



# **ITMD 536 Software Testing & Maintenance**

## **Chapter 2**

# **Testing throughout the software life cycle**



## Objectives

- ▶ What is beyond Unit Testing?
- ▶ What is testing throughout the software life cycle
- ▶ What is the difference between Waterfall, V-Model & Agile Testing?
- ▶ What are the Test different Levels?
- ▶ What are the different Test Types?
- ▶ What is Maintenance Testing?



# Testing throughout the software life cycle

- ▶ Test is not a stand-alone activity
- ▶ There are many forms of testing
- ▶ There are various models, test levels and types of testing
- ▶ Maintenance testing



## 2.1 Software Development Models

- ▶ Testing is focused on verification and validation testing
- ▶ **Verification:** Confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled (ISO)



## 2.1 Software Development Models

- ▶ **Validation:** Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have fulfilled.



## 2.1 Software Development Models

- ▶ **Waterfall Testing** is the top-down development and design by functional decomposition.

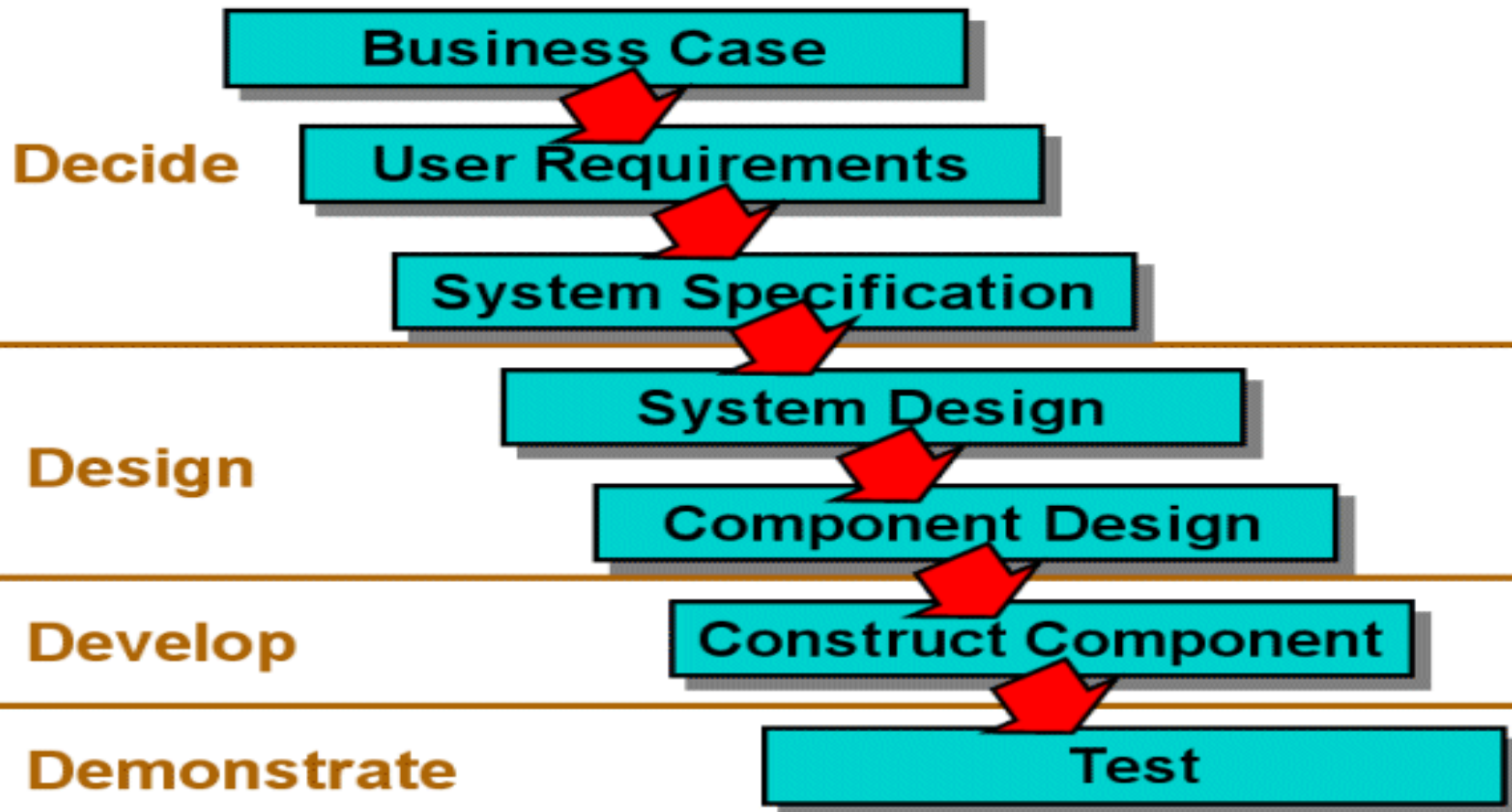


## Waterfall Model

- ▶ Need, wish, policy, law
- ▶ User Requirements,
- ▶ System Requirements,
- ▶ Global Design,
- ▶ Detail Design,
- ▶ Implementation
- ▶ Testing



## Waterfall Model







## V-Model Testing

- ▶ **V-Model Testing** - A framework to describe the software development lifecycle activities from requirements specification to maintenance. The V-model illustrates how testing activities can be integrated into each phase of the software development lifecycle.



## V-Model Test Level

- ▶ A group of test activities that are organized and managed together.
- ▶ A test level is linked to the responsibilities in a project.
- ▶ Examples of test levels are component test, integration test, system test and acceptance test



V-Model uses four levels of testing

- ▶ **Component testing** – searching for defects in and verifies the functioning of software components (modules, programs, objects, classes, etc.)
- ▶ **Integration testing** – The process of combining components or systems into a large assemblies.



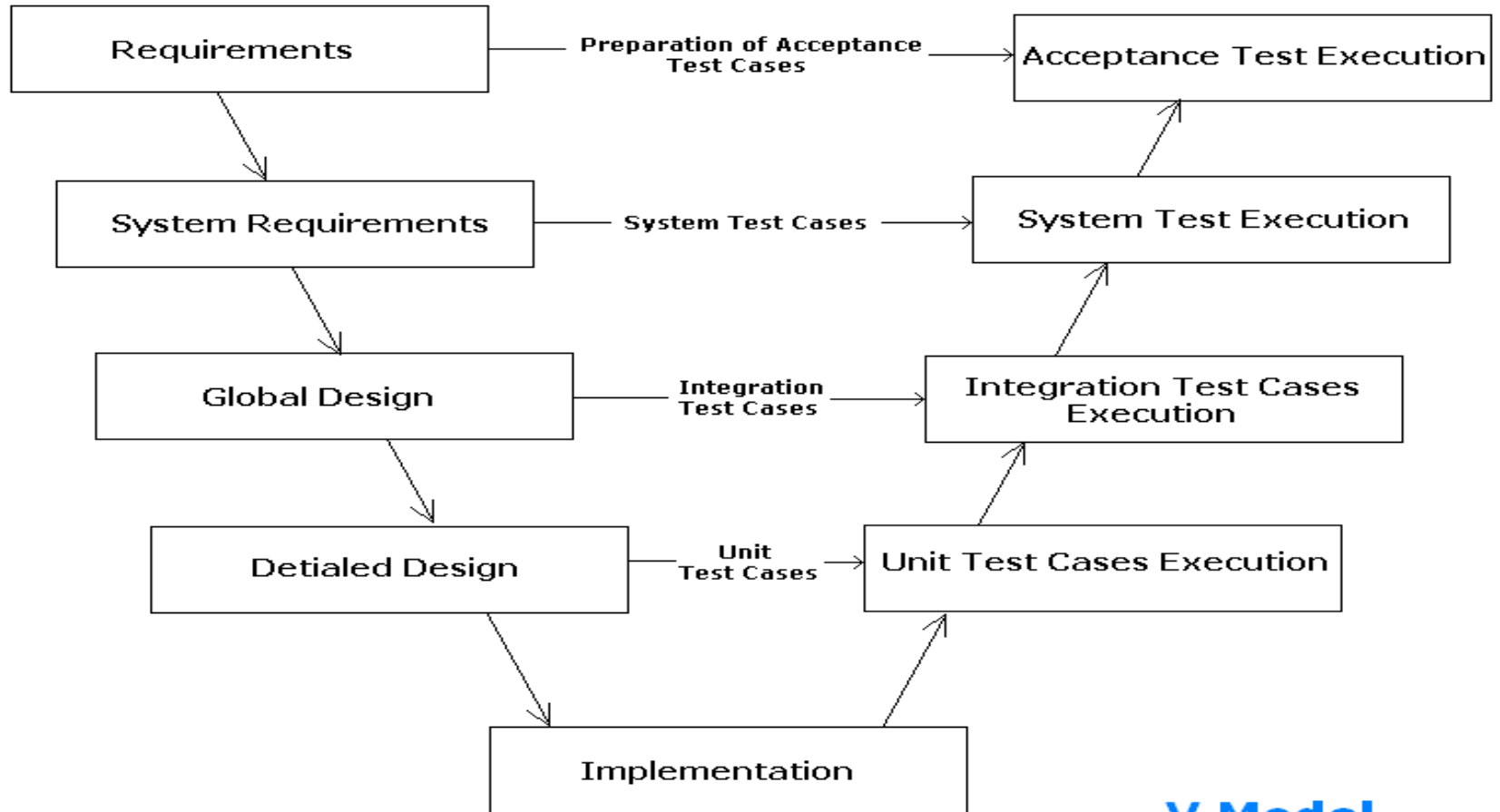
V-Model uses four levels of testing

- ▶ **System testing** – The process of testing an integrated system to verify that it meets specified requirements.
- ▶ **Acceptance testing** – Validation testing with respect to user needs, requirements and business processes conducted to determine whether or not to accept the system.



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY



**V Model**



## Commercial off-the-shelf software (COTS)

- ▶ A software product that is developed for the general market, i.e. for a large number of customers, and that is delivered to many customers in identical format.
- ▶ Purchaser may only perform integration testing and may be acceptance testing.



## Commercial off-the-shelf software (COTS)

- ▶ The acceptance testing can include both testing of system functions but also testing quality attributes such as performance and other non-functional tests.



## Performance Testing

- ▶ **Performance testing** – The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate.
- ▶ The process of testing to determine the performance of a software product.





## Performance Testing Tools

- ▶ Load generation can simulate either multiple users or high volumes of input data.
- ▶ During execution, response time measurements are taken from selected transactions and these are logged.
- ▶ Performance testing provides test logs and graphs of load against response times.



# Performance – Stress Testing



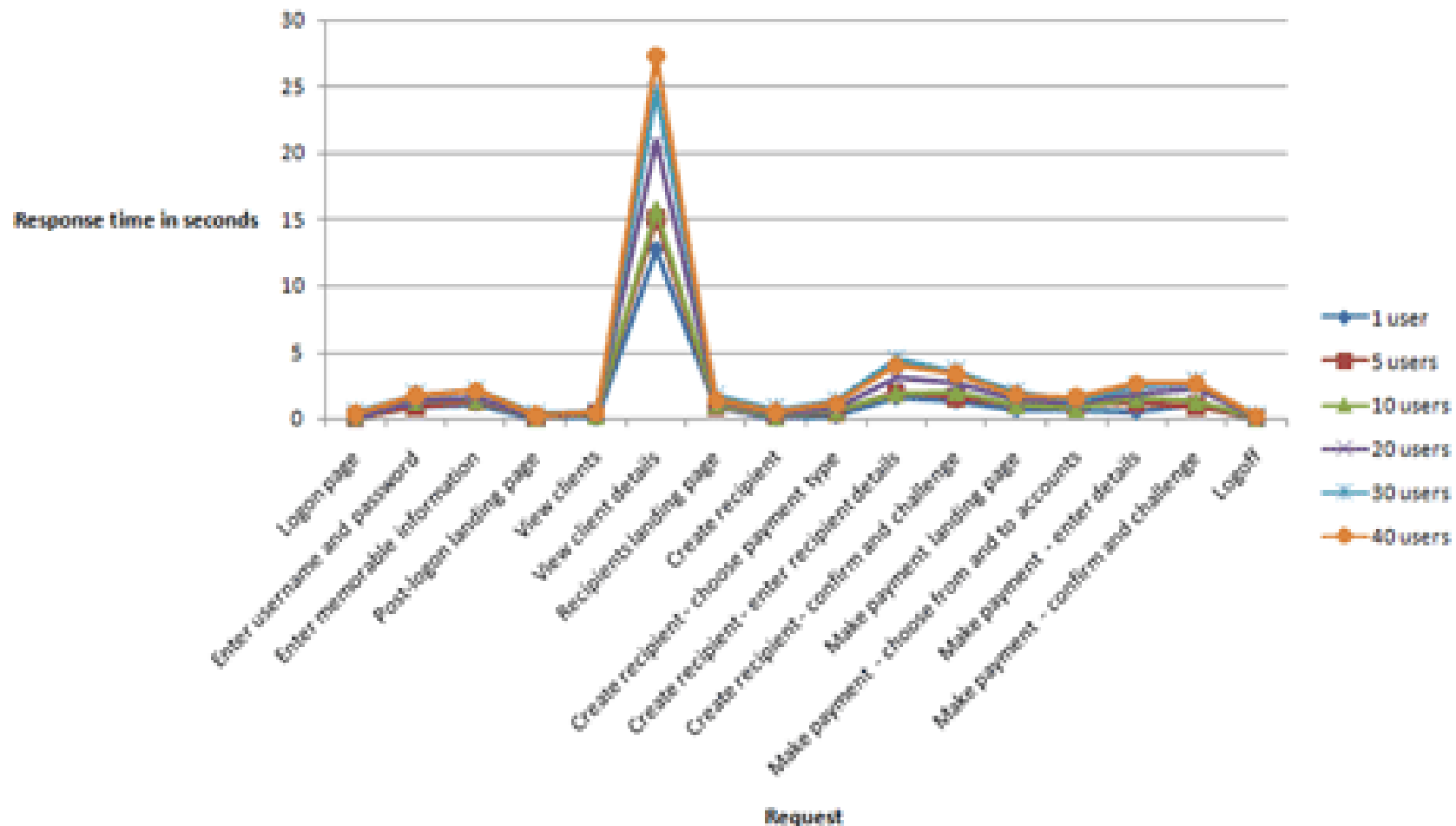


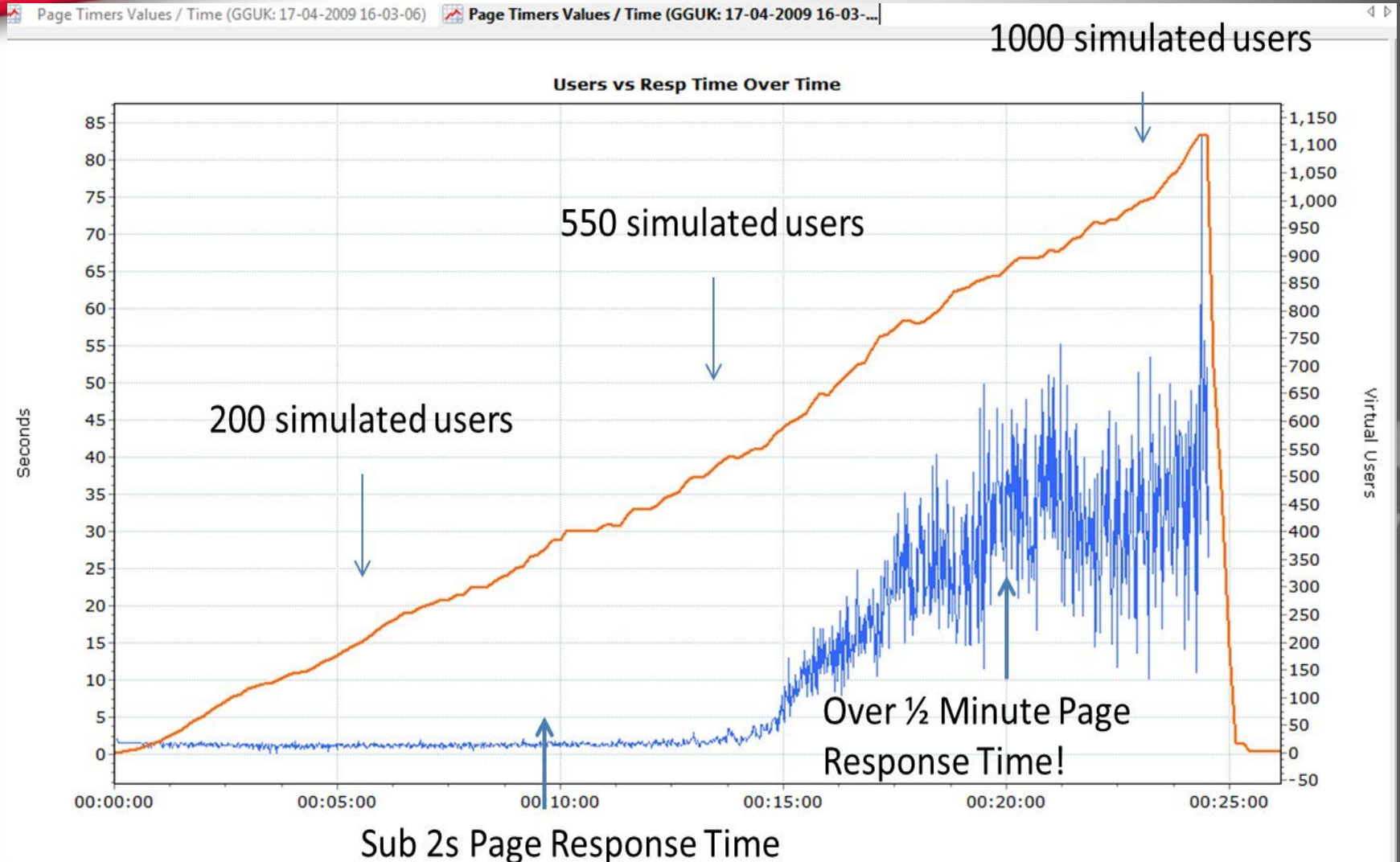
## Load Testing

- ▶ A type of performance testing conducted to evaluate the behavior of a component or system with increasing load, e.g., numbers of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system.



## Average response times (raw)





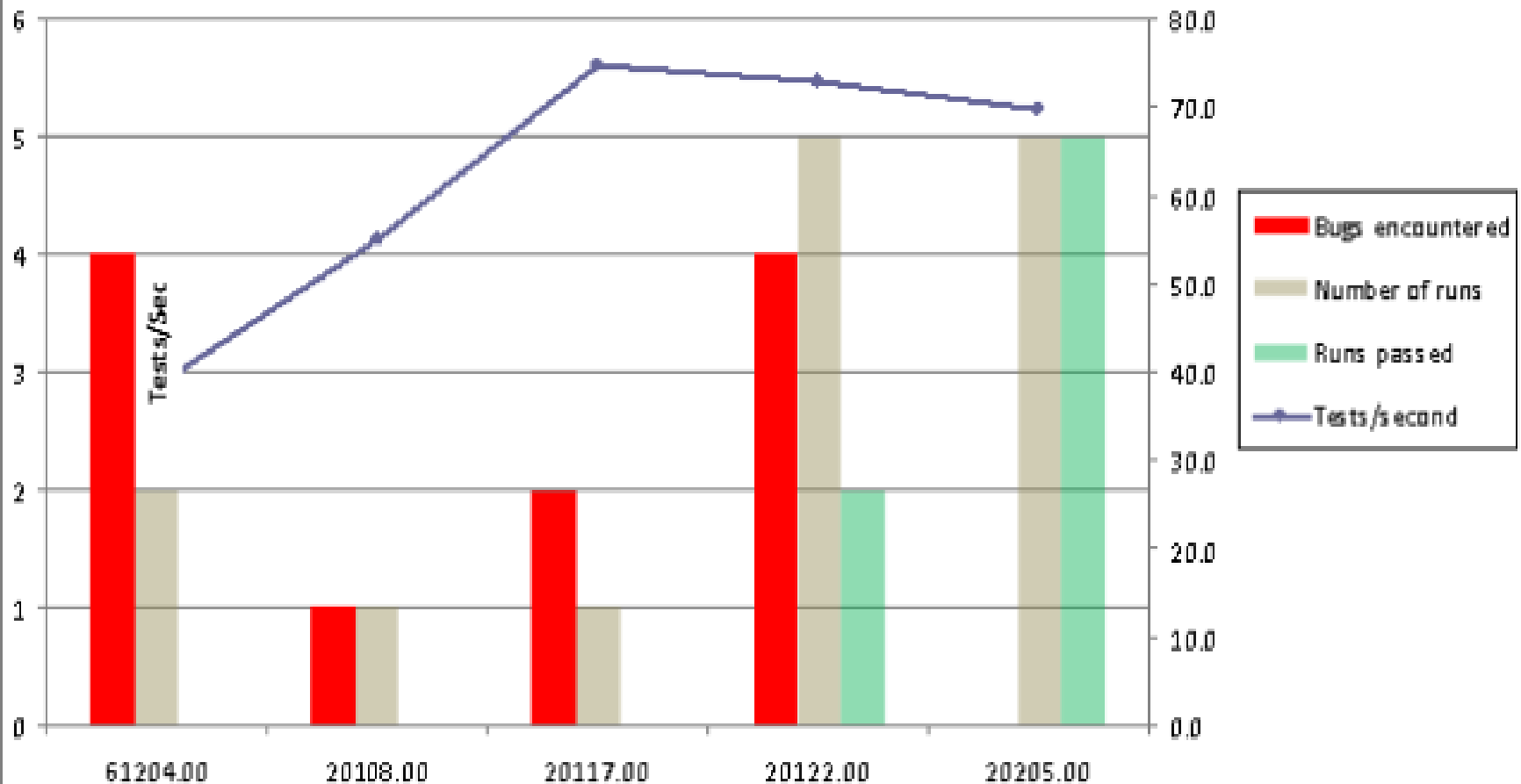


## Stress Testing

- ▶ A type of performance testing conducted to evaluate a system or component at or beyond the limits of its anticipated or specified work loads, or with reduced availability of resources such as access to memory or servers.



## TSADT - Stress Trends Short Haul





## Capability Maturity Model (CMM)

- ▶ A first level staged framework that describes the key elements of an effective software process.
- ▶ The Capability Maturity Model covers best-practices for planning, engineering and managing software development and maintenance.





## Capability Maturity Model Integration (CMMi)

- ▶ A framework that describes the key elements of an effective product development and maintenance process.
- ▶ The Capability Maturity Model Integration covers best-practices for planning, engineering and managing product development and maintenance.



## Iterative life cycles

- ▶ **Iterative development model** – A development lifecycle where a project is broken into a usually large number of iterations. An iteration is a complete development loop resulting in a release (internal or external) of an executable product, a subset of final product and grows into final product.



Phase I					Phase II					Phase III				
Requirements	Development	Build	Testing	Deployment	Requirements	Development	Build	Testing	Deployment	Requirements	Development	Build	Testing	Deployment

Figure 3: Iterative Development Model



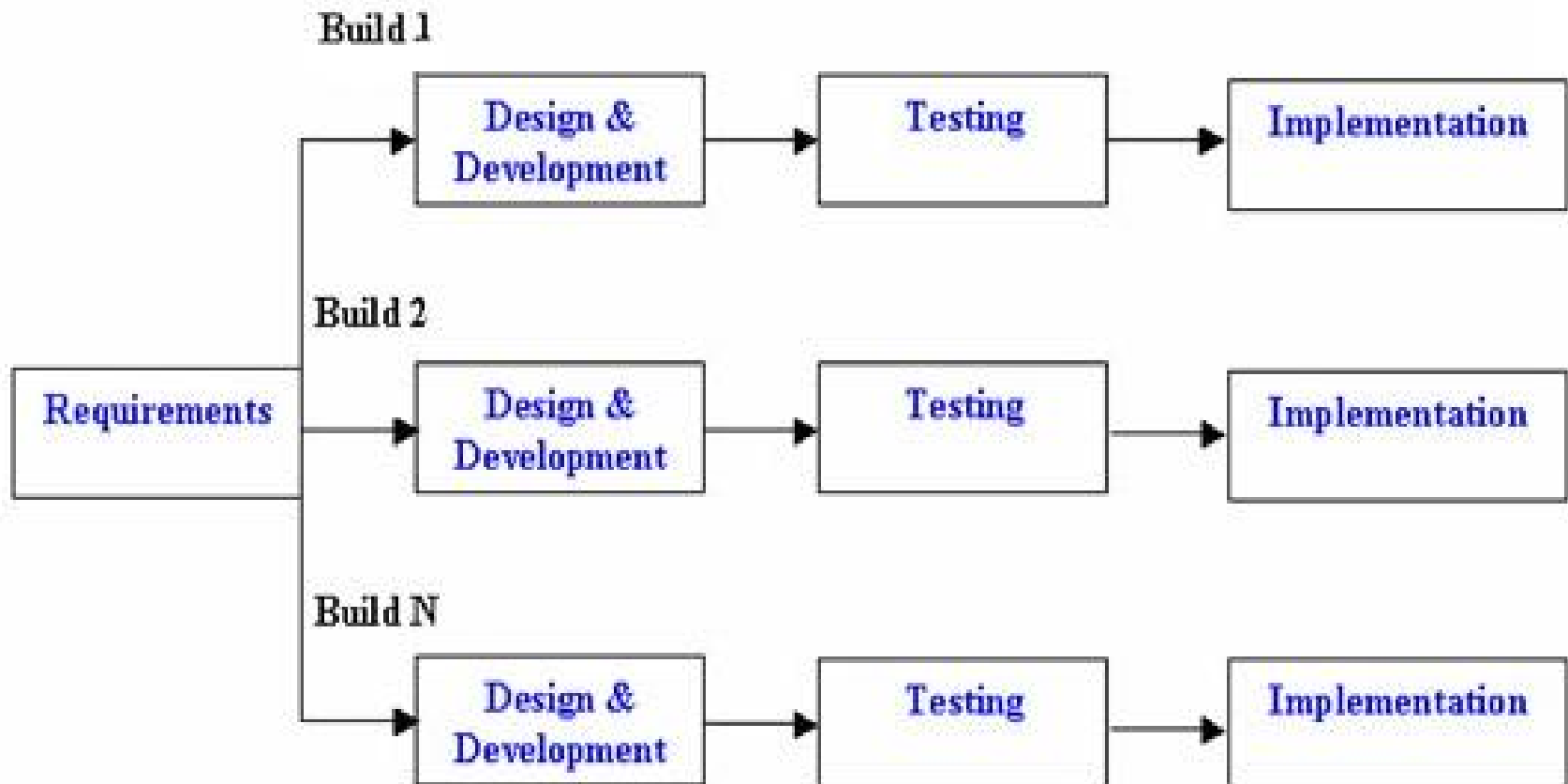
## Incremental Development Model

- ▶ A development lifecycle where a project is broken into a series of increments, each of which delivers a portion of the functionality in the overall project requirements.
- ▶ Requirements are prioritized and delivered in priority order in the appropriate increments.



## Incremental Testing

- ▶ Testing where components or systems are integrated and tested one or some at a time, until all the components or systems are integrated and tested.



Incremental Life Cycle Model



## Rapid Application Development - RAD

- ▶ Rapid Application Development is formally a parallel development of functions and subsequent integration.
- ▶ This methodology allows early validation of technology risks and a rapid response to changing customer requirements.



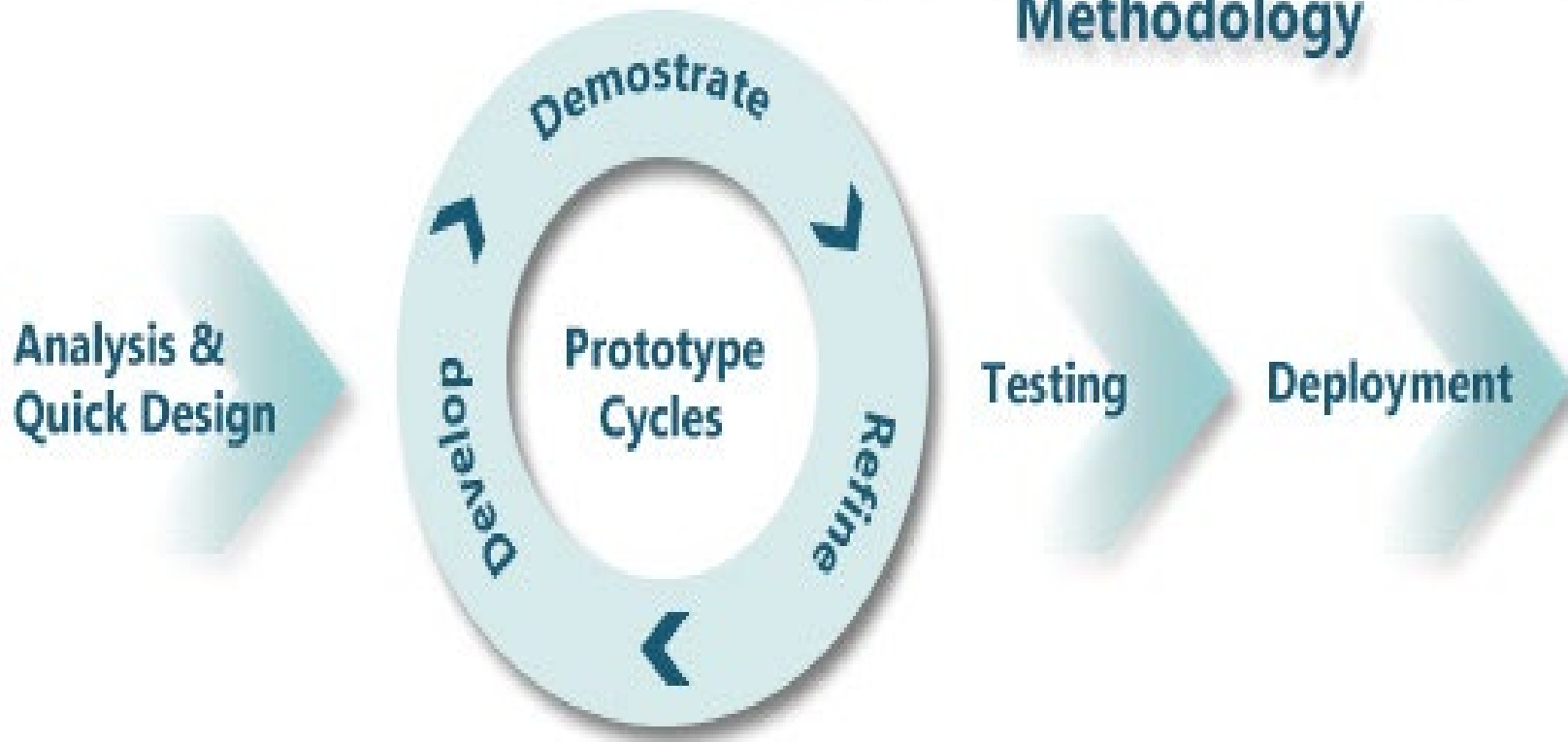
## Rapid Application Development - RAD

- ▶ RAD provides early visibility of the product to the customer.
- ▶ Customer can provide feedback on the design and can decide, based on the existing functionality, whether to proceed with the development, what functionality to include in the next delivery cycle or stop the project.





## Rapid Application Development Methodology





## Agile Software Development

- ▶ A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.



## Agile Teams

- ▶ Most agile teams use Scrum, a management framework for iterative incremental development projects.
- ▶ Agile teams consist of 5-9 people



## Agile Manifesto a Statement of Value

- ▶ Individuals & Interactions
- ▶ Working Software
- ▶ Customer collaboration
- ▶ Responding to Change
- ▶ Process & Tools
- ▶ Comprehensive documentation
- ▶ Contract negotiations
- ▶ Following a Plan



## Test Driven Development

- ▶ A way of developing software where the test cases are developed, and often automated, before the software is developed to run those test cases.



## Agile Testing

- ▶ Testing practice for a project using agile methodologies, such as extreme programming (XP), treating development as the customer of testing and emphasizing the test-first design paradigm.



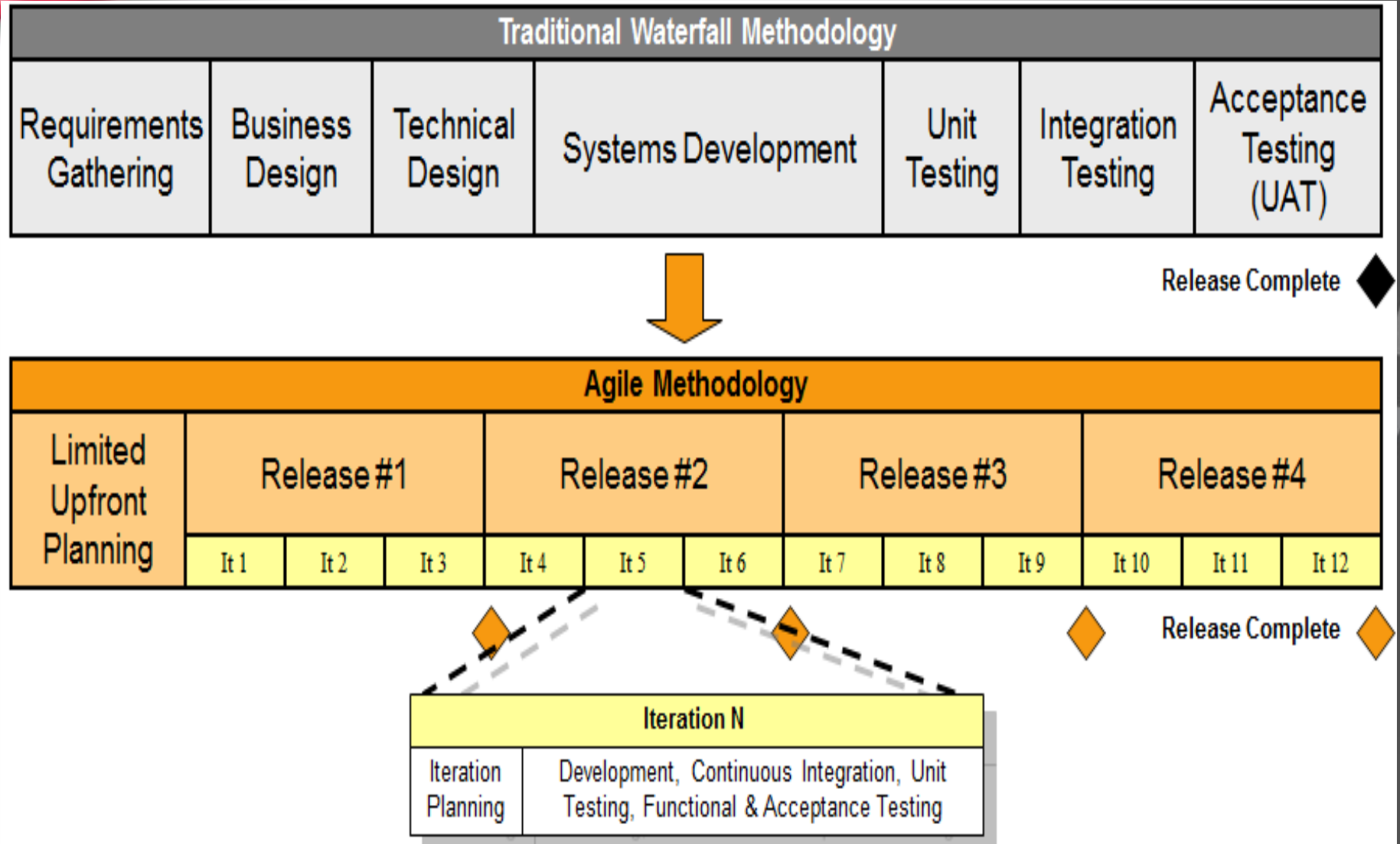
## Benefits of Agile Testing

- ▶ Focus on working software and good quality
- ▶ Test driven development
- ▶ Accessibility of business stakeholders to help testers
- ▶ Self-organizing teams
- ▶ Simplicity of design that should be easier to test



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY







## Test driven development

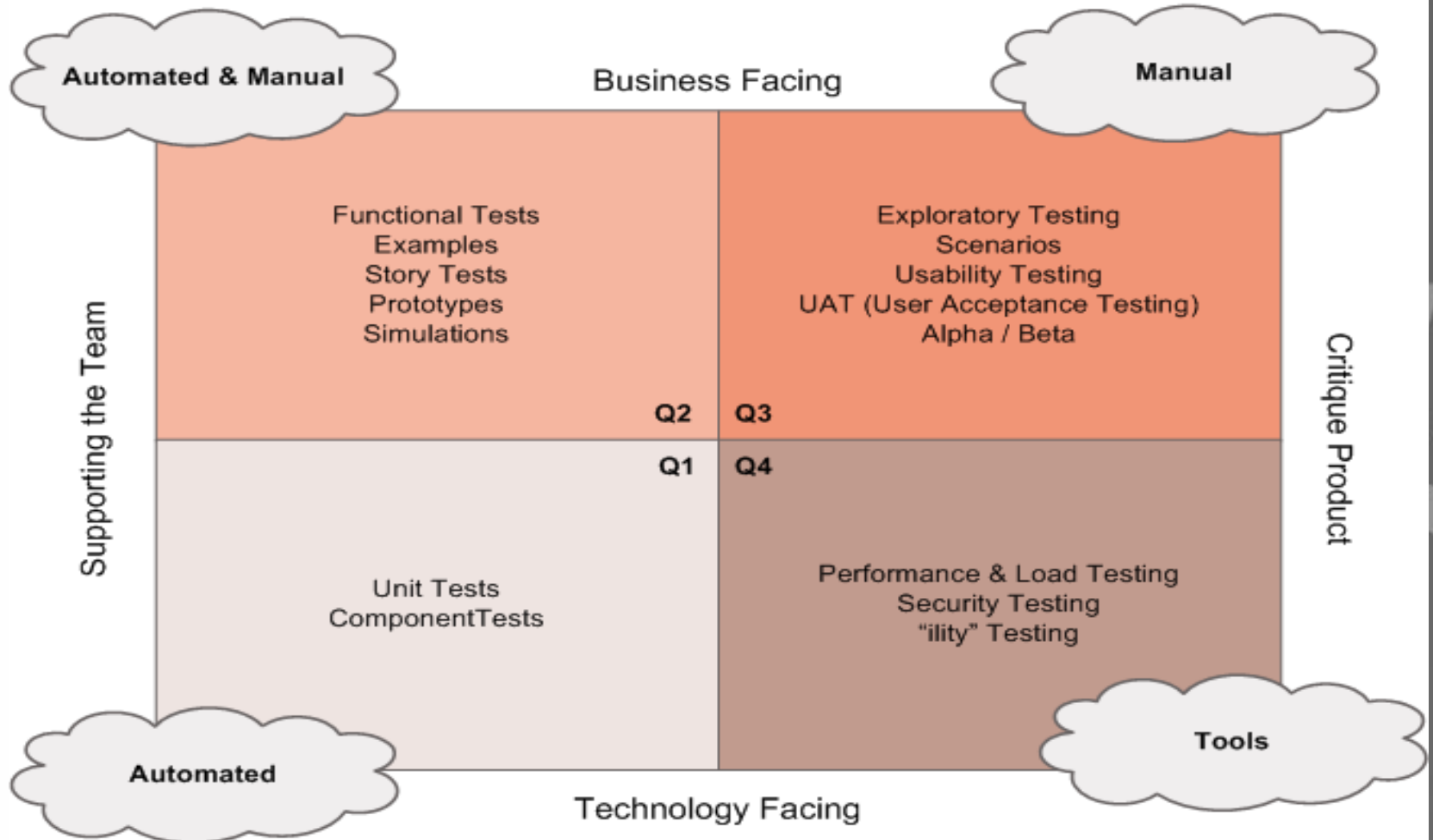
- ▶ Test driven development is a way of developing software where the test cases are developed, and often automated, before the software is developed to run those test cases.



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

## Agile Testing Quadrants





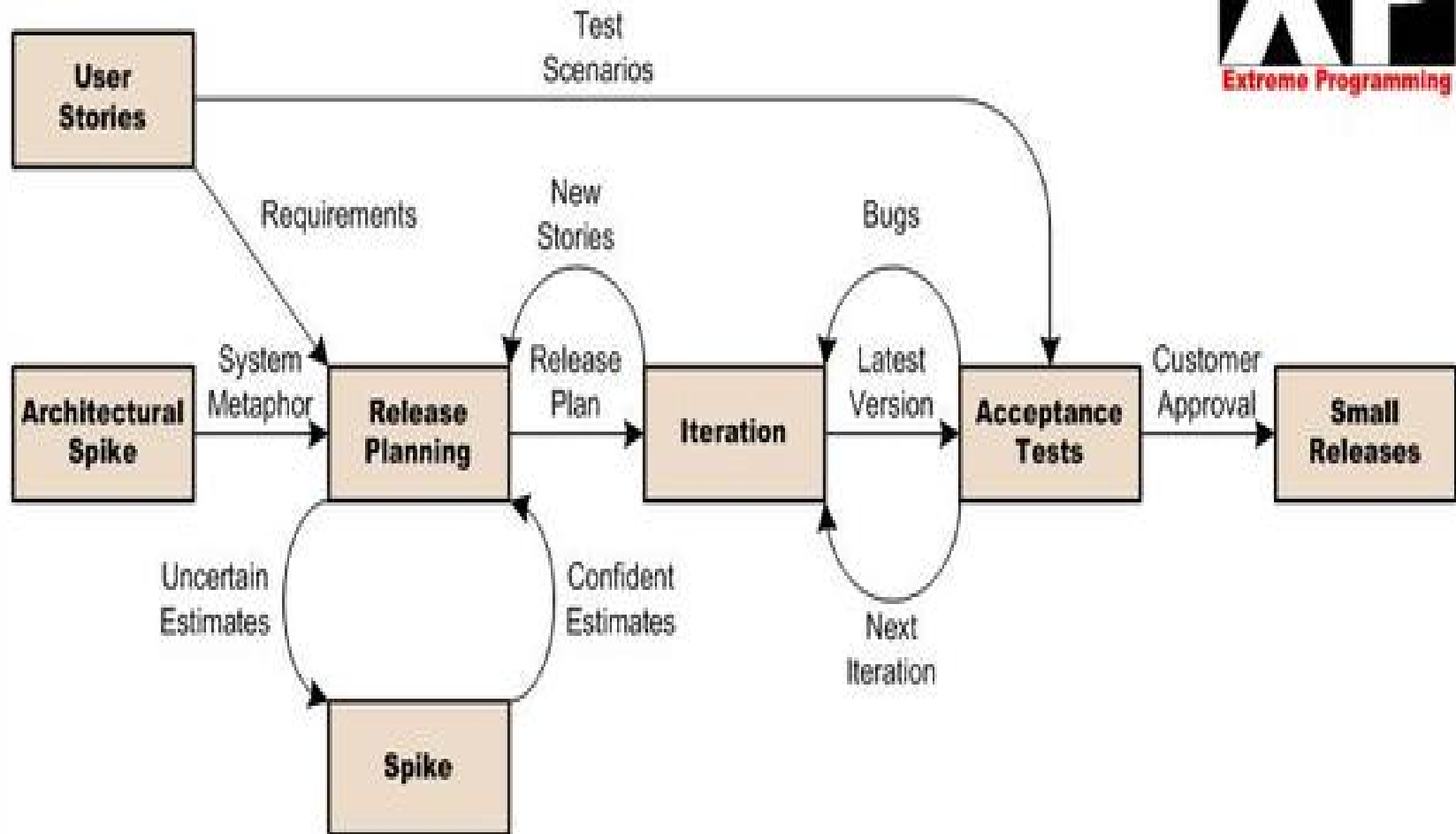
## Extreme Programming (XP)

- ▶ A software engineering methodology used within agile software development whereby core practices are programming in pairs, doing extensive code review, unit testing of all code, and simplicity and clarity in code.



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY





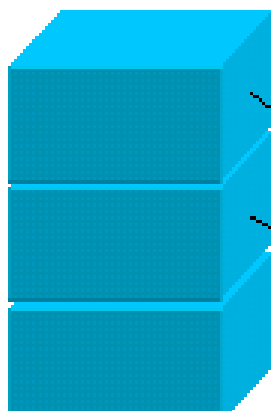
## SCRUM

- ▶ Scrum is an iterative incremental framework for managing projects commonly used with agile software development.
- ▶ Changes are incorporated continuously into the software build.

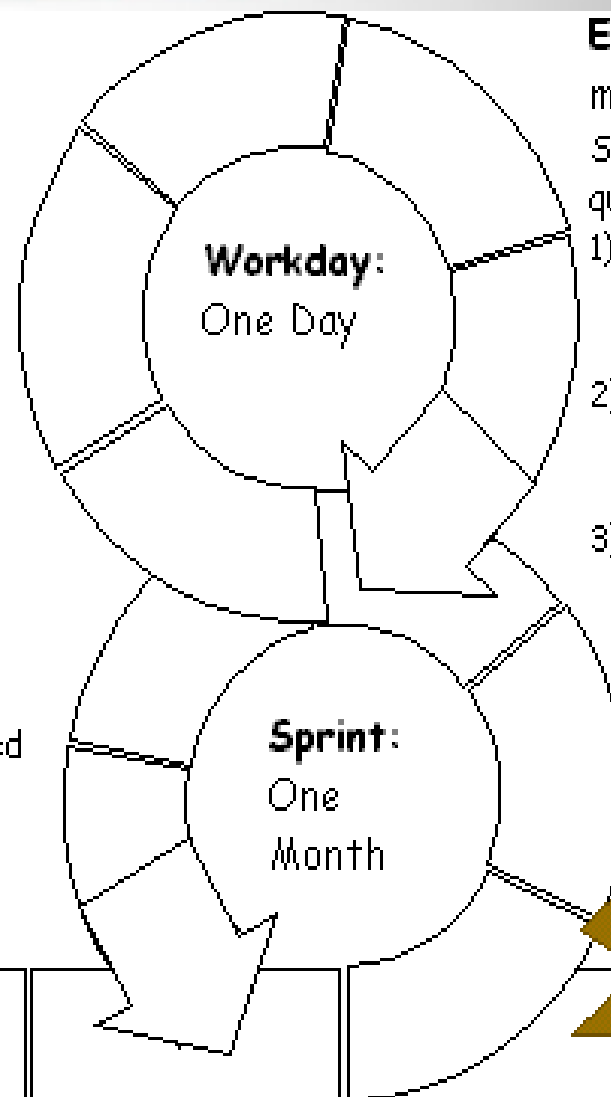
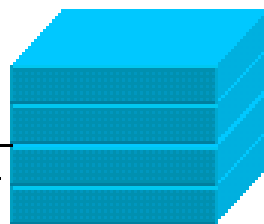


## SCRUM Sprint Cycle

**Product Backlog:**  
Prioritized list of features required by the customer



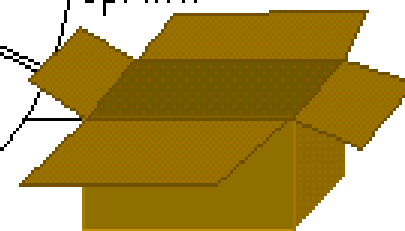
**Sprint Backlog:**  
Features to be done this sprint  
Features are expanded into smaller tasks.



**Every Day**, a 15-minute meeting is held, and the SCRUM Master asks the 3 questions:

- 1) What have you accomplished since the last meeting?
- 2) Are there any obstacles in the way of meeting your goal?
- 3) What will you accomplish before the next meeting?

**New Functionality** is demonstrated at the end of each sprint.



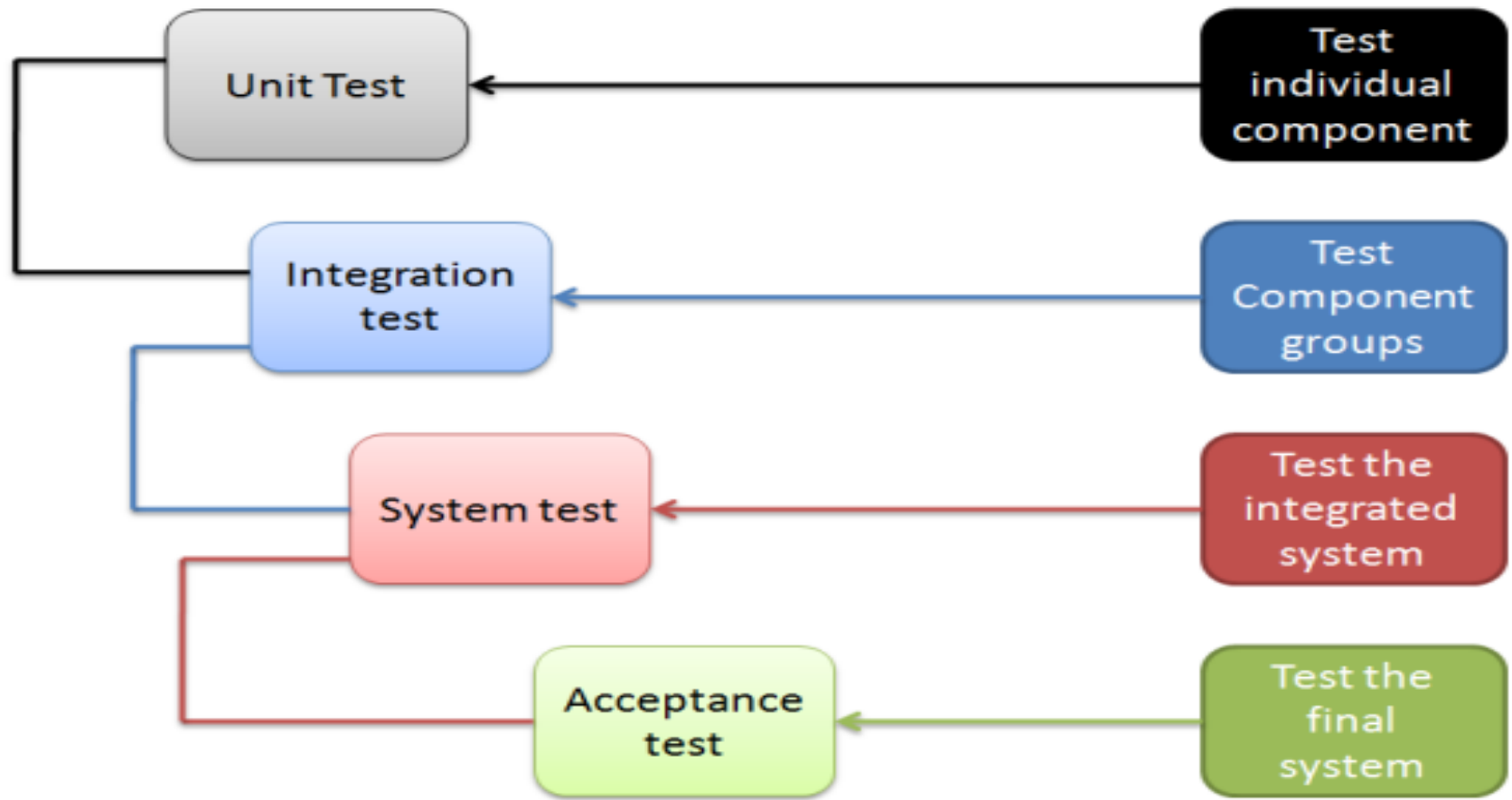


## 2.2 Test Levels

- ▶ A group of test activities are organized and managed together.
- ▶ A test level is linked to the responsibilities in a project.
- ▶ Component Test,
- ▶ Integration Test
- ▶ System Test and
- ▶ Acceptance Test



## Test Levels







## Testing Best Practices

- ▶ Measurable effectiveness & efficiency
- ▶ The test basis
- ▶ The item build – test object
- ▶ The typical defects and failures
- ▶ Applicable requirements for test harnesses
- ▶ Approaches we intend to use



## Efficiency and Efficiency Testing

- ▶ The capability of the software product to provide appropriate performance, relative to the amount of resources used under state conditions. (ISO)
- ▶ The process of testing to determine the efficiency of a software product.



## Component Testing (Unit or Module)

- ▶ A single procedure
- ▶ A function
- ▶ A body of code that implements a single function
- ▶ Source code that fits on one page
- ▶ Code done in 4-40 hours – 300 lines
- ▶ The smallest body of code that can be compiled and executed by itself



## Stubs and Drivers

- ▶ Component testing may be done in isolation from the rest of the system depending on the context of the development life cycle and the system.
- ▶ Stubs and drivers are used to replace the missing software and simulates interface between software components



## Stub

- ▶ A skeletal or special-purpose implementation of a software component, used to develop or test a component that calls or is otherwise dependent on it. It replaces a called component.

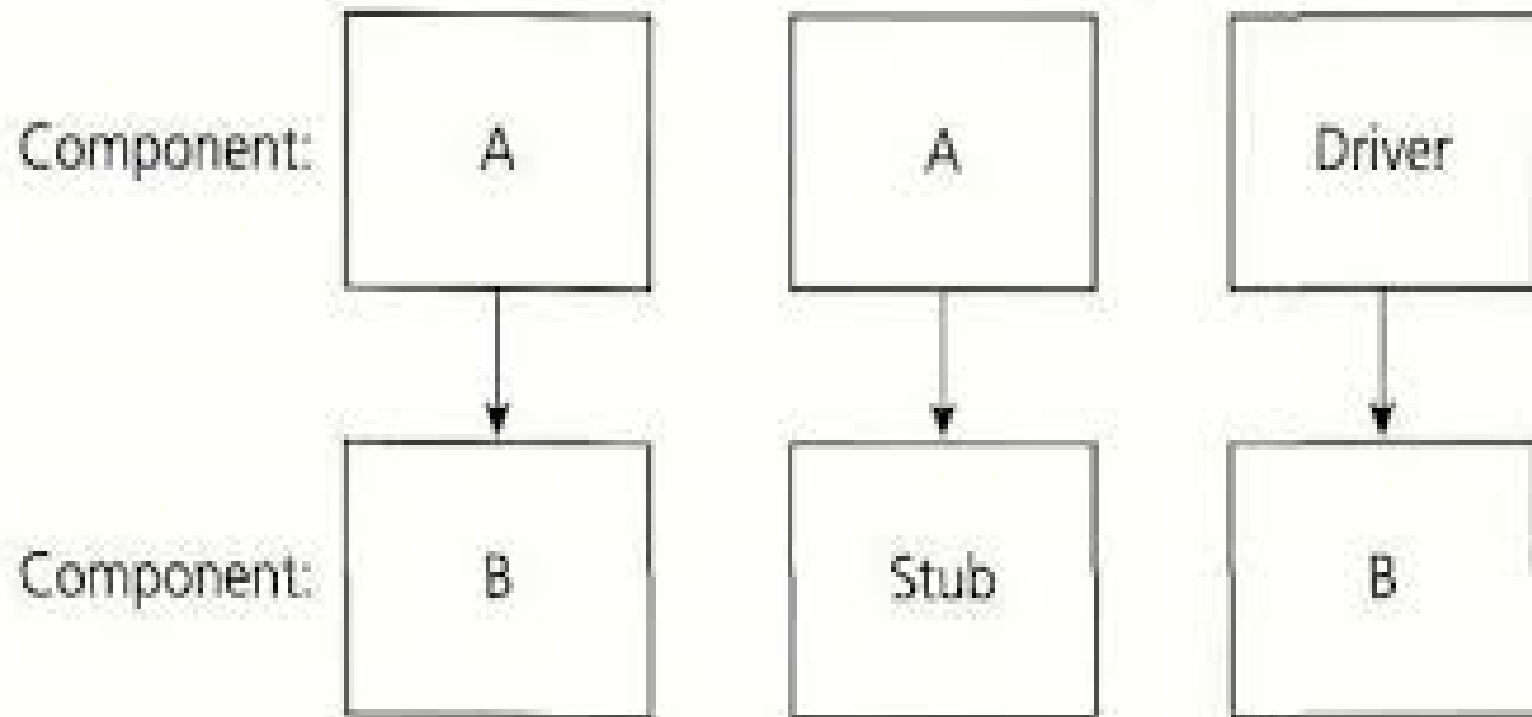


## Driver (Test Driver)

- ▶ A software component or test tool that replaces a component that takes care of the control and/or the calling of a component or system.



## Stubs and Drivers





## Robustness Testing

- ▶ Robustness: The degree to which a component or system can function correctly in the presence of invalid inputs or stressful environmental conditions
- ▶ Testing to determine the robustness of the software product
- ▶ Error or fault tolerance (ISO)





## Component Testing

- ▶ Include testing of functionality and specific non-functional characteristics such as resource-behavior (e.g. memory leaks), performance or robustness testing, as well as structural testing (e.g. decision coverage)



## Test Driven Development

- ▶ A way of developing software where the test cases are developed and often automated, before the software is developed to run those test case
- ▶ Extreme programming (XP), prepares and automates test cases before coding

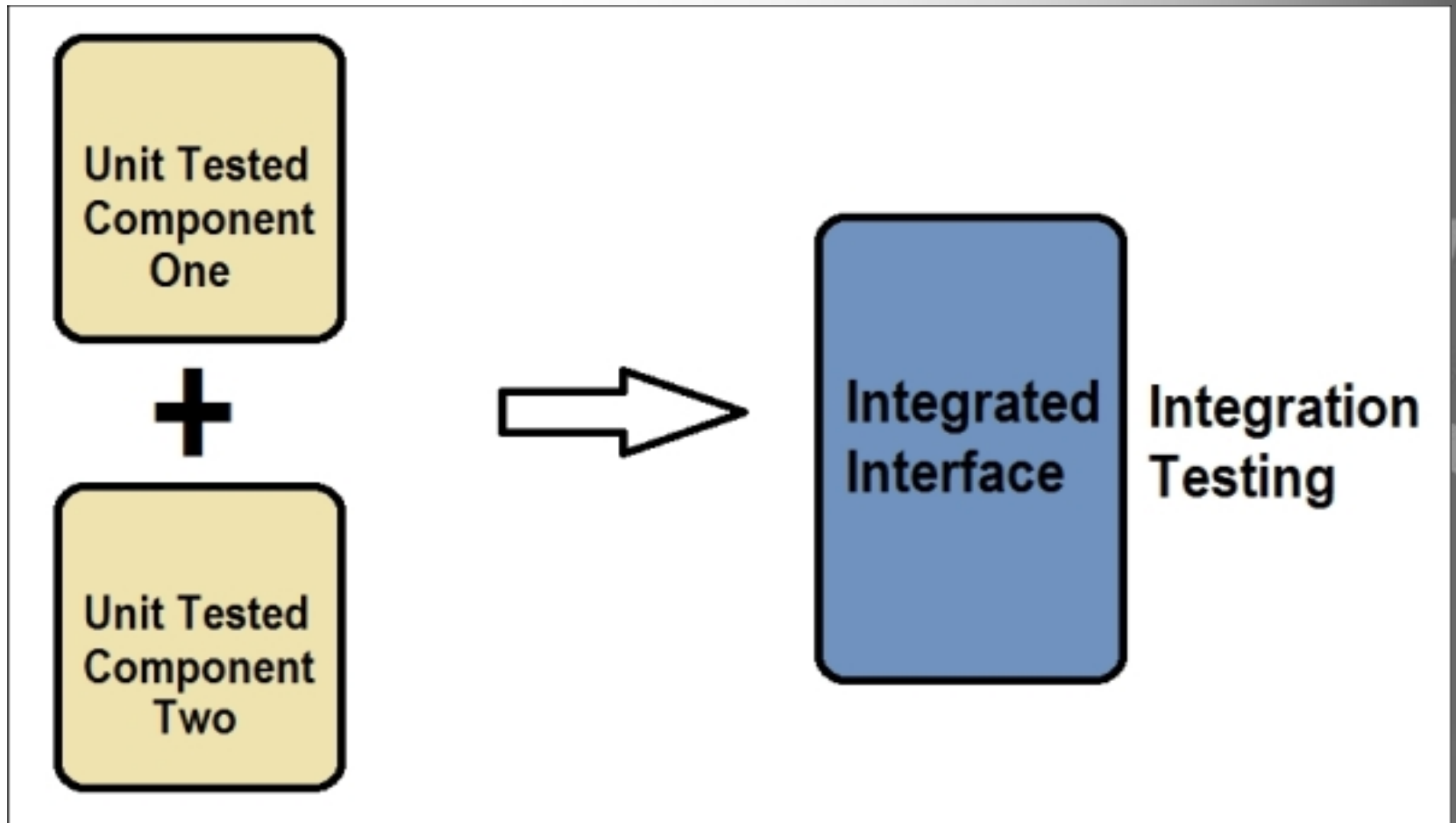


## Integration Testing

- ▶ The process of combining components or systems into larger assemblies
- ▶ Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems



## Component to Integration Testing





## System Integration Testing

- ▶ Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet)
- ▶ Integration testing may be carried out by the developers, but can be done by a separate team of specialist integration testers



## System Testing

- ▶ The process of testing an integrated system to verify that it meets specified requirements
- ▶ Includes tests based on risk analysis reports, system, functional, or software requirements specifications, business processes, use cases, and interactions with the operating system and system resources



## System Testing

- ▶ System testing is most often the final test on behalf of development to verify that the system to be delivered meets the specification and its purpose may be to find as many defects as possible.



## System Testing

### System Testing

- Performance
- Stress
- Scalability
- Localization
- Interoperability
- Reliability
- Security





## System Testing

- ▶ System testing should investigate both functional and non-functional requirements of the system.



## System Testing

- ▶ **Functional Requirement** – A requirement that specifies a function that a component or system must perform
- ▶ **Non-Functional Requirement** – A requirement that attributes to reliability, efficiency, usability, maintainability and probability

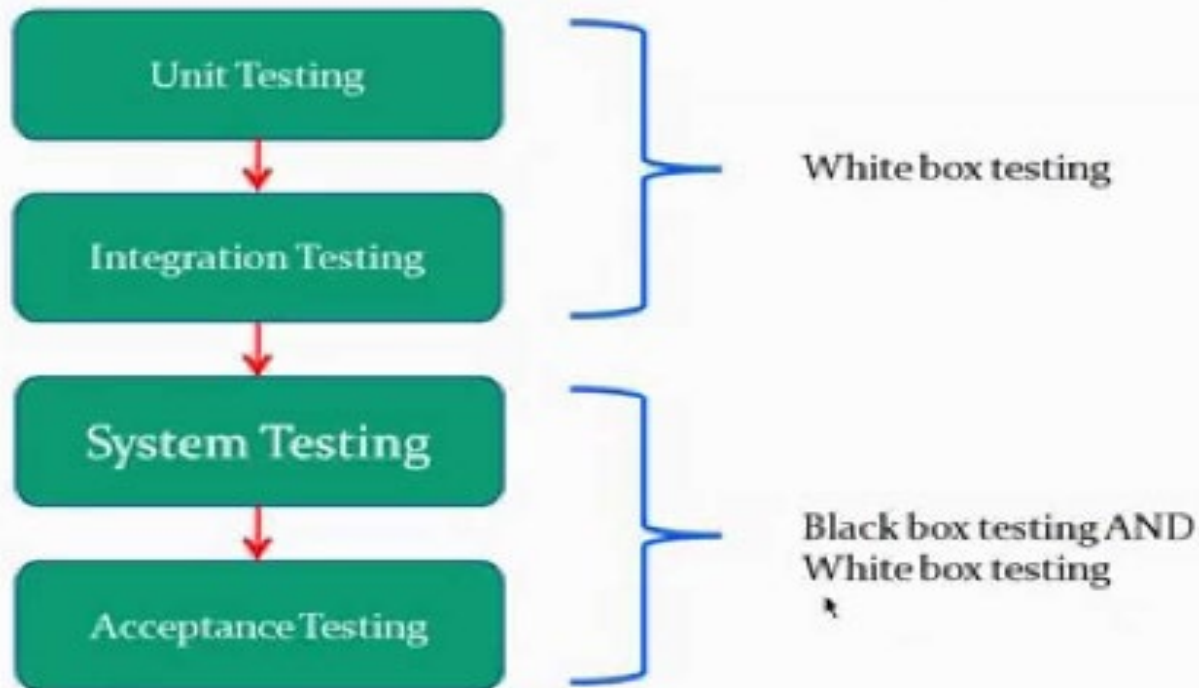


## System Testing

- ▶ System testing requires a controlled **test environment** including controlled software versions, testware and the test data
- ▶ The test environment should be similar to the production environment in order to minimize the risk of environment-specific failures



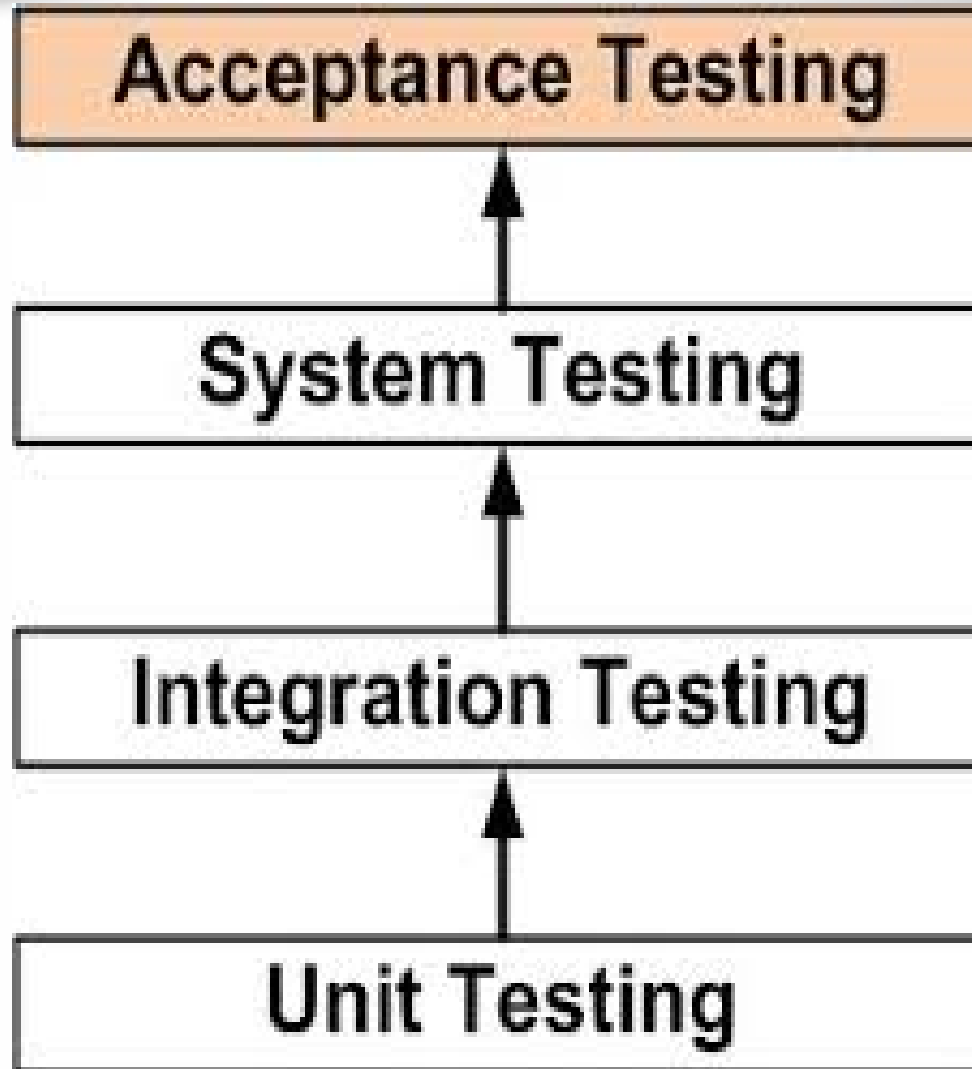
## System Testing





## Acceptance Testing

- Formal testing with respect to user need, requirements and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system





IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

# Acceptance Tests





## Alpha Testing

- ▶ Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing.





IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

# ALPHA TESTING

WHERE IT ALL BEGINS



## Beta Testing

- Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. It is an external form of testing to acquire the feedback from the market



## Beta Testing (Field Testing)





## Test Types

- ▶ A group of test activities aimed at testing a component or system focused on a specified test objective, i.e. functional test, usability test, regression test etc. A test type may take place on one or more test levels or test plans.



## Functional Testing

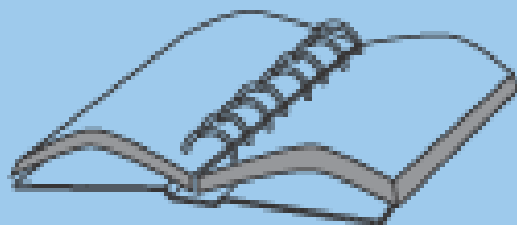
- ▶ Testing based on analysis of the specifications of the functionality of a component or system
- ▶ Functional testing is also referred to as black-box testing (specification based testing)
- ▶ The process of testing to determine the functionality of a software product (ISO)



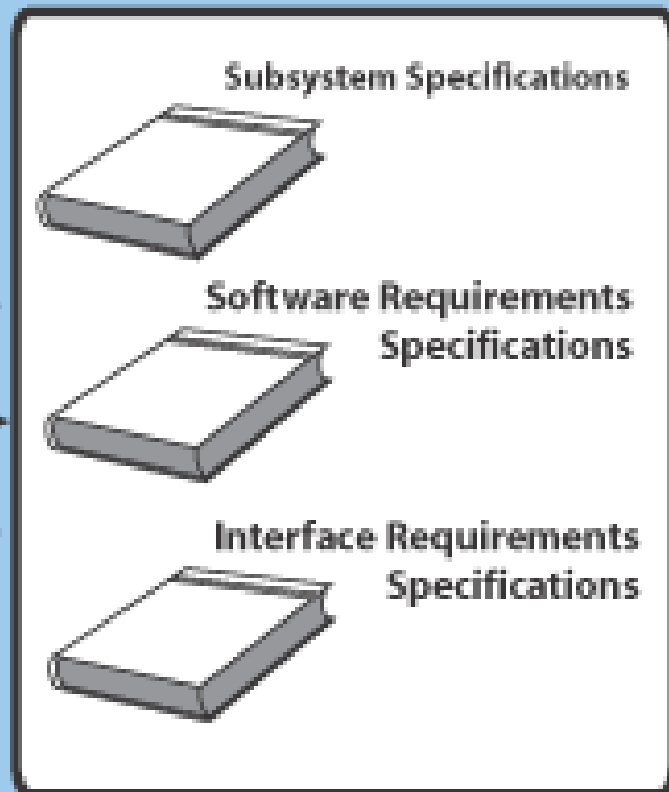
## Function Test

### Objectives

- Demonstrate that all software requirements are met
- Demonstrate that security safeguards are met



**System Requirements Specifications**





## Functional Testing

- ▶ Focused on suitability, interoperability testing, security, accuracy and compliance.
- ▶ Interoperability testing determines the interoperability of a software product



## Functional Testing

- ▶ **Security:** Attributes of software products that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data
- ▶ **Security Testing:** Testing to determine the security of the software product





# Security Testing





## Non-Functional Testing

- ▶ Non-functional testing is performed at all levels
- ▶ Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, maintainability and portability



## Non-Functionality Testing

- ▶ **Reliability Testing:** The process of testing determine the reliability of a software product
- ▶ The ability of the software product to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations (robustness)



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY



## Reliability

/re-ly-a-bi-li-ti/

The ability of the software  
to function under the given  
conditions



## Non-Functionality Testing

- ▶ **Usability Testing:** Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions
- ▶ The capability of the software to be understood, learned, used and attractive to user when used under specified conditions – easy to use



# IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY





## Non-Functional Testing

- ▶ **Efficiency Testing:** The process of testing to determine the efficiency of a software product
- ▶ The capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions



## Non-Functional Testing

- ▶ **Maintainability Testing:** The process of testing to determine the maintainability of the software product
- ▶ The ease with which a software product can be modified to meet new requirements, modified to make future maintenance easier, or adapted to changed environment





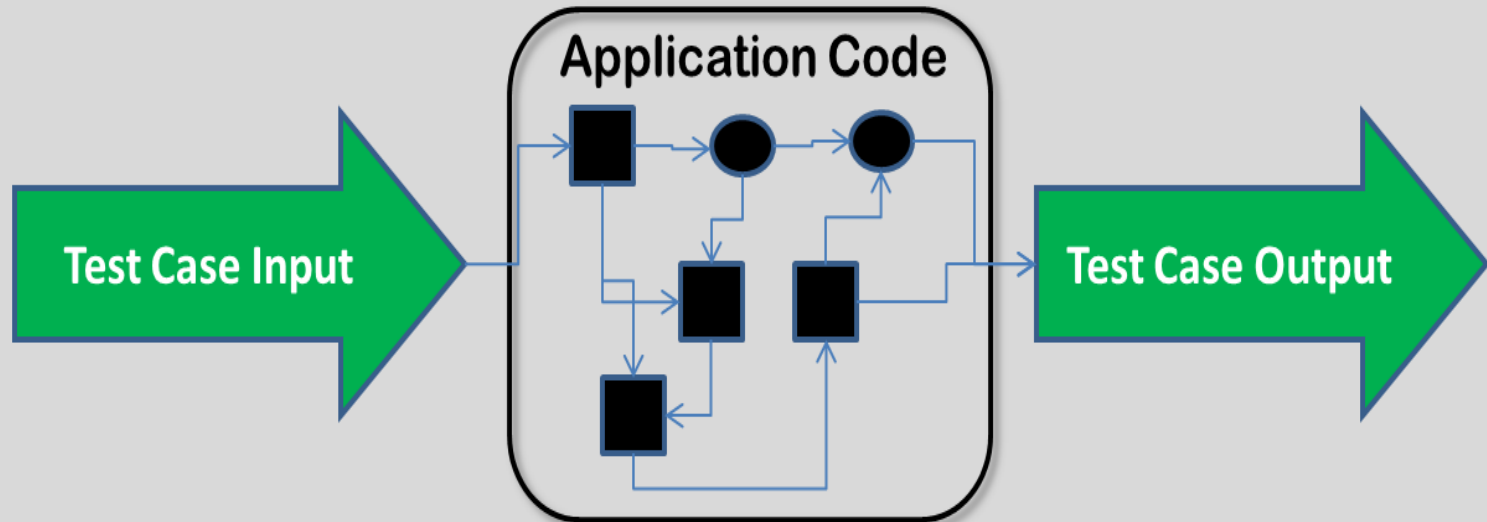
## Non-Functional Testing

- ▶ **Portability Testing:** The process of testing to determine the portability of a software product
- ▶ The ease with which the software product can be transferred from one hardware or software environment to another



## Structural Testing (White-Box)

- ▶ Testing is based on an analysis of the internal structure of the component or system (code coverage)
- ▶ Procedures to derive and/or select test cases based on an analysis of the internal structure of a component or system



**WHITE BOX TESTING APPROACH**



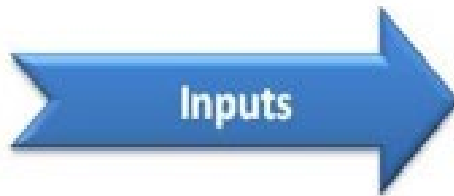
## Black-Box Testing (specification based)

- ▶ Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure
- ▶ Requirement based testing

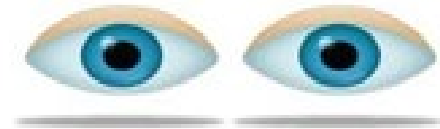


## Black Box Testing

Requirements Document



Validate output





## Confirmation Testing (Re-Testing)

- ▶ Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions
- ▶ It's a verification of fixed issues
- ▶ It needs to be executed in the same manner as it was initially executed

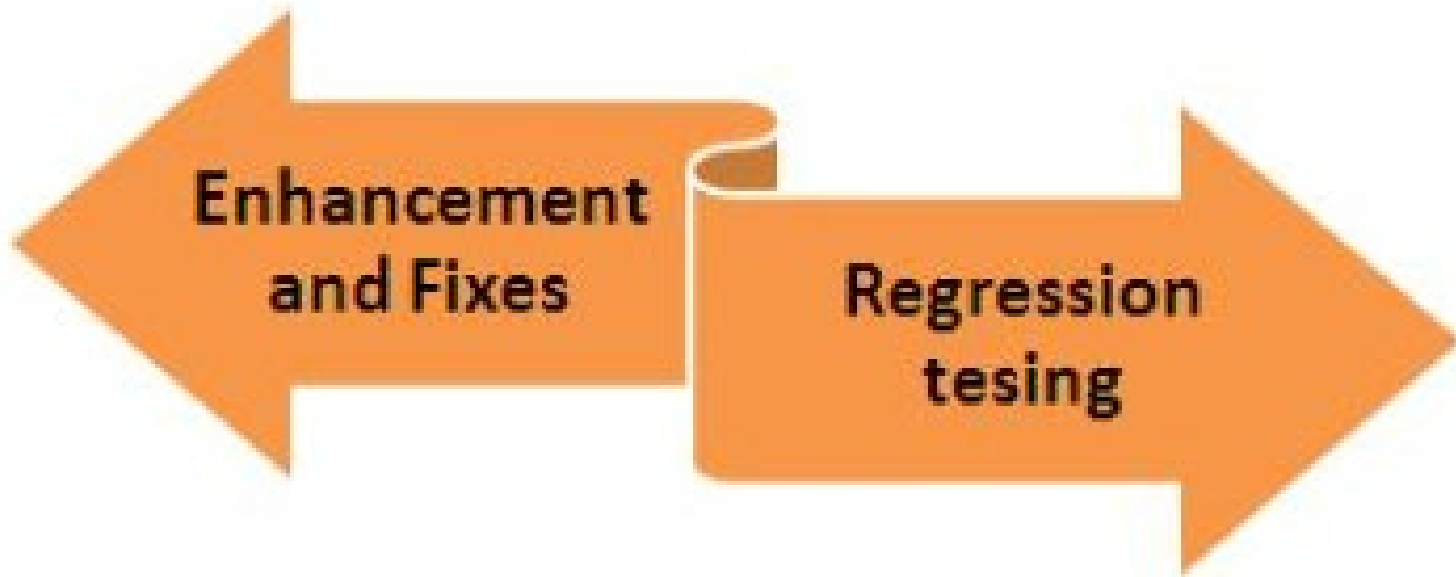


## Regression Testing

- ▶ Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed



# Regression Testing







## 2.4 Maintenance Testing

- ▶ Testing the changes to an operational system or the impact of a changed environment to an operational system
- ▶ Modifications from planned enhancement changes a minor release includes bug fixes or urgent emergency changes



## Impact Analysis & Regression Testing

- ▶ The assessment of change to the layers of development documentation, test documentation and components, in order to implement a given change to specified requirements



## Triggers for Maintenance Testing

- ▶ **Planned modifications:**
- ▶ New function or enhance performance
- ▶ Adaptive modifications – environmental changes (new hardware, software, system or legislation)
- ▶ Corrections of defects



## Ad-hoc Corrective Modifications

- ▶ Ad-hoc corrective modifications are required for immediate solution (production issue)
- ▶ Testing carried out informally; no formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity



## 2. Testing Throughout the Software Life Cycle – Question 1

- ▶ 1 What are good practices for testing within the development life cycle?
- ▶ a. Early test analysis and design.
- ▶ b. Different test levels are defined with specified objectives.
- ▶ c. Testers will start to get involved as soon as coding is done.
- ▶ d. A and B above.



## 2. Testing Throughout the Software Life Cycle – Question 2

- ▶ Which option best describes objectives for test levels with a life cycle model?
- ▶ a. Objectives should be generic for any test level.
- ▶ b. Objectives are the same for each test level.



## 2. Testing Throughout the Software Life Cycle – Question 2 – answers

- ▶ c. The objectives of a test level don't need to be defined in advance.
- ▶ d. Each level has objectives specific to that level.



## 2. Testing Throughout the Software Life Cycle – Question 3

- ▶ Which of the following is a test type?
- ▶ a. Component testing
- ▶ b. Functional testing
- ▶ c. System testing
- ▶ d. Acceptance testing





## 2. Testing Throughout the Software Life Cycle – Question 4

- ▶ Which of the following is a non-functional quality characteristic?
- ▶ a. Feasibility
- ▶ b. Usability
- ▶ c. Maintenance
- ▶ d. Regression



## Questions - 2. Testing Throughout the Software Life Cycle – Question 5

- ▶ Which of these is a functional test?
- ▶ a. Measuring response time on an on-line booking system.
- ▶ b. Checking the effect of high volumes of traffic in a call-center system.



## 2. Testing Throughout the Software Life Cycle – Question 5 - answers

- ▶ c. Checking the on-line bookings screen information and the database contents against the information on the letter to the customers.
- ▶ d. Checking how easy the system is to use.



## 2. Testing Throughout the Software Life Cycle – Question 6

- ▶ Which of the following is a true statement regarding the process of fixing emergency charges?
- ▶ a. There is no time to test the change before it goes live, so only the best developers should do this work and should not involve testers as they slow down the process.



## 2. Testing Throughout the Software Life Cycle – Question 6 – answers

- ▶ b. Just run the retest of the defect actually fixed.
- ▶ c. Always run a full regression test of the whole system in case other parts of the whole system have been adversely affected.



## 2. Testing Throughout the Software Life Cycle – Question 6 - answers

- ▶ d. Retest the changed area and then use risk assessment to decide on a reasonable subset of the whole regression test to run in case other parts of the system have been adversely affected.



## 2. Testing Throughout the Software Life Cycle – Question 7

- ▶ A regression test:
  - ▶ a. Is only run once.
  - ▶ b. Will always be automated.
  - ▶ c. Will check unchanged areas of the software to see if they have been affected.
  - ▶ d. Will check changed area of the software to see if they have been affected.



## 2. Testing Throughout the Software Life Cycle – Question 8

- ▶ Non-functional testing includes:
  - ▶ a. Testing to see where the system does not function correctly.
  - ▶ b. Testing the quality attributes of the system including reliability and usability.
  - ▶ c. Gaining user approval for the system





## 2. Testing Throughout the Software Life Cycle – Question 8 - answer

- ▶ d. Testing a system feature using only the software required for that function.



## 2. Testing Throughout the Software Life Cycle – Question 9

- ▶ Beta testing in:
- ▶ a. Performed by customers as their own site.
- ▶ b. Performed by customers at the software developer's site.
- ▶ c. Performed by an independent test team.



## 2. Testing Throughout the Software Life Cycle – Question 9

- ▶ d. Useful to test software developed for a specific customer or user.