



Ripon Al Wasim

[riponalwasim@gmail.com](mailto:riponalwasim@gmail.com)

01712550093



# Course Outline

- Chapter 1: Fundamentals of Testing
- Chapter 2: Testing Throughout the Software Development Lifecycle
- Chapter 3: Static Testing
- Chapter 4: Test Techniques
- Chapter 5: Test Management
- Chapter 6: Tool Support for Testing



# Course Outline

- Chapter 1: Fundamentals of Testing

□ No. of Session: 02

❖ Session 01: 1.1 - 1.3

❖ Session 02: 1.4 - 1.5.2





# Chapter 1: Fundamentals of Testing

## ***Session 01***

### 1.1 What is Testing?

#### 1.1.1 Typical Objectives of Testing

#### 1.1.2 Testing and Debugging

### 1.2 Why is Testing Necessary?

#### 1.2.1 Testing's Contributions to Success



# Chapter 1: Fundamentals of Testing

## ***Session 01 (continued)***

1.2.2 Quality Assurance and Testing

1.2.3 Errors, Defects, and Failures

1.2.4 Defects, Root Causes and Effects

1.3 Seven Testing Principles





# Chapter 1: Fundamentals of Testing

## ***Session 02***

### 1.4 Test Process

#### 1.4.1 Test Process in Context

#### 1.4.2 Test Activities and Tasks

#### 1.4.3 Test Work Products



# Chapter 1: Fundamentals of Testing

## ***Session 02 (continued)***

1.4.4 Traceability between the Test Basis and Test Work Products

1.5 The Psychology of Testing

1.5.1 Human Psychology and Testing

1.5.2 Tester's and Developer's Mindsets





# Chapter 1: Fundamentals of Testing

## What is Testing?

Verify and validate application quality to ensure it meets user requirements.

**ISTQB Definition:** The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.



# Chapter 1: Fundamentals of Testing

## Typical Objectives of Testing /1

- To **prevent defects** by evaluate work products such as requirements, user stories, design, and code
- To **verify** whether all specified requirements have been fulfilled
- To **check** whether the test object is complete and validate if it works as the users and other stakeholders expect
- To **build confidence** in the level of quality of the test object

# Chapter 1: Fundamentals of Testing

## Typical Objectives of Testing /2

- To find defects and failures thus reduce the level of risk of inadequate software quality
- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object
- To comply with contractual, legal, or regulatory requirements or standards, and/or to verify the test object's compliance with such requirements or standards



# Chapter 1: Fundamentals of Testing

## Testing and Debugging

Testing

Debugging

Correcting  
defects

Retesting

- ❑ Testing & Retesting/Confirmation testing are testing role
- ❑ Debugging & Correcting defects are development role
- ✓ Testing is conducted to verify a software system's functionality, performance, and reliability to identify defects or errors.
- ✓ Debugging is investigating and resolving those defects, aiming to eliminate issues and ensure smooth operation.

Source: [Browserstack](#)

Ques: Is Testing debugging?



# Chapter 1: Fundamentals of Testing

## Why is Testing Necessary? /1

Testing is necessary because we all make mistakes.

Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong - humans make mistakes all the time - it is what we do best!



# Chapter 1: Fundamentals of Testing

## Why is Testing Necessary? /2

Some reasons why Software Testing is important

1. Helps in saving money
2. Security
3. Quality of the product
4. Satisfaction of the customer
5. Enhancing the development process
6. Easy while adding new features
7. Determining the performance of the software

Source: <https://www.indiumsoftware.com/blog/why-software-testing/>





# Chapter 1: Fundamentals of Testing

## Testing's Contributions to Success



- ✓ Involvement of testers in requirement reviews and user story refinement
- ✓ Testers working closely with the system designers
- ✓ Having testers work closely with developers
- ✓ Testers verifying and validating the software before its release

Source:

<https://www.toolsqa.com/software-testing/istqb/why-is-testing-necessary/#:~:text=Let's%20start%20by%20considering%20Why,make%20mistakes%20at%20any%20time.>



# Chapter 1: Fundamentals of Testing

## Quality Assurance and Testing /1

- ❑ Quality assurance (QA) systematically checks and rechecks a product's quality to ensure it lives up to its stated promises.
- ❑ Software testing is a portion of quality assurance that comprises running the program to check for flaws and malfunctions and also to see if it performs as anticipated.

Source:

<https://www.browserstack.com/guide/quality-assurance-vs-testing>

<https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/>

# Chapter 1: Fundamentals of Testing

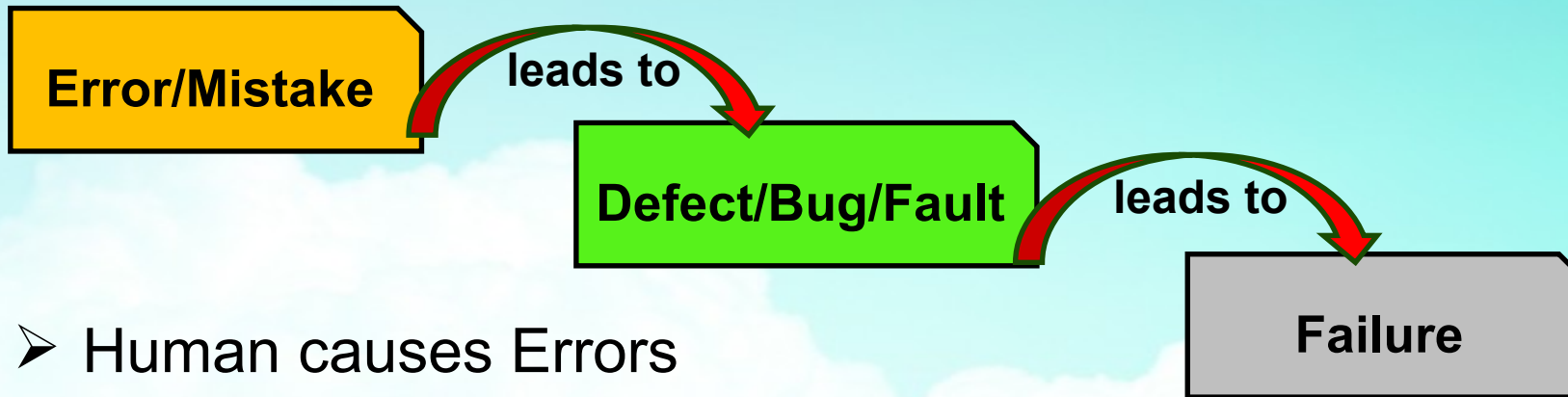
## Quality Assurance and Testing /2

	QA	Testing
Purpose	Prevent issues through establishing quality standards	Detect and fix quality issues
Focus	Development processes	Different aspects of the product: functionality, integrations, performance etc.
Who	External stakeholders, business analysts, QA engineers, software developers	QA engineers, software developers
When	Throughout the entire product development life cycle	At the testing stage or along the development process
Doing what	Introducing standards, creating guidelines, improving development processes	Reviewing code, running tests, addressing defects



# Chapter 1: Fundamentals of Testing

## Errors, Defects, and Failures /1



- Human causes Errors
- Error causes Defects
- Defect/Bug causes Failures



# Chapter 1: Fundamentals of Testing

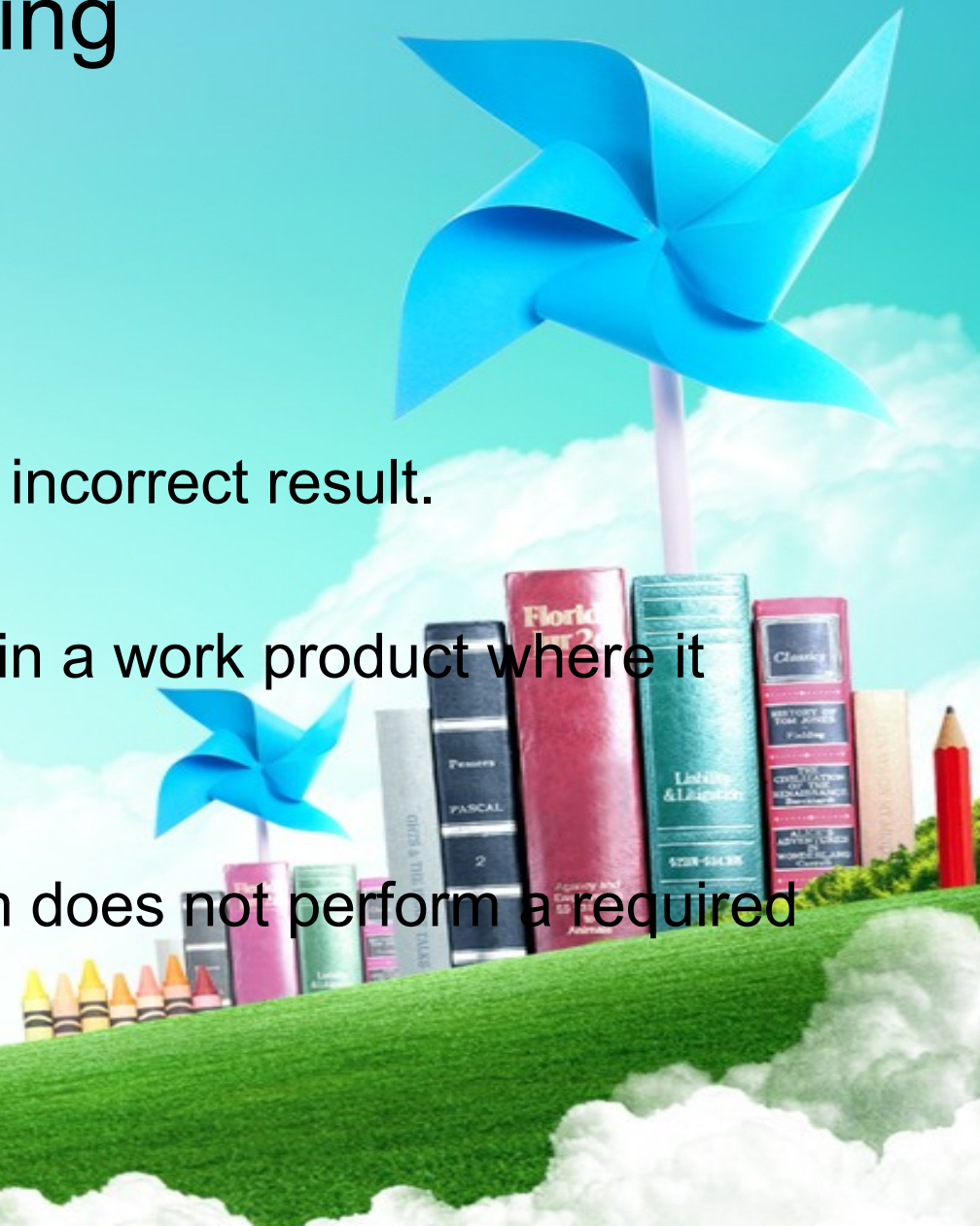
## Errors, Defects, and Failures /2

ISTQB Definition:

**Error** (mistake) A human action that produces an incorrect result.

**Defect** (bug, fault) An imperfection or deficiency in a work product where it does not meet its requirements or specifications.

**Failure** An event in which a component or system does not perform a required function within specified limits.



# Chapter 1: Fundamentals of Testing

## Defects, Root Causes and Effects /1

User story: As a user, I want to calculate & pay interest.

The product owner misunderstood how interest was to be calculated, so was unable to clearly specify what the interest calculation should have been.

Developer has written the code based on the description of user story - calculation is wrong. So the failure here is the incorrect interest calculations for customers.

Defect	Root cause	Effect
wrong calculation in the code	the product owner's lack of knowledge about how interest should be calculated	customer complaints



# Chapter 1: Fundamentals of Testing

## Defects, Root Causes and Effects /2

- ✓ A root cause is generally an organizational issue, whereas a cause for a defect is an individual action.

```
for(int i = 0; i < 10; i++) {  
    System.out.println("Value of i " + i);  
}
```



```
for(int i = 0; i > 10; i++) {  
    System.out.println("Value of i " + i);  
}
```



❑ Typical root causes of defects are:

- ❖ Unclear, Missing, or Wrong Requirements
- ❖ Logical Errors
- ❖ Data Errors
- ❖ Incorrect Design

- ❖ Environment Issue
- ❖ Time Pressure
- ❖ Carelessness



# Chapter 1: Fundamentals of Testing

## Seven Testing Principles /1

#	Testing Principle	Description
P-1	Testing shows presence of defects	Testing can show that defects are present, but cannot prove that there are no defects. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness
P-2	Exhaustive testing is impossible	Testing everything (all combinations of inputs and preconditions) is not feasible except for trivial cases. Instead of exhaustive testing, we use risks and priorities to focus testing efforts
P-3	Early testing	Testing activities should start as early as possible in the software or system development life cycle and should be focused on defined objectives
P-4	Defect clustering	A small number of modules contain most of the defects discovered during prerelease testing or show the most operational failures
P-5	Pesticide paradox	If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new bugs. To overcome this 'pesticide paradox', the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects
P-6	Testing is context dependent	Testing is done differently in different contexts. For example, safety-critical software is tested differently from an e-commerce site
P-7	Absence-of-errors fallacy	Finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations

# Chapter 1: Fundamentals of Testing

## Seven Testing Principles /2

#	Statement	Relation to Testing Principle
1	Is the software defect free?	Principle - Testing shows presence of defects
2	How much testing is enough?	Principle - Exhaustive testing is impossible
3	When can we meet our test objectives?	Principle - Early testing
4	Focusing on defects can help us plan our tests	Principle - Defect clustering
5	The defect clusters change over time	Principle - Pesticide paradox
6	Software systems context	Principle - Testing is context dependent
7	If we don't find defects does that mean the users will accept the software?	Principle - Absence of errors fallacy



**Thank you!**

**Happy learning..  
Learn more...**



# Chapter 1: Fundamentals of Testing

## Test Process in Context /1

The factors that influence the particular test process include the followings:

- Software development life cycle (SDLC) model and project methodologies being used. An Agile project developing **mobile apps** will have quite a different test process to an organization producing **medical devices such as pacemakers**.
- Test levels and test types being considered.
- Product and project risks.
- Business domain (e.g. mobile apps versus medical devices).



# Chapter 1: Fundamentals of Testing

## Test Process in Context /2

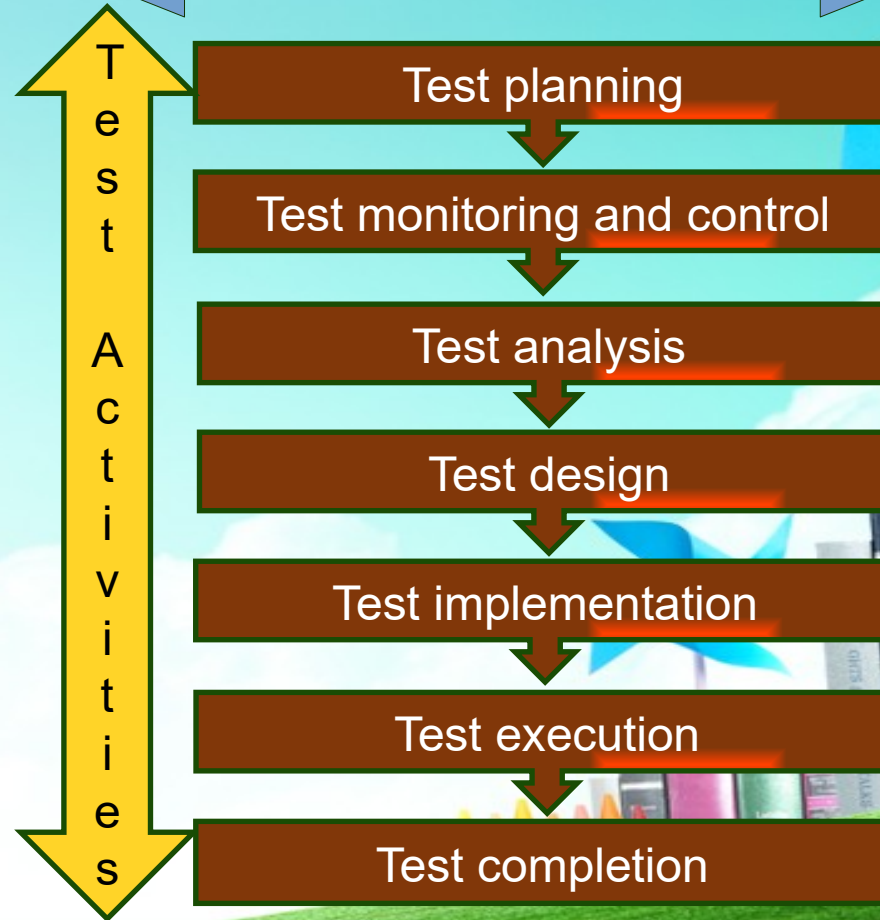
- Operational constraints, including:
  - budgets and resources
  - timescales
  - complexity
  - contractual and regulatory requirements.
- Organizational policies and practices.
- Required internal and external standards.



# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /1

Test Process depending on their context



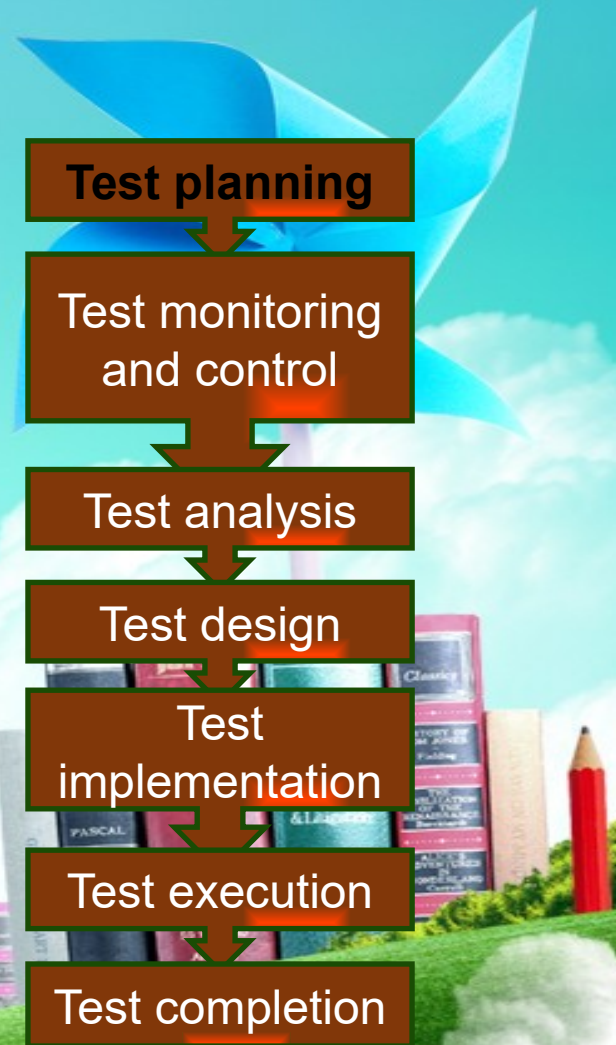


# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /2

### Test planning:

- ❑ Defining the objectives of testing and the approach for meeting those objectives within project constraints and contexts.
  - deciding on suitable test techniques to use
  - deciding what tasks need to be done
  - formulating a test schedule and other things
- ❑ Determining the scope and risk.
- ❑ Determining the required test resources (e.g. people, test environment, PCs)
- ❑ Scheduling test analysis and design tasks, test implementation, execution and evaluation.
- ❑ Determining the exit criteria of testing.



# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /3

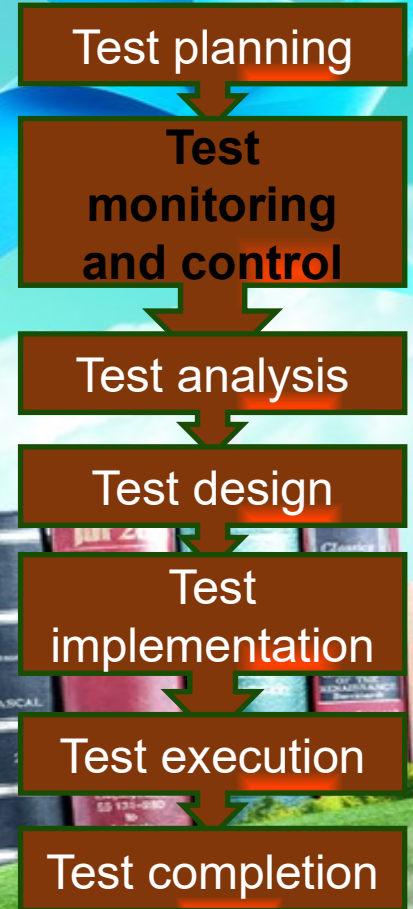
### Test monitoring and control:

#### ❖ test monitoring

- ☐ Compare actual progress against the plan.
- ☐ Check on the progress of test activities.
- ☐ Report the test status and any necessary deviations from the plan.

#### ❖ test control

- ☐ Taking whatever actions are necessary to meet the mission and objectives of the project, and/or adjust the plan.





# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /4

### Test monitoring and control

One way we can monitor test progress is by using exit criteria, also known as 'definition of done' in Agile development.

The exit criteria for test execution might include:

- Checking test results and logs against specified coverage criteria (we have not finished testing until we have tested what we planned to test).
- Assessing the level of component or system quality based on test results and logs (e.g. the number of defects found or ease of use).
- Assessing product risk and determining if more tests are needed to reduce the risk to an acceptable level.

# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /5

### Test analysis:

Test analysis addresses 'what to test'

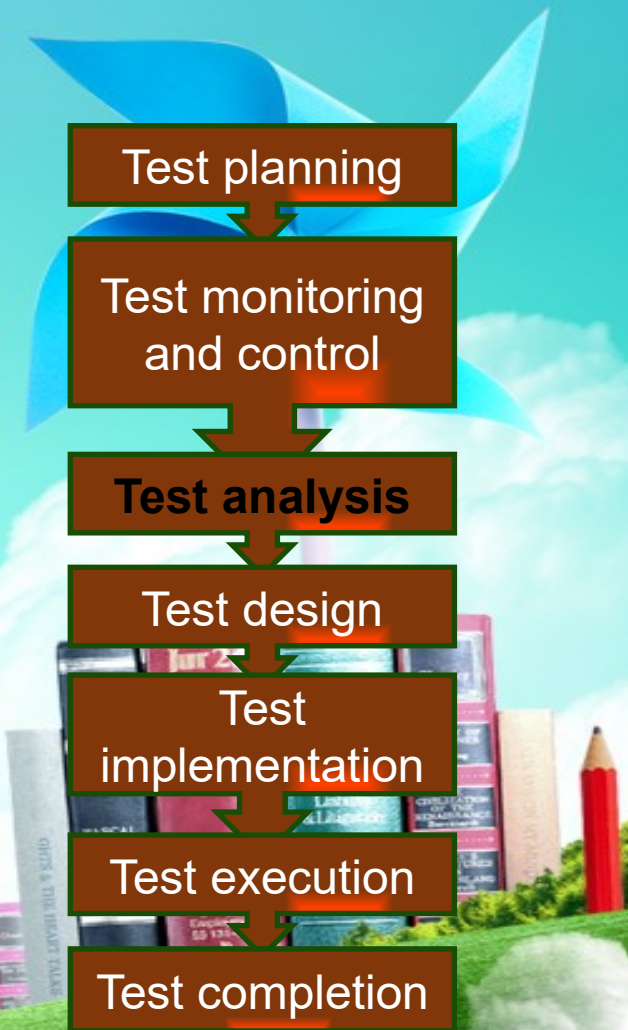
Test analysis includes the following major activities and tasks:

- ❑ Analyze the test basis appropriate to the test level being considered
- ❑ Evaluate the test basis and test items to identify various types of defects that might occur

- ambiguities
- omissions
- inconsistencies

- inaccuracies
- contradictions
- superfluous statements

- ❑ Identify features and sets of features to be tested
- ❑ Identify and prioritize test conditions for each feature
- ❑ Capture bi-directional traceability between each element of the test basis and the associated test conditions





# Chapter 1: Fundamentals of Testing

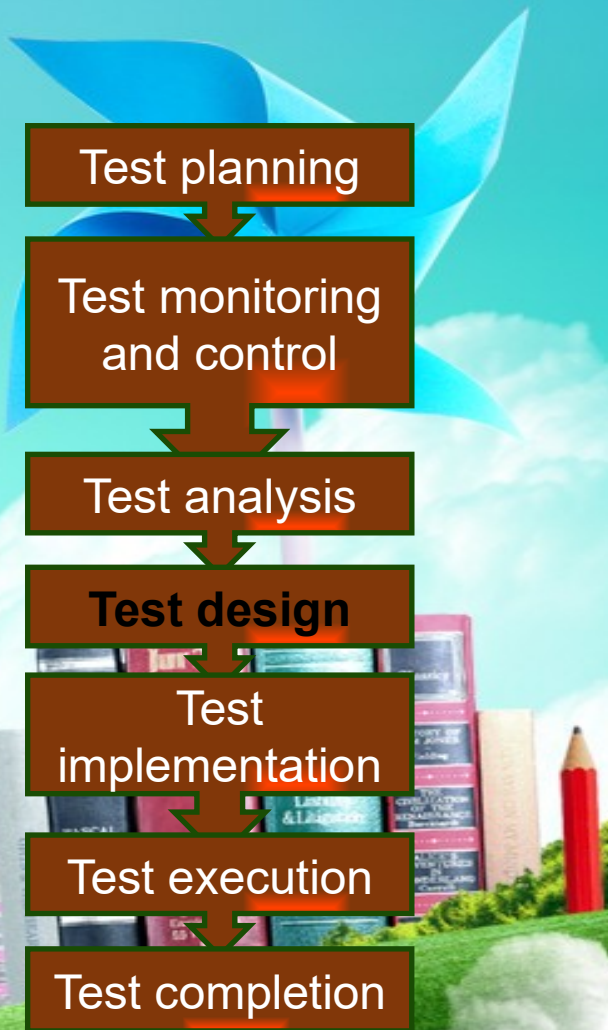
## Test Activities and Tasks /6

### Test design:

Test design address 'how to test'

Test design includes the following major activities:

- ❑ Design and prioritize test cases and sets of test cases
- ❑ Identify the necessary test data to support the test conditions and test cases as they are identified and designed
- ❑ Design the test environment, including set-up, and identify any required infra- structure and tools
- ❑ Capture bi-directional traceability between the test basis, test conditions, test cases and test procedures



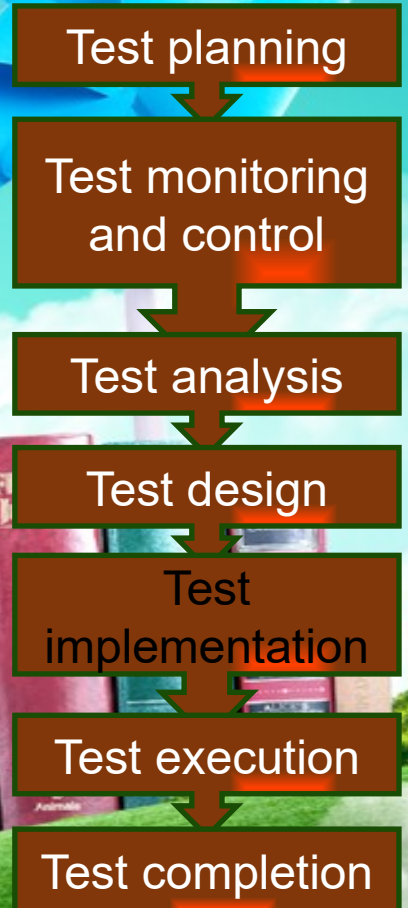
# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /7

### Test implementation:

Test implementation includes the following major activities:

- ☐ Develop and prioritize the test procedures and potentially create automated test scripts.
- ☐ Create test suites from the test procedures and automated test scripts (if any).
- ☐ Arrange the test suites within a test execution schedule in a way that results in efficient test execution.
- ☐ Build the test environment and verify that everything needed has been set up correctly.
- ☐ Prepare test data and ensure that it is properly loaded in the test environment.
- ☐ Verify and update the bi-directional traceability between the test basis, test conditions, test cases, test procedures and test suites.



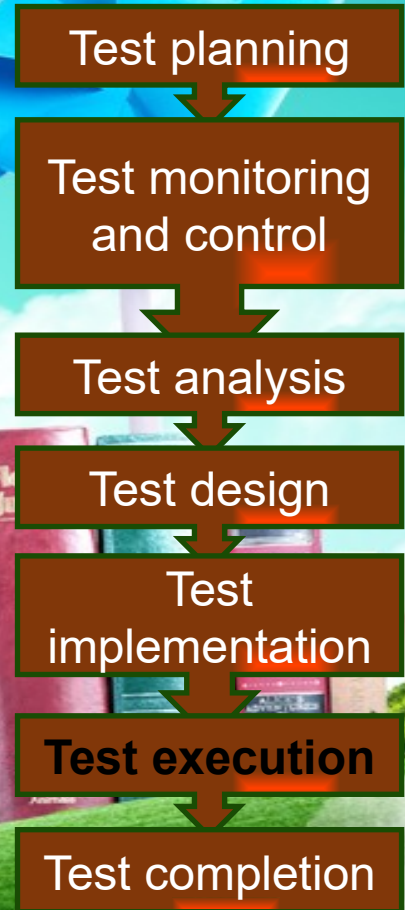


# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /8

### Test execution:

- ☐ Record the identities and versions of all of the test items, test objects, test tools and other testware.
- ☐ Execute the tests either manually or by using an automated test execution tool, according to the planned sequence.
- ☐ Compare actual results with expected results, observing where the actual and expected results differ.
- ☐ Analyze the anomalies in order to establish their likely causes.
- ☐ Report defects based on the failures observed.
- ☐ Log the outcome of test execution (e.g. pass, fail or blocked).



# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /9

### Test execution:

- ☐ Retest/Confirmation test (to confirm a fix).
- ☐ Regression test (to see whether defects have been introduced in unchanged areas of the software or to see whether a fixed defect now makes another defect apparent).
- ☐ Verify and update the bi-directional traceability between the test basis, test conditions, test cases, test procedures and test results.





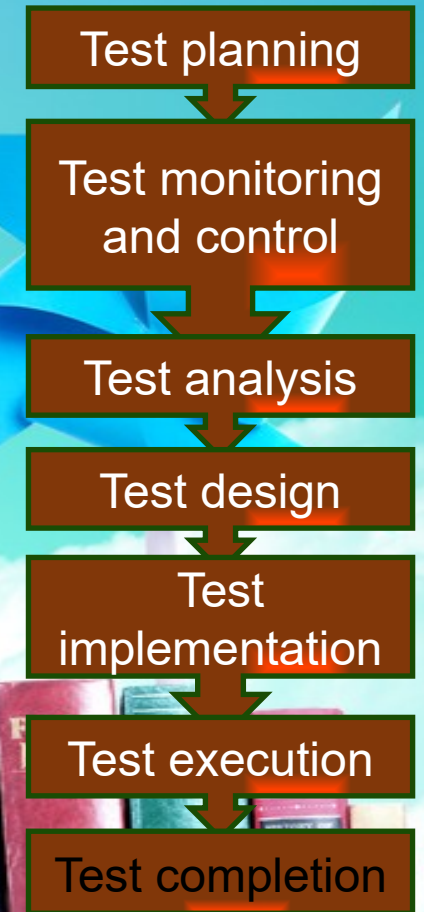
# Chapter 1: Fundamentals of Testing

## Test Activities and Tasks /10

### Test completion:

Test completion includes the following major activities:

- ☐ Check whether all defect reports are closed.
- ☐ Create a test summary report to be communicated to stakeholders.
- ☐ Finalize and archive the test environment, the test data, the test infrastructure and other testware for later reuse.
- ☐ Hand over the testware to the maintenance teams, other project teams, and/or other stakeholders who could benefit from its use.
- ☐ Analyze lessons learned from completed test activities to determine changes needed for future iterations, releases and projects (i.e. perform a retrospective).
- ☐ Use the information gathered to improve test process maturity, especially as an input to test planning for future projects.



# Chapter 1: Fundamentals of Testing

## Test Work Products

Test work products are created as part of the test process, and there is significant variation in the types of work products created.

Test Work Products	Test Activities
Test planning work products	Test planning
Test monitoring and control work products	Test monitoring and control
Test analysis work products	Test analysis
Test design work products	Test design
Test implementation work products	Test implementation
Test execution work products	Test execution
Test completion work products	Test completion



# Chapter 1: Fundamentals of Testing

## Traceability between the Test Basis and Test Work Products /1

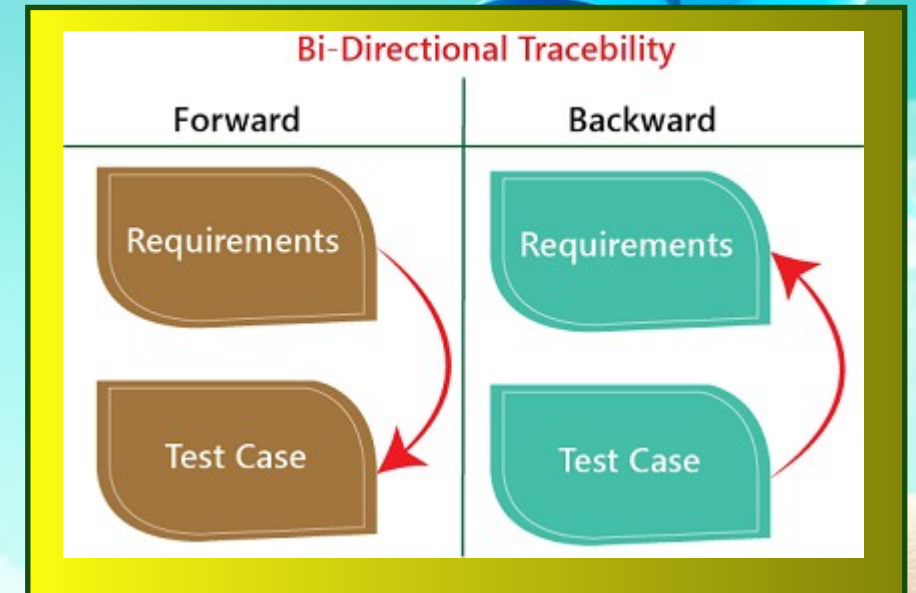
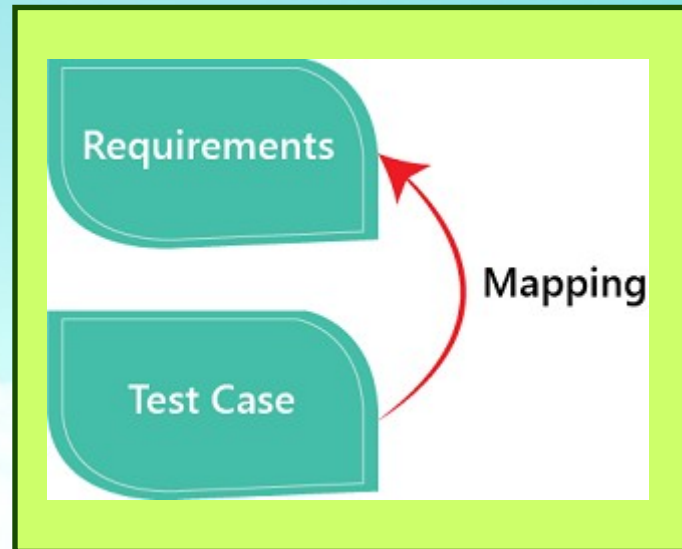
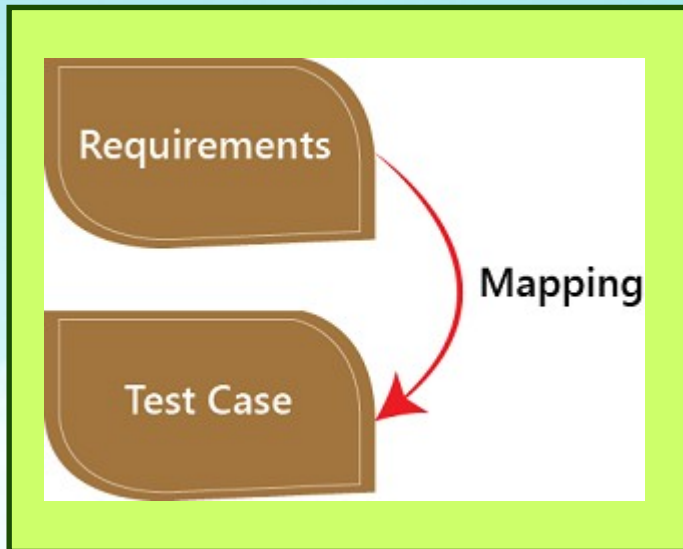
### RTM Template

REQUIREMENT TRACEABILITY MATRIX												
Project Name :												
Created By :												
Date :												
Test case IDs → Req. IDs ↓	Req. desc.	Req. type	Test case no. 1	Test case no. 2	Test case no. 3	Test case no. 4	Test case no. 5	Test case no. 6	Test case no. 7	Test case no. 8	Test case no. 9	Test case no. 10
Requirement no. 1			✓						✓			
Requirement no. 2				✓								✓
Requirement no. 3										✓		
Requirement no. 4					✓			✓				✓
Requirement no. 5											✓	
Requirement no. 6						✓						
Requirement no. 7									✓			
Requirement no. 8							✓				✓	
Requirement no. 9										✓		
Requirement no. 10						✓						

Source: <https://www.geeksforgeeks.org/requirement-traceability-matrix/>

# Chapter 1: Fundamentals of Testing

## Traceability between the Test Basis and Test Work Products /2



3 types of Traceability matrix:

- Forward traceability
- Backward or reverse traceability
- Bi-directional traceability
- Source: <https://www.javatpoint.com/traceability-matrix>



# Chapter 1: Fundamentals of Testing

## Traceability between the Test Basis and Test Work Products /3

Good traceability also supports the following:

- ☐ analyzing the impact of changes, whether to requirements or to the component or system
- ☐ making testing auditable, and being able to measure coverage
- ☐ meeting IT governance criteria (where applicable)
- ☐ improving the coherence of test progress reports and test summary reports to stakeholders, as described above
- ☐ relating the technical aspects of testing to stakeholders in terms that they can understand
- ☐ providing information to assess product quality, process capability and project progress against business goals

# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### Human Psychology and Testing /1

- Identifying defects may unfortunately be perceived in **development** as a **criticism**
- Finding defects in **testing** should be **constructive criticism**

Some of the factors are:

**Confirmation Bias:** Confirmation bias is the tendency to search for, interpret, favor, and recall information in a way that confirms or supports one's prior beliefs or values. It is difficult for humans to become aware of errors committed by themselves. For example, generally, developers have a confirmation bias that makes it difficult to accept that their code is incorrect.



# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### Human Psychology and Testing /2

**Cognitive Bias:** A cognitive bias is a systematic pattern of deviation from norm or rationality in judgment. Individuals create their own "subjective reality" from their perception of the input. This may make it difficult for people to understand or accept information produced by testing.

**Blame the bearer:** It is a common human trait to blame the bearer of bad news, and information produced by testing often contains bad news.

**Testing as Criticism:** Some people may perceive testing results as a criticism of the product and its author. For example, when testers identify defects during testing, these defects may be seen as a criticism of the product and author rather than the status of product quality.



# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### How to Overcome Psychological Barriers

**Provide information in a constructive way:** During static and dynamic testing, information about defects and failure should be provided in a constructive way to reduce the tension between testers, analysts, product owners, designers, or developers.

**Develop good interpersonal skills:** Testers and test managers must have good interpersonal skills to communicate effectively about test results, test progress, risks, defects, and failure. Interpersonal skills are required to build a positive relationship with different team members.



# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### How to Communicate Well

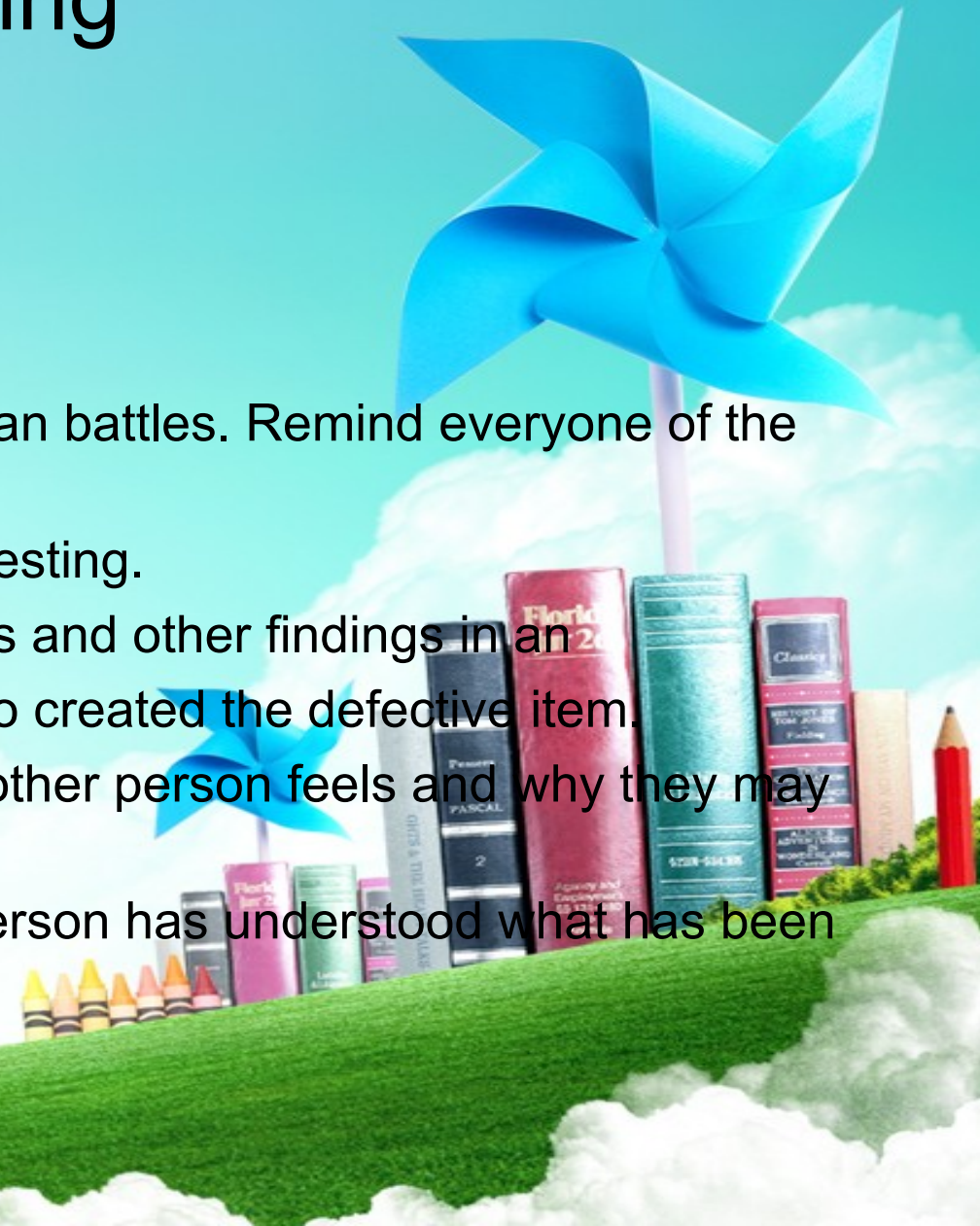
**Start with Collaboration:** Start with collaboration rather than battles. Remind everyone of the common goal of better quality systems.

**Emphasize Testing benefits:** Highlight the usefulness of testing.

**Communication of Test results:** Communicate test results and other findings in an unbiased, fact-focused way without blaming the person who created the defective item.

**Understand how people feel:** Try to understand how the other person feels and why they may react negatively to the information.

**Confirmation of understanding:** Confirm that the other person has understood what has been said and vice versa.



# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### Tester's and Developer's Mindsets /1

Different mindsets, same goal

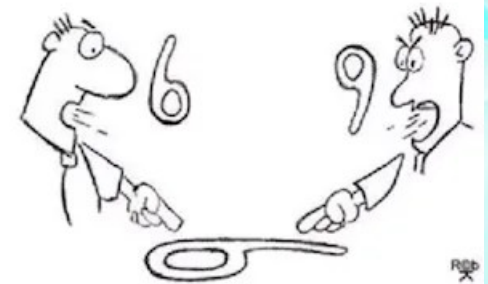
=> A developer will ask: "What do I need to build, how should I do it?"

=> A tester will ask: "What can go wrong? What can I do to break the application or find the weaknesses?"

**Mindset of a Tester:** A tester's mindset should include

- Curiosity
- Professional pessimism
- A critical eye
- Attention to detail
- Experience
- Good communication skills

Developer V/s Tester





# Chapter 1: Fundamentals of Testing

## The Psychology of Testing

### Tester's and Developer's Mindsets /2

**Mindset of a Developer:** A developer's mindset may include some elements of a tester's mindset.

Still, successful developers are often more interested in designing and building solutions than in contemplating what might be wrong with those solutions.

