Unit 307
Mobile and Operating Systems

1.3 End to End Testing

- Describe the end to end testing of platforms
  - Test Selection
  - Test Plan
  - Test Execution
  - Analyse Results
  - Corrective Action

Testing

- What is Testing?
  - Evaluating a system
    - Against specified requirements
    - Identify errors
    - Missing requirements
- Carried out by test team
  - QA
  - Tester
  - Quality Insurance Engineer

Testing

- Start as soon as the requirements are defined
- When should testing stop?
  - Deadlines
  - Completion of tests
  - Error rates below certain level
  - Management Decision

Testing – Verification and Validation

Verification	Validation
Are you building it right?	Are you building the right thing?
Ensures the system meets all the functionality	Ensures that the system meets the intended behaviour
Takes place first	After verification – checks overall system
By the System implementer	By the System testers
Objective and no subjective decisions	Subjective process and needs subjective decisions/

Testing

- Interact with system as end user
  - Test plans
  - Test cases
  - Test scenarios

Testing Methods

- Black Box Testing
  - No knowledge of interior working or system architecture
  - Interact with systems user interface
  - Provide inputs and examine outputs
- White Box Testing
  - Tester has knowledge of inner workings
  - Interact with parts of the system
  - May have to use specialized equipment for stages

Testing Levels – Functional Testing

- Functional Testing
  - Uses system specification
  - Provide inputs and examine output
  - Uses a complete system
- Unit Testing
  - Tests a part of the system
  - Unique data input
  - Verify each part of the system

Testing Levels – Functional Testing

- Integration Testing
  - Combination of units within the system
  - Bottom Up
    - Unit Testing first
    - Increasingly adds units
  - Top Down
    - Tests groups of units first
    - Progresses to the parts of the system

Testing Levels – Functional Testing

- System Testing
  - Complete System
  - Rigorous Tests
  - Verify system meets functional and technical specifications
  - Very close setup to how the system will be deployed

Testing Levels – Functional Testing

- Regression Testing
  - Whenever changes to system are made
  - Changes can affect other parts of the system
  - Important part of the test process
- Acceptance Testing
  - Most important as satisfies the customers requirements
  - Pre written scenarios
  - Determines how the system will work when deployed

Testing Levels – Non Functional Testing

- Non Functional Testing
  - Requirements that are non functional in nature
    - Performance
    - Security
    - User Interfaces
- Performance Testing
  - Identify bottlenecks
  - Network delays
  - Database connections
  - Load balancing

Testing Levels – Non Functional Testing

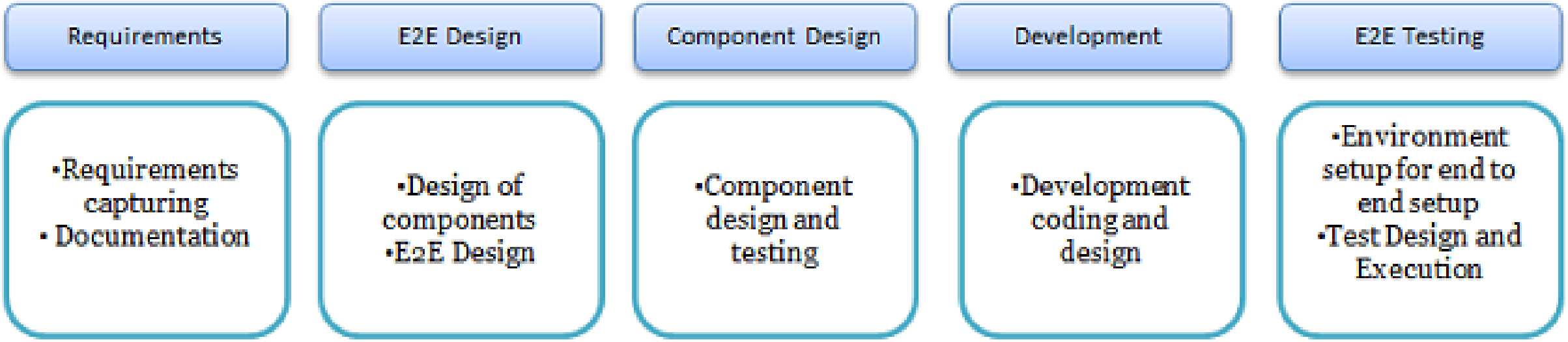
- Load Testing
  - How the system works under stress
  - Large Data Input
  - Often automated
- Stress Testing
  - Behaviour under abnormal conditions
  - Remove some resources
  - Try to break it and see how it recovers

Testing Levels – Non Functional Testing

- Usability
- User Interfaces
- Security
  - Confidentially
  - Integrity
  - Authentication
  - Availability
  - Authorisation

End to End Testing (e2e)

- Validates the complete system
- Checks integration with external interfaces



End to End Testing (e2e)

- Study of end to end testing requirements
- Test Environment setup and hardware/software requirements
- Describe all the systems and its subsystems processes.
- Description of roles and responsibilities for all the systems
- Testing methodology and standards
- End to end requirements tracking and designing of test cases
- Input and output data for each system

End to End Testing (e2e)

- End to End Testing Design framework consists of three parts
  1. Build user functions
  2. Build Conditions
  3. Build Test Cases

End to End Testing (e2e) – Build User Functions

- List down the features of the system and their interconnected components
- List the input data, action and the output data for each feature or function
- Identify the relationships between the functions
- Determine whether the function can be reusable or independent

End to End Testing (e2e) – Build User Functions

Consider a scenario where you login into your bank account and transfer some money to your friends account

1. Login into the banking system
2. Check for the balance amount in the account
3. Transfer some amount from your account to your friends account
4. Check the your latest account balance
5. Logout of the application

End to End Testing (e2e) – Build Conditions

- Building a set of conditions for each user function defined
- Conditions include sequence, timing and data conditions
- Use the Requirements Specification document
- Think about possible scenarios

End to End Testing (e2e) – Build Conditions

Login Example

- Invalid User Name and Password
- Valid user name and password
- Password strength checking
- Checking of error messages

Check Balance

- Check the current balance after transfer
- Check for the error message if the transfer amount is greater than the current balance amount

End to End Testing (e2e) – Build Test Scenario

A Test Scenario Example

- Login into the system
- Check of bank balance amount
- Transfer the bank balance amount

Build multiple test cases for each scenario identified.

Test scenarios are vague and cover a wide range of possibilities

End to End Testing (e2e) – Build Test Cases

- Testing is all about being very specific – need test cases
- A Test Case is a set of actions executed to verify a particular feature or functionality of the system
- Test Scenario - Login into the system
  - Test Case 1: Check results on entering valid User Id & Password
  - Test Case 2: Check results on entering Invalid User ID & Password
  - Test Case 3: Check response when User ID is Empty & Login Button is pressed
  - etc

End to End Testing (e2e) – Build Test Cases

- Test cases need to specify input conditions – test data
- Test data – needs to be recorded for test to be repeated
- Define expected result – again documented
- Specify any pre conditions – things that must be in place for the test
- Specify any post conditions – things that must happen as a result of the test

End to End Testing (e2e) – Build Test Cases

- The description of what requirement is being tested
- The explanation of how the system will be tested
- The test setup like: version of application under test, software, data files, operating system, hardware, security access, physical or logical date, time of day, prerequisites such as other tests and any other setup information pertinent to the requirements being tested
- Inputs and outputs or actions and expected results
- Any proofs or attachments
- Test Case should not be more than 15 steps

End to End Testing (e2e) – Build Test Cases

- Test Cases need to be simple and transparent
- Create Test Case with End User in mind
- Avoid Test Case repetition
- Do not Assume – specify!
- Make each test identifiable
- Each test must be repeatable

End to End Testing (e2e) – Build Test Cases

- Example test case document:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass / Fail
TU01	Check Customer Login with valid Data	Enter UserId Enter Password Click Submit	Userid = mike Password = mike99	User should Login into application		
TU02	Check Customer Login with invalid Data	Enter UserId Enter Password Click Submit	Userid = mike Password = mike22	User should not Login into application		

End to End Testing (e2e)

- Analyse Results
  - Certain % failed
  - Critical tests failed

- Corrective Action
  - Use the results to correct system
  - Retest