testing, is a user experience research methodology. It consists of a randomized experiment that usually involves two variants (A and B), although the concept can be extended to multiple variants of the same variable
- In A/B testing, A refers to the ‘control’ or the original testing variable, whereas B refers to the ‘variation’ or a new version of the original testing variable<sup>2</sup>. The version that moves your business metric(s) in the positive direction is known as the ‘winner’
- Implementing the changes of this winning variation on your tested page(s) / element(s) can help optimize your website and increase business ROI

# A/B Testing –contd.



# Why A/B Testing



# Quiz on Testing

## 1 - What is regression testing?

- a. Testing to ensure that new changes do not break existing functionality
- b. Testing to ensure that the software meets the requirements
- c. Testing to ensure that the software is scalable
- d. Testing to ensure that the software is secure

## 2- What is black box testing?

- a. Testing based on the internal structure of the code
- b. Testing based on the external behavior of the code
- c. Testing based on the color of the code
- d. None of the above

### 3- What is white box testing?

- a. Testing based on the internal structure of the code
- b. Testing based on the external behavior of the code
- c. Testing based on the color of the code
- d. None of the above

### 4- What is acceptance testing?

- a. Testing to ensure that new changes do not break existing functionality
- b. Testing to ensure that the software meets the requirements of the customer or user
- c. Testing to ensure that the software is scalable
- d. Testing to ensure that the software is secure

## 5- What is performance testing?

- a. Testing to ensure that new changes do not break existing functionality
- b. Testing to ensure that the software meets the requirements of the customer or user
- c. Testing to measure how well the software performs under load or stress conditions
- d. Testing to ensure that the software is secure

## 6- What is security testing?

- a. Testing to ensure that new changes do not break existing functionality
- b. Testing to ensure that the software meets the requirements of the customer or user
- c. Testing to measure how well the software performs under load or stress conditions
- d. Testing to identify and address potential vulnerabilities in the software

## 7- What is functional testing?

- a. Testing to verify whether each function of an application works as per its requirement
- b. Testing to verify whether each function of an application works as per its design
- c. Testing to verify whether each function of an application works as per its code
- d. None of these

## 8- What is non-functional testing?

- a. Testing which verifies whether each function of an application works as per its requirement
- b. Testing which verifies whether each function of an application works as per its design
- c. Testing which verifies whether each function of an application works as per its code
- d. Testing which verifies attributes like performance, usability, reliability, etc.

### 9 - What does a Traceability Matrix correlate and trace?

- a) Business, application, security or any other requirements to their implementation, testing or completion
- b) Different versions of a software
- c) Bugs in the software to their solutions
- d) Software requirements to their sources

### 10- What is Requirement Traceability Matrix (RTM)?

- a) A document that maps and traces user requirement with test cases
- b) A type of software testing
- c) A method used in Agile development
- d) A document that lists all the requirements of a software

### 11- What is the main purpose of A/B testing?

- a) To optimize a website or app's performance
- b) To find bugs in the software
- c) To check the compatibility of the software with different systems
- d) To ensure the software meets all the requirements

### 12- In A/B testing, what does 'A' refer to?

- a) The new version of the original testing variable
- b) The control or the original testing variable
- c) The result of the test
- d) The process of testing

## Principles of Agile Testing





Automation in  
Agile

# Agile Automation



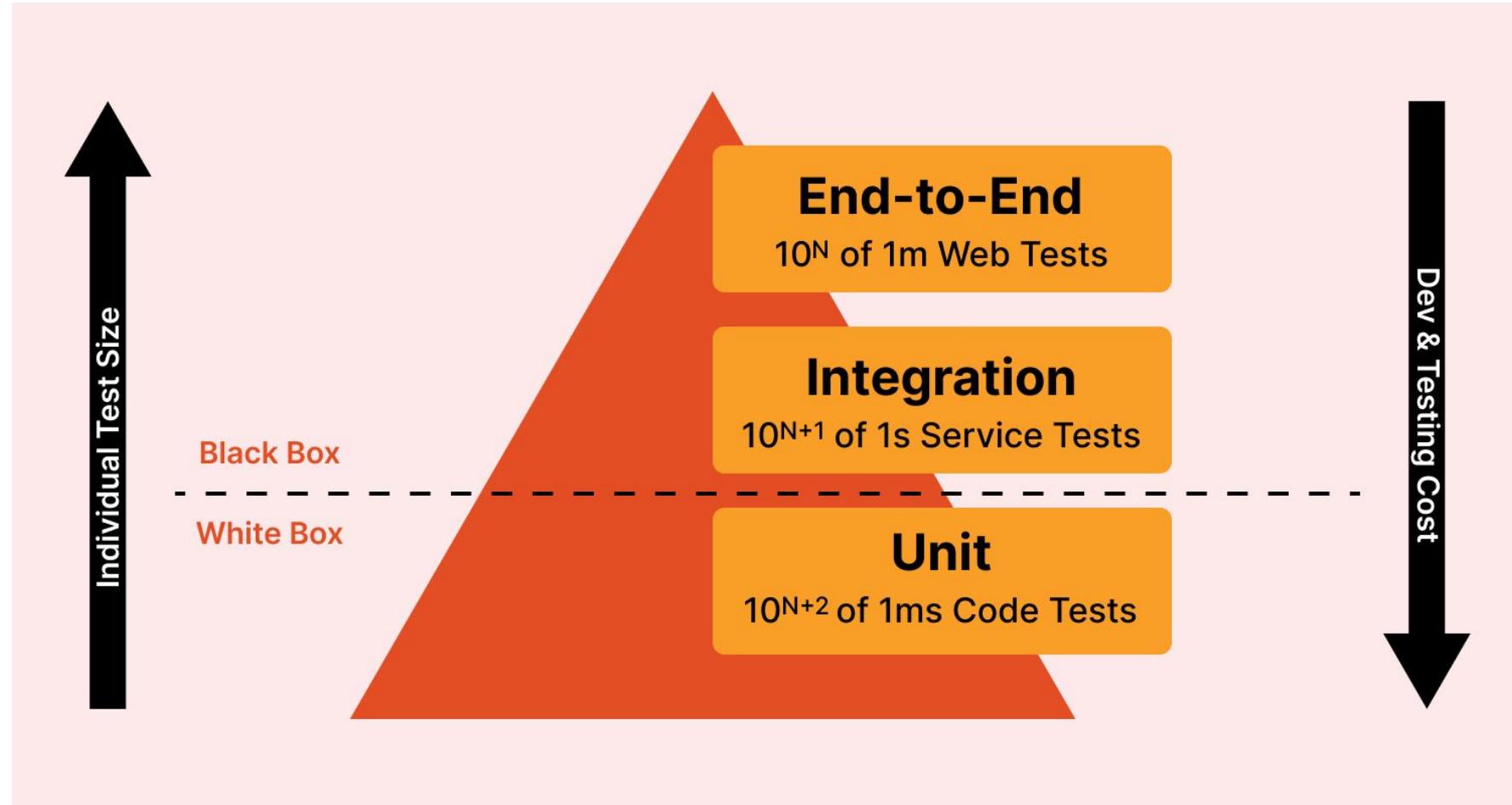
- Agile Automation Testing in software development is an approach of using test automation in agile methodologies
- The purpose of agile automation testing is to make the software development process more effective and efficient while maintaining the quality and time as well as resource consumption<sup>1</sup>
- It involves an upfront effort that fully defines the process before development work begins in earnest
- This work ensures that the automation project will integrate with the wider business and comply with regulatory requirements and other constraints
- An automated build verifies changes. As a result, there's a reduction in integration debt and a continually shippable main branch
- These practices, like all Agile practices, carry the Agile label, because they're consistent with the principles in the Agile manifesto

# Test Automation Pyramid



- Test automation pyramid consists of three levels:
  - Unit Tests
  - Integration Tests
  - End to end Tests

# Test Automation Pyramid



# Test Automation Tools: Selenium



# What Selenium can do ?



- 1. Automate Web Applications for Testing:** Selenium is widely used for automating web application testing, allowing testers to write scripts in various programming languages to simulate user interactions and verify application behavior.
- 2. Cross-Browser Testing:** Selenium supports multiple browsers like Chrome, Firefox, Safari, and Edge, enabling testers to ensure their web applications work consistently across different browsers.
- 3. Data Extraction and Web Scraping:** Selenium can be used to automate the extraction of data from websites, making it useful for web scraping tasks where data needs to be collected from various web pages.

# What Selenium can do –contd.

?



- 4. Integration with CI/CD Tools:** Selenium can be integrated with Continuous Integration/Continuous Deployment (CI/CD) tools like Jenkins, enabling automated testing as part of the software development pipeline.
- 5. Simulate Complex User Interactions:** Selenium can simulate complex user interactions such as mouse movements, clicks, form submissions, and keyboard inputs, making it possible to test intricate user workflows.

# Test Automation Tools: UiPath- RPA tool

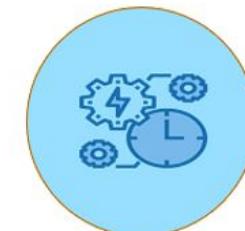
UiPath is a versatile Robotic Process Automation (RPA) tool that can automate a wide range of tasks



Drag and drop workflow



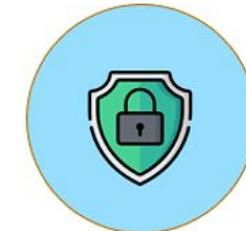
Record and playback



Inbuilt activities in UiPath



Advanced Scrapping options



High security and robustness

# What UiPath can do ?



- 1. Automate Repetitive Tasks:** UiPath can automate repetitive tasks such as data entry, form filling, and file transfers, freeing up human workers to focus on more complex activities
- 2. Web Scraping and Data Extraction:** UiPath can extract data from websites, PDFs, and other sources, making it useful for tasks like web scraping and data aggregation

# What UiPath can do ? – contd.



- 3. Email Automation:** UiPath can automate email-related tasks, such as sending, receiving, and processing emails, which is particularly useful for customer service and marketing
- 4. Excel Automation:** UiPath can automate Excel operations, including data manipulation, report generation, and complex calculations, enhancing productivity in data-intensive tasks
- 5. Application Integration:** UiPath can integrate with various applications and systems, allowing seamless automation across different platforms and enhancing workflow efficiency

# Traceability Matrix

- A **Traceability Matrix** is a document used in software development and testing to ensure that all requirements are covered by test cases.
- It maps and traces user requirements with test cases, helping to validate that the software product meets all specified requirements.

# Traceability Matrix – key points

- **Purpose:** It ensures that all requirements are tested and helps identify any missing functionality.
- **Structure:** Typically, it is a table that correlates requirements with their corresponding test cases.
- **Usage:** It is used throughout the software development lifecycle to track the status of requirements and their verification.

A Requirements Traceability Matrix (RTM) might include columns for requirement IDs, descriptions, test case IDs, and the status of each test case

# Activity 1 (Submit it on classroom)



- Write 2-5 test cases each testing category for a mobile phone

- 1) Sanity Testing
- 2) Security Testing
- 3) Stress testing
- 4) Performance Testing

# Activity 2 – Tests for Electric Car (Submit it on classroom)



- Write 2-5 test cases for each testing category for an EV 4 wheeler
- 1) Functional Testing
  - 2) Stress testing
  - 3) Performance Testing
  - 4) A/B Tests

# Performance Testing

# Performance Testing- Definition

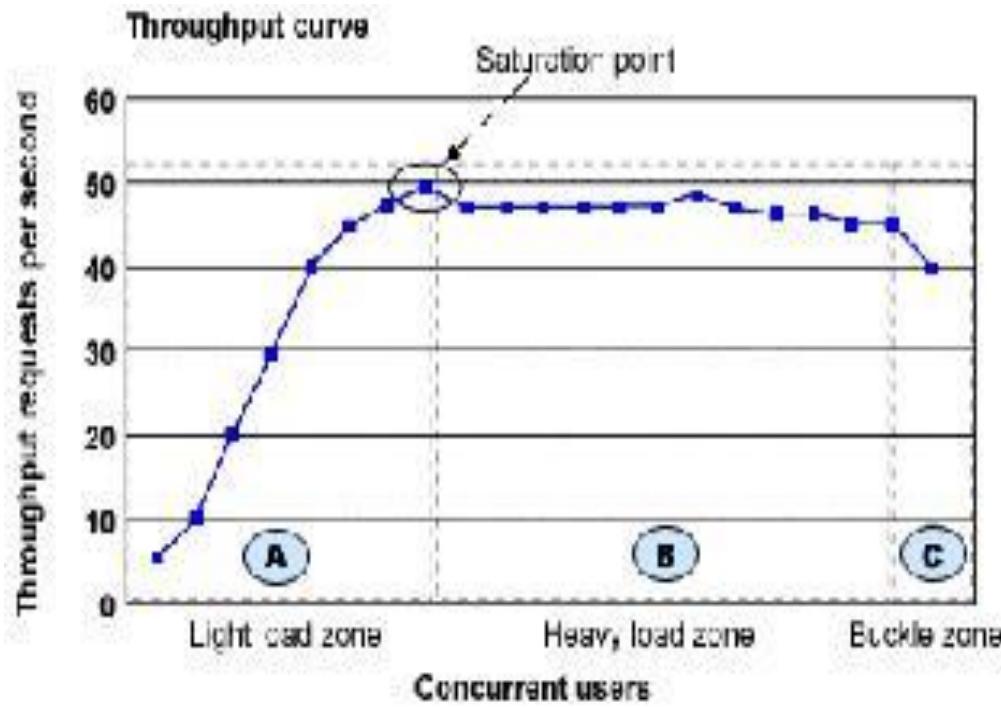


- The testing to evaluate the response time (speed), throughput and utilization of system to execute its required functions in comparison with different versions of the same product or a different competitive product is called Performance Testing.
- Performance testing is done to derive benchmark numbers for the system.
- Heavy load is not applied to the system
- Tuning is performed until the system under test achieves the expected levels of performance.

# Performance Testing

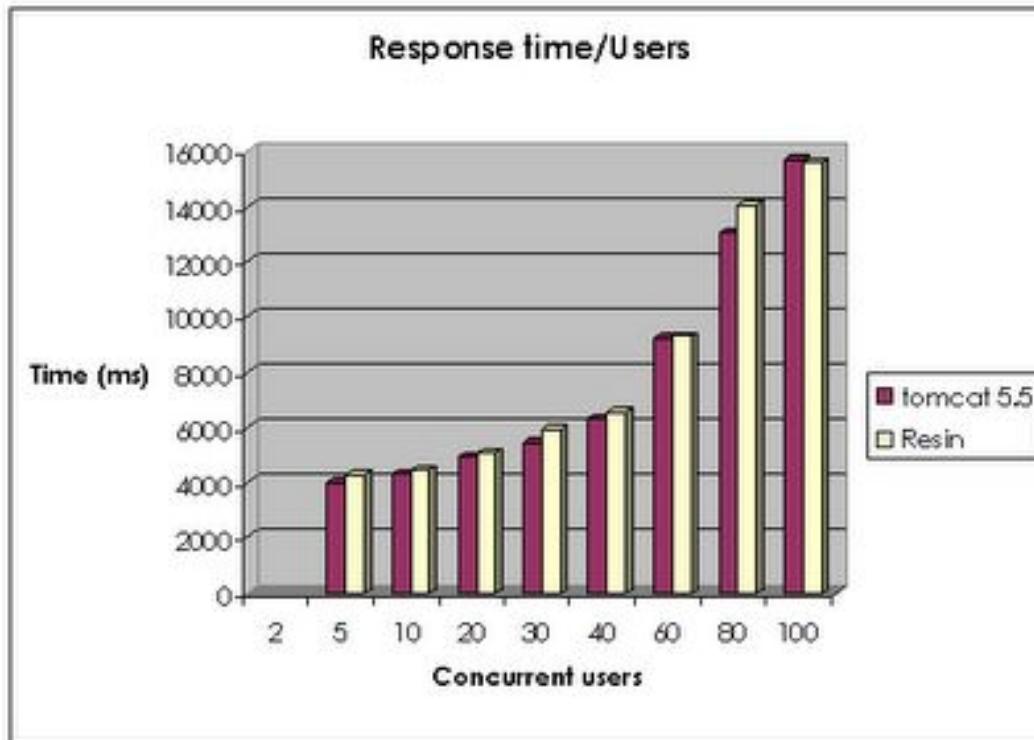
- Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device.
- Before going into the details, we should understand the factors that governs Performance testing:
  - ✓ Throughput
  - ✓ Response Time
  - ✓ Tuning
  - ✓ Benchmarking

# Throughput



- Capability of a product to handle multiple transactions in a given period.
- Throughput represents the number of requests/business transactions processed by the product in a specified time duration.
- As the number of concurrent users increase, the throughput increases almost linearly with the number of requests. As there is very little congestion within the Application Server system queues.

# Response Time

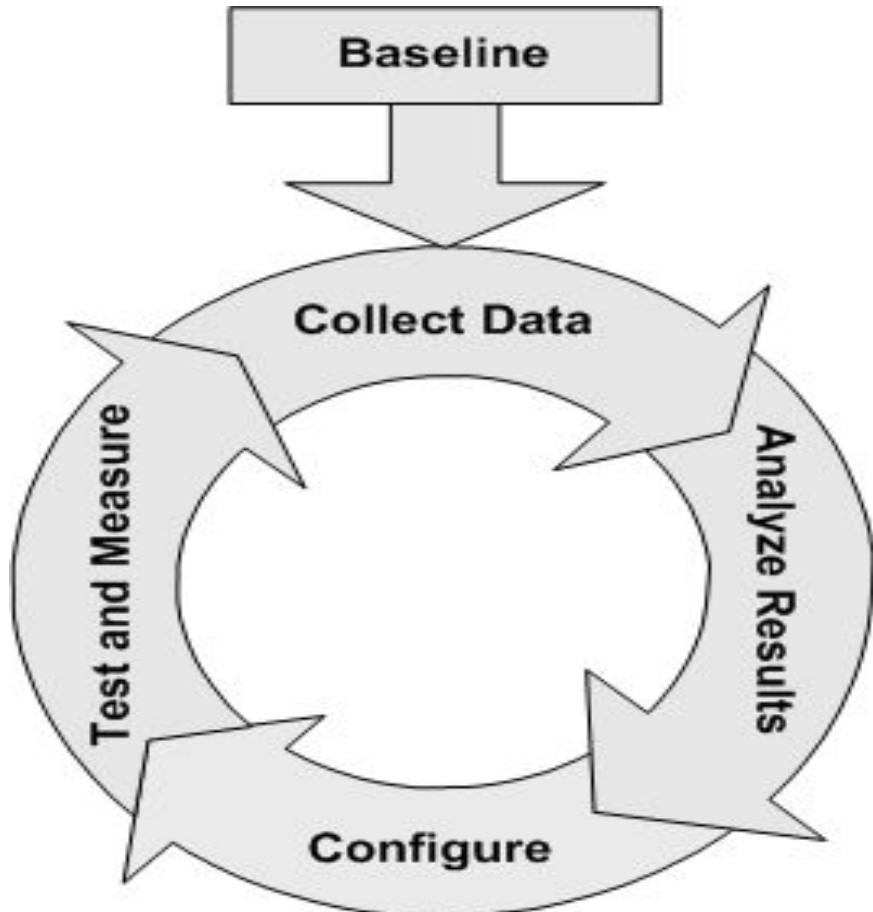


- It is equally important to find out how much time each of the transactions took to complete.

Response time is defined as the delay between the point of request and the first response from the product.

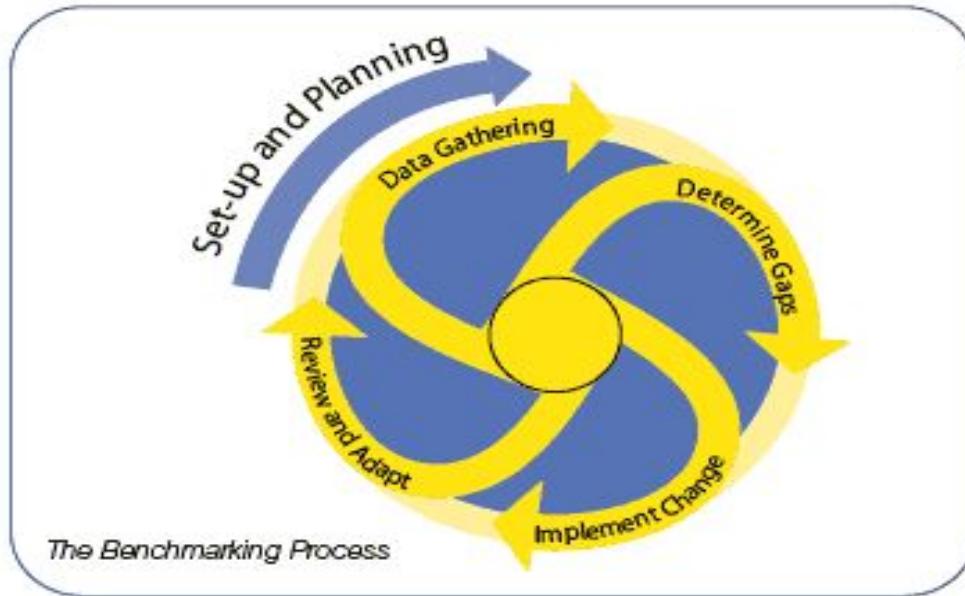
The response time increases proportionally to the user load.

# Tuning



- Tuning is the procedure by which product performance is enhanced by setting different values to the parameters of the product, operating system and other components.
- Tuning improves the product performance without having to touch the source code of the product.

# Benchmarking



- A very well-improved performance of a product makes no business sense if that performance does not match up to the competitive products.

A careful analysis is needed to chalk out the list of transactions to be compared across products so that an apple-apple comparison becomes possible.

# Difference between Performance, Load and Stress Testing



## Load Testing

- Process of exercising the system under test by feeding it the largest tasks it can operate with.
- Constantly increasing the load on the system via automated tools to simulate real time scenario with virtual users.

## Examples:

- Testing a word processor by editing a very large document.
- For Web Application load is defined in terms of concurrent users or HTTP connections.

# Difference between Performance, Load and Stress Testing



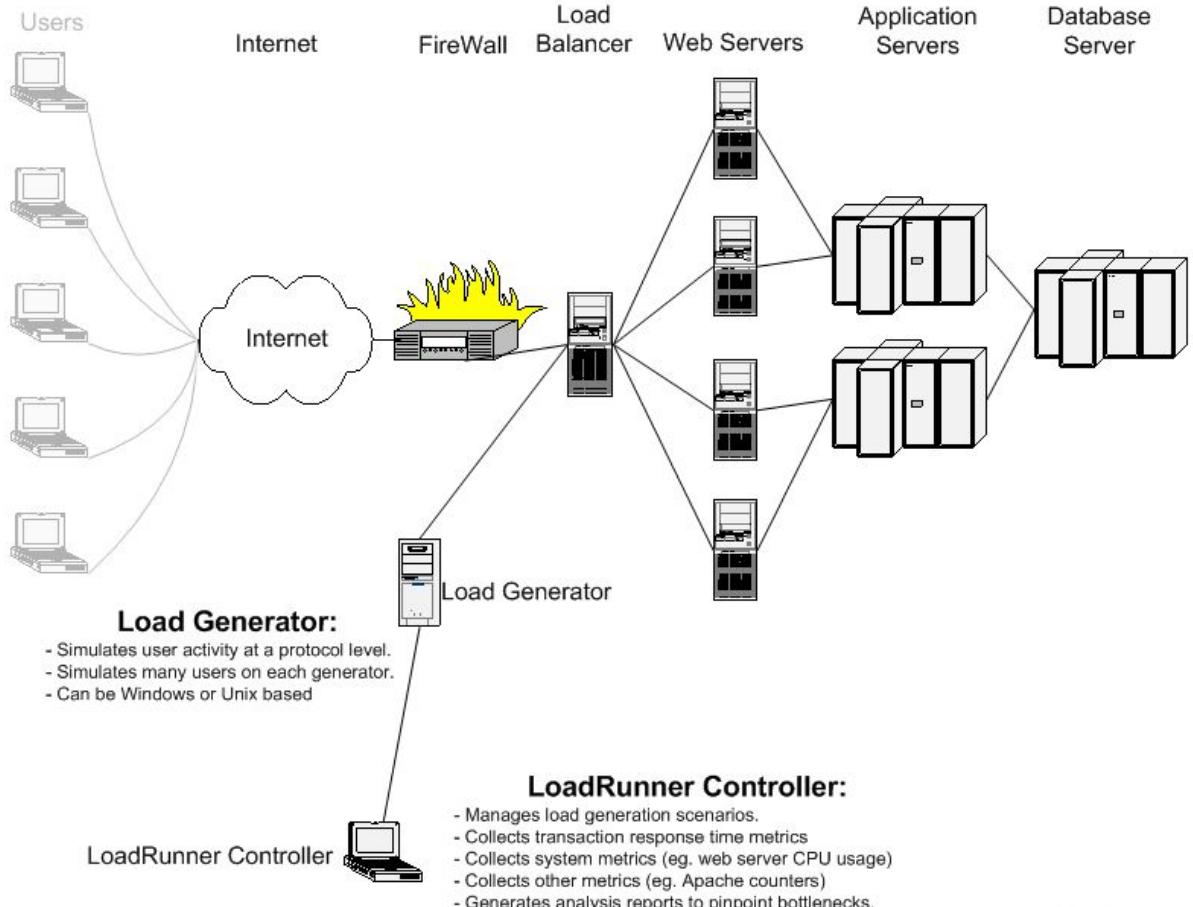
## Stress Testing

- Trying to break the system under test by overwhelming its resources or by taking resources away from it.
- Purpose is to make sure that the system fails and recovers gracefully.

## Example:

- Double the baseline number for concurrent users/HTTP connections.
- Randomly shut down and restart ports on the network switches/routers that connects servers.

# Load Test configuration for a web system



# Load Test Environment

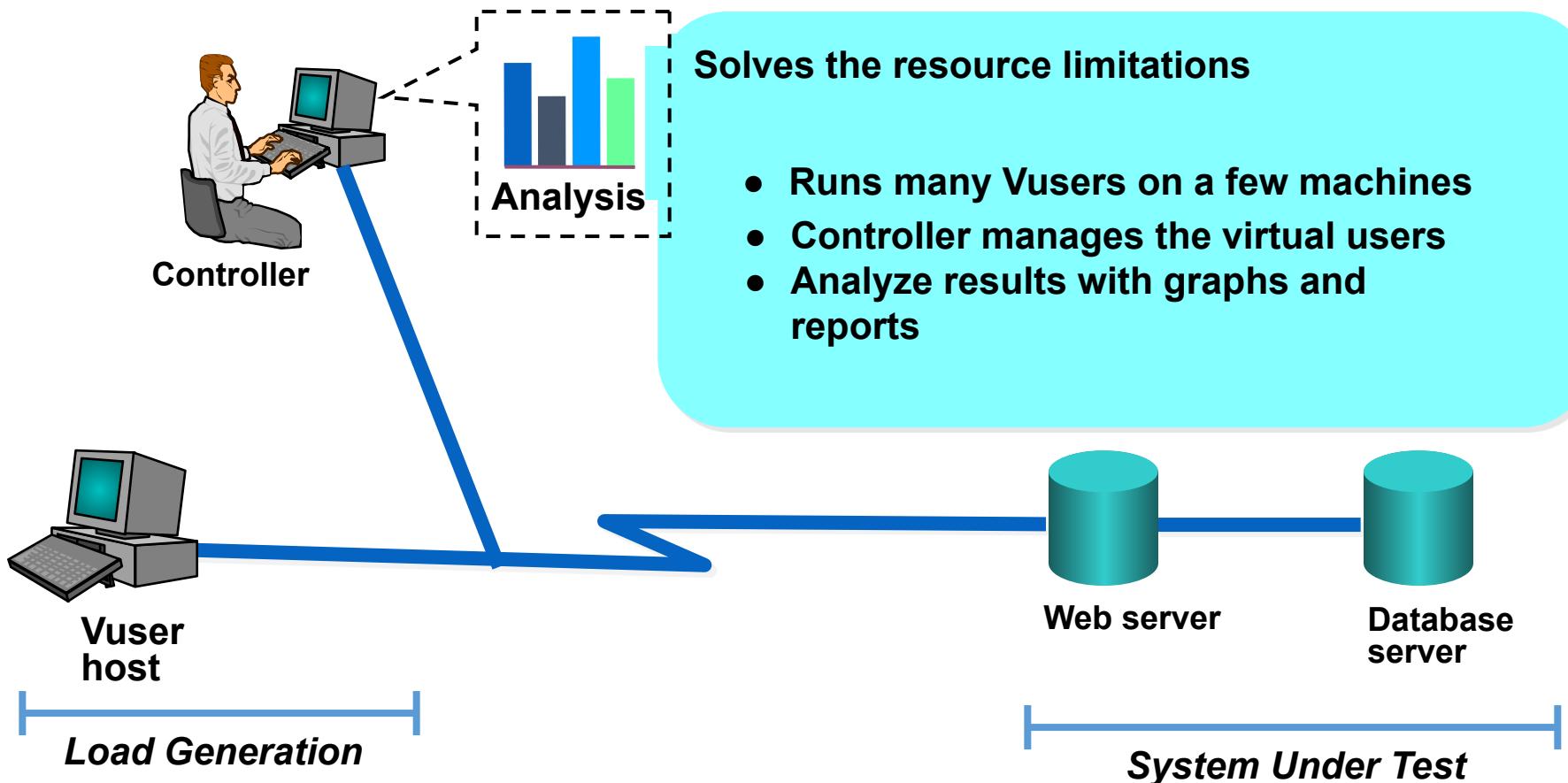
## Load test Tool:

- ✓ Ability to parameterize data.
- ✓ Ability to capture dynamic data and use on subsequent requests.
- ✓ Application infrastructure monitoring.
- ✓ Support for the application's protocols

## Load test Lab must include the following:

- ✓ Test Servers.
- ✓ Databases.
- ✓ Network elements, operating systems and clients and server hardware.

# Load Test Automation



# Tools used for Performance Testing



## Open Source

- OpenSTA
- Diesel Test
- TestMaker
- Grinder
- LoadSim
- Jmeter
- Rubis

## Commercial

- LoadRunner
- Silk Performer
- Qengine
- Empirix e-Load

# Jmeter



- 100% Java desktop application
- For Web and FTP, Java, SOAP/XML-RPC, JDBC applications

## **Advantages:**

- Open Source
- The distributed testing
- Various target systems
- Extensibility: Pluggable samplers allow unlimited testing capabilities

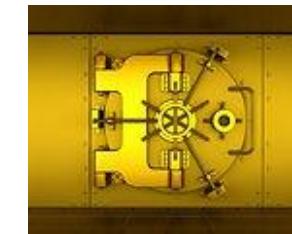
## **Drawbacks:**

- Chart representation quite confuse
- Terminology not very clear
- Necessary to start remote machine one by one
- Remote machines must be declared in a property file before starting application

# Security Testing

# Secure Software

- Confidentiality
  - ❖ Disclosure of information to only intended parties
- Integrity
  - ❖ Determine whether the information is correct or not
- Data Security
  - ❖ Privacy
  - ❖ Data Protection
  - ❖ Controlled Access
- Authentication
  - ❖ Access to Authorized People
- Availability
  - ❖ Ready for Use when expected
- Non Repudiation
  - ❖ Information Exchange with proof



# Defect Escapes and Root Cause Analysis

# Defect Escapes

- Defect escape refers to a situation where a defect wasn't discovered by the test teams, but instead, was found by customers
- When problems are exposed by customers, they can be quite costly
- The further back in the software development process that defects are uncovered, the less expensive they are
- Defect escape rate examines the rate at which code containing bugs or other flaws is pushed into production
- It is a measure of the testing process's effectiveness and the quality assurance (QA) strategy overall