IMT SCHOOL

Testing

Lecture 1

Software Engineering
The 7 Testing Principles
Why We Need Testing
Fundamental Test Process

The Generic Phases For The Projects Life Cycle

Definition

Requirements Analysis

Design

Coding

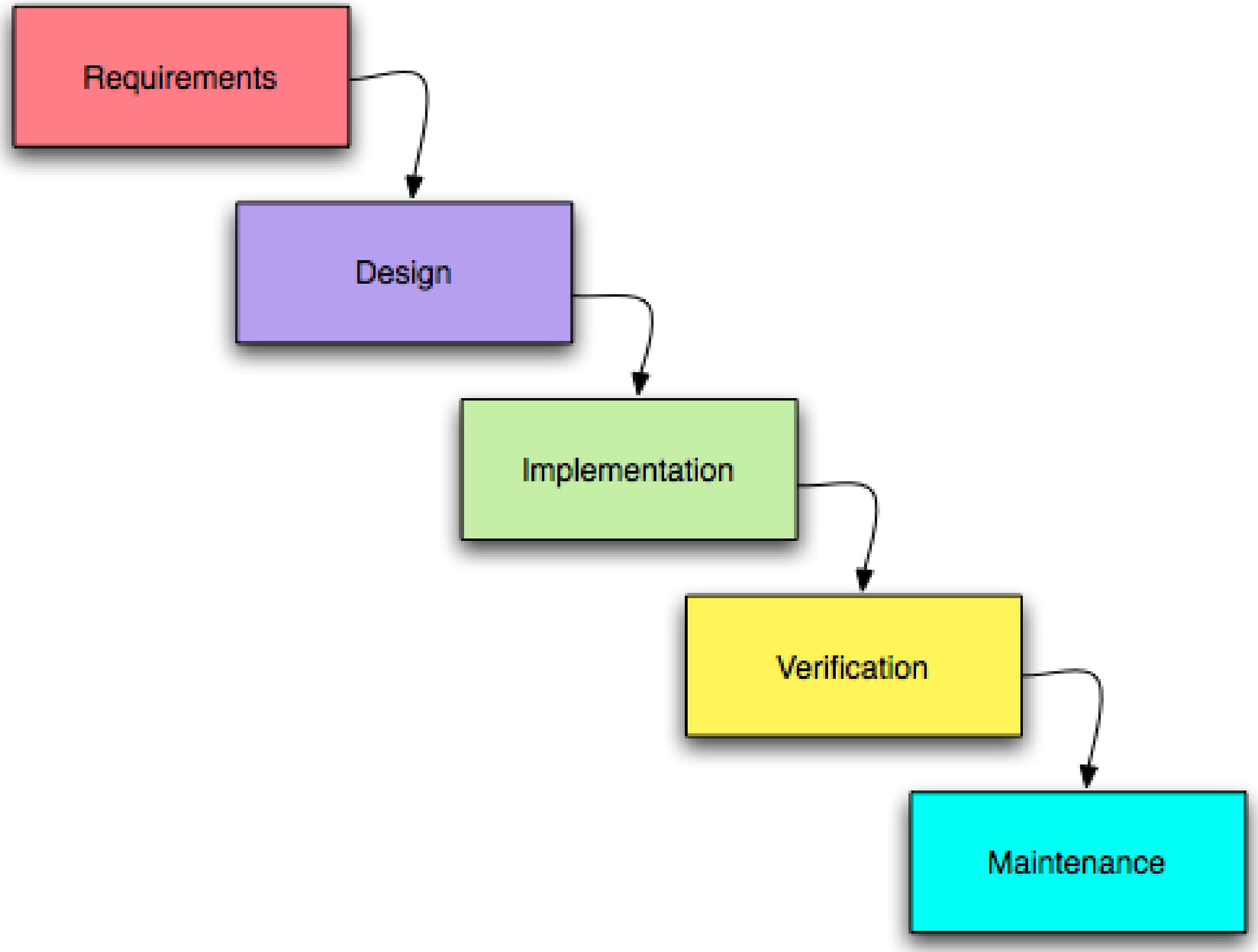
Testing

Software Engineering Models

- Most Known Software Development Plans
  - Water Fall
  - V-Model
  - Agile

Waterfall Model

Waterfall Model



Waterfall Model

❖Water Fall Advantages

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model
- Each phase has specific deliverables and a review process.

Waterfall Model

❖Water Fall Advantages

- In this model phases are processed and completed one at a time. Phases do not overlap.

- Waterfall model works well for smaller projects where requirements are very well understood.

Waterfall Model

❖Water Fall Disadvantages

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.

Waterfall Model

❖Water Fall Disadvantages

- Not a good model for complex and object-oriented projects.

- Poor model for long and ongoing projects.

- Not suitable for the projects where requirements are at a moderate to high risk of changing.

Waterfall Model

When to use the waterfall model

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.

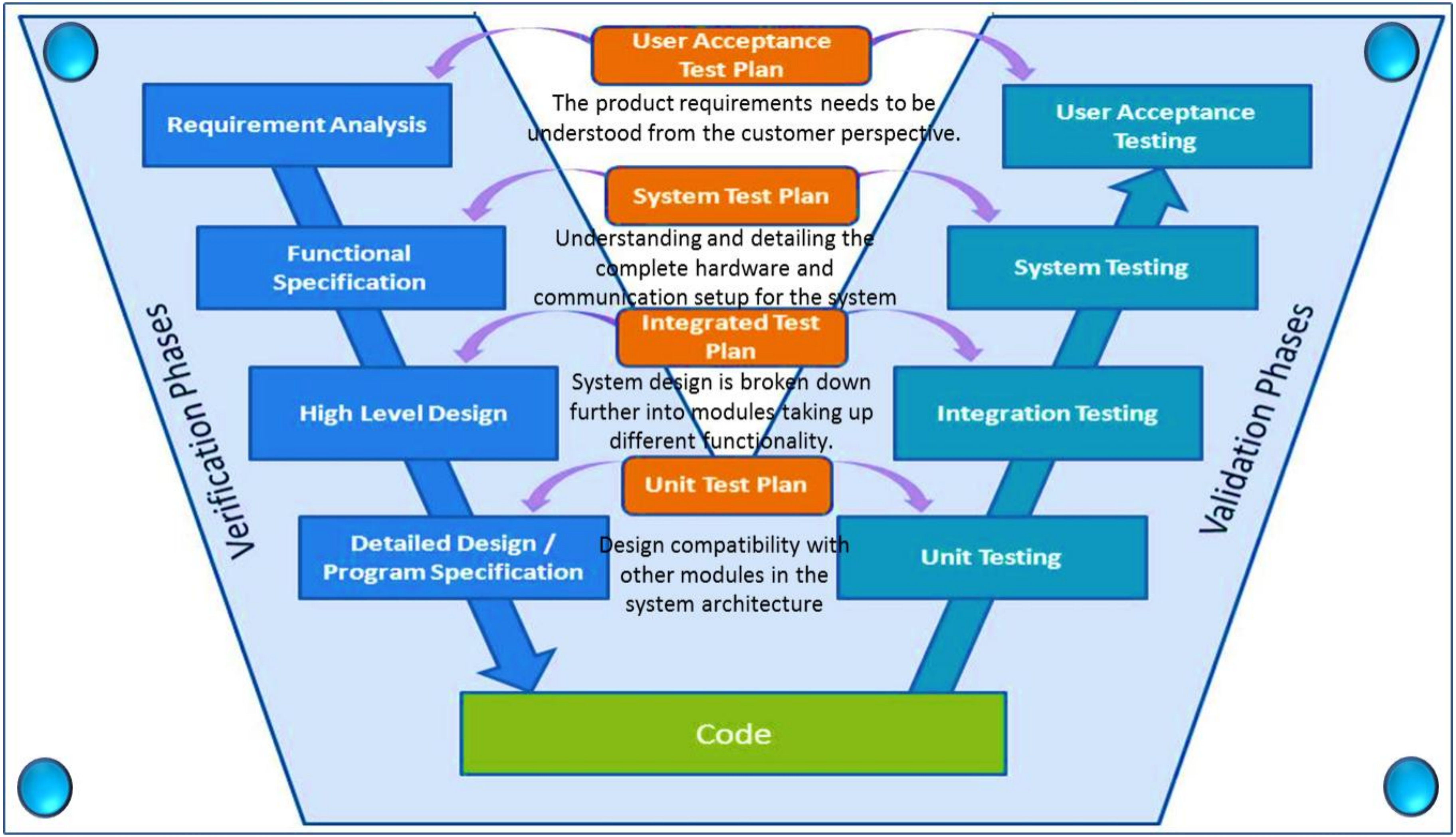
Waterfall Model

❖When to use the waterfall model

- There are no ambiguous requirements.
- Ample resources with required expertise are available freely.
- The project is short.

V - Model

V - Model



The product requirements needs to be understood from the customer perspective.

Understanding and detailing the complete hardware and communication setup for the system

System design is broken down further into modules taking up different functionality.

Design compatibility with other modules in the system architecture

V - Model

- V - Model Advantages

  - Simple and easy to use.

  - Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.

V - Model

- V - Model Advantages

  - Proactive defect tracking – that is defects are found at early stage.

  - Avoids the downward flow of the defects.

  - Works well for small projects where requirements are easily understood.

V - Model

❖ V - Model Disadvantages

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

V - Model

❖ When to use V - Model

- The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.

- The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

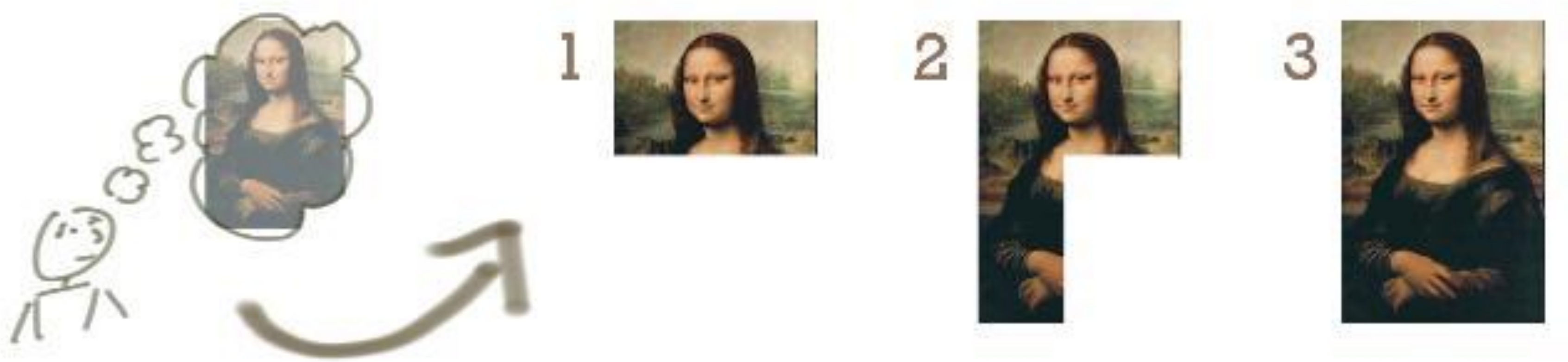
V - Model

❖When to use the waterfall model

- There are no ambiguous requirements.
- Ample resources with required expertise are available freely.
- The project is short.

Agile

Agile



Agile

❖ Advantages of Agile Model

- Customer satisfaction by rapid, continuous delivery of useful software.
- Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).

Agile

❖ Advantages of Agile Model

- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Agile

- Disadvantages of Agile Model

  - In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.

  - There is lack of emphasis on necessary designing and documentation.

Agile

❖ Disadvantages of Agile Model

- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

Agile

❖ When To Use Agile Model

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.

- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.

Agile

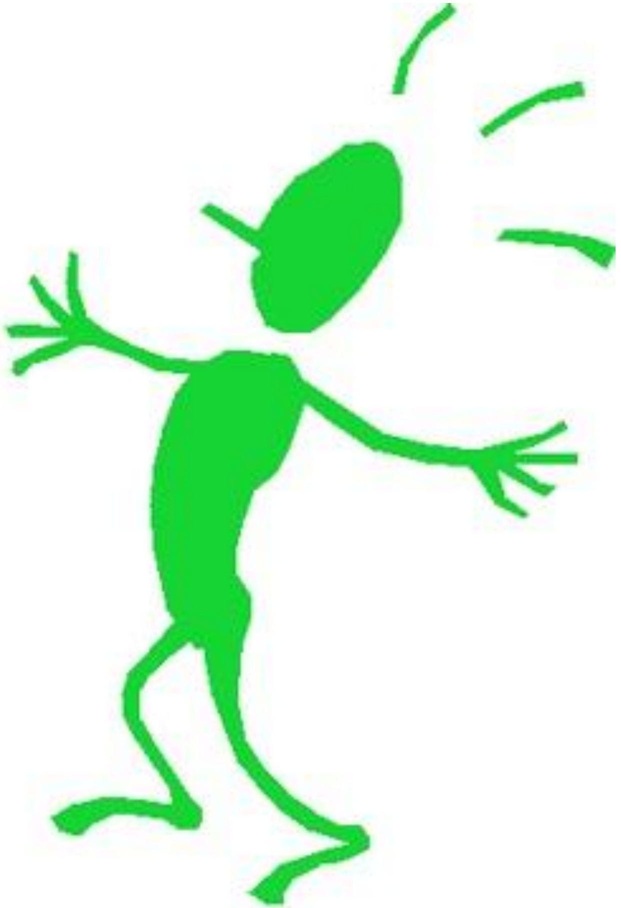
- Unlike the Water fall, in agile model very limited planning is required to get started with the project.

- Waterfall development completes the project-wide work-products of each discipline in one step before moving on to the next discipline in the next step.

- Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.

Any questions

… ?



IMT SCHOOL

Why Is Testing Necessary?

BUGS
BUGS EVERYWHERE

Why Is Testing Necessary …

- **Software Systems context**

  - Software systems are an integral part of life, Most of people have had an experience with software that did not work as expected.

Why Is Testing Necessary …

❖**What happens when a system go wrong?**

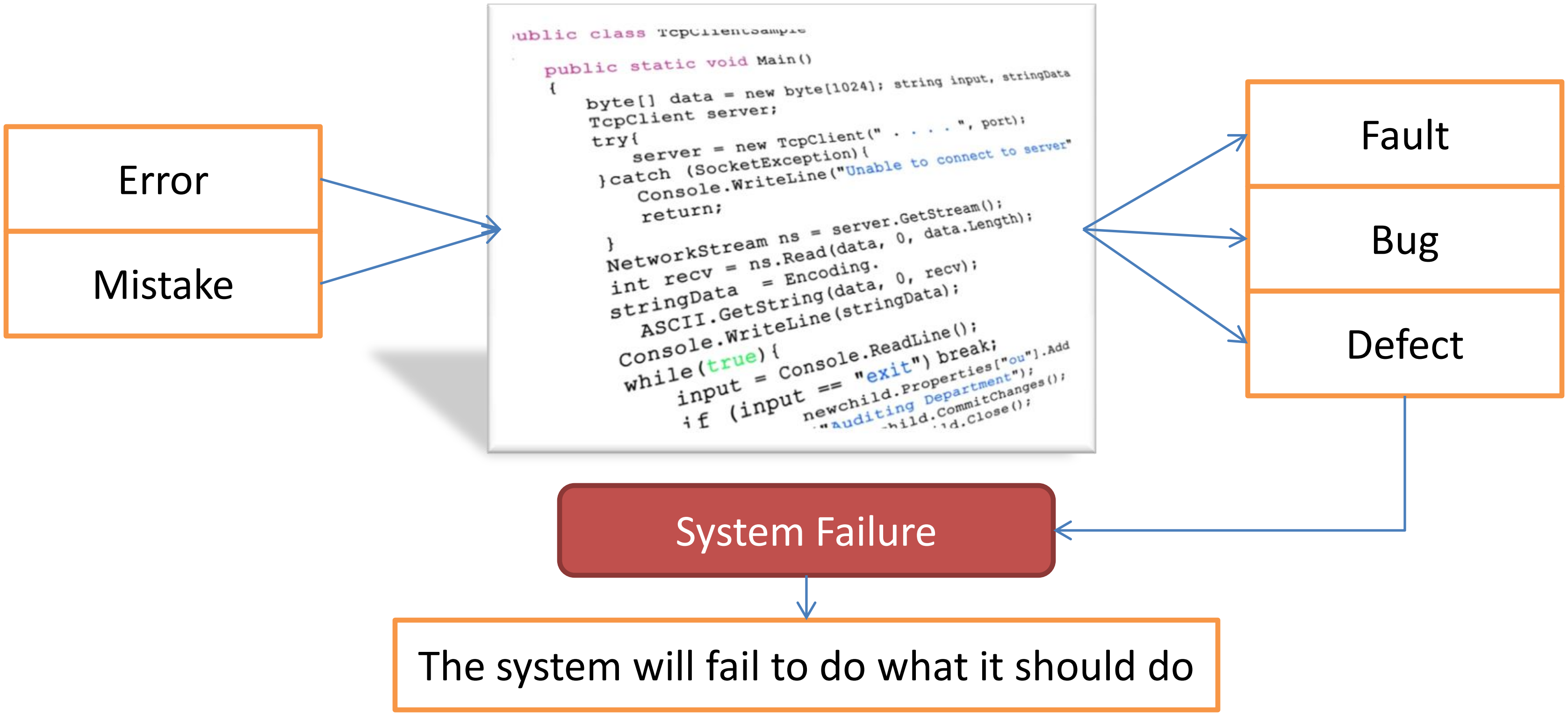
- User inconvenience
- Loss of money
- Loss of time
- Loss of business reputation
- Loss of life

Why Is Testing Necessary …

- We all make mistakes …
- Some mistakes are unimportant.
- Some of them are expensive or dangerous.

Failure Mechanism

Causes of Software Defects



Error

Mistake

```
ublic class TcpClientSample

  public static void Main()
  {
      byte[] data = new byte[1024]; string input, stringData
      TcpClient server;
      try{
          server = new TcpClient(" . . . . ", port);
      }catch (SocketException){
          Console.WriteLine("Unable to connect to server"
          return;
      }
      NetworkStream ns = server.GetStream();
      int recv = ns.Read(data, 0, data.Length);
      stringData  = Encoding.
          ASCII.GetString(data, 0, recv);
      Console.WriteLine(stringData);
      while(true){
          input = Console.ReadLine();
          if (input == "exit") break;
              newchild.Properties["ou"].Add
              "Auditing Department");
              hild.CommitChanges();
              ld.Close();
```

Fault

Bug

Defect

System Failure

The system will fail to do what it should do

Failure Causes

Errors & Faults Causes

Human Nature

Project Nature

Failure Causes

Failure Causes

- Errors in the requirement, specification, design, implementation
- Errors in use
- Environmental conditions
- Intentional damage

7 Testing Principles

7 Testing Principles

1. Testing Shows Presence of Defects

❑ Testing

- Shows defects.
- Can’t show there are no defects.
- Reduces the probability of undiscovered defects remaining in the SW.
- Not a proof of correctness.

7 Testing Principles

1. Testing Shows Presence of Defects

❑ Farming analogy

- If you see eaten leaves, you know you have bugs in your garden.
- If you had not seen the symptoms, could you be sure you had no bugs?
- Some bugs are easy to spot, others aren’t.

7 Testing Principles

2. Exhaustive Testing is Impossible

- Exhaustive testing == Testing every thing
- It takes impractical finite amount of time.

7 Testing Principles

2. Exhaustive Testing is Impossible

- Test time will be always limited.
- Use priorities to answer:
  - What to test first?
  - What to test most?
  - What not to test this time?

7 Testing Principles

3. Early Testing

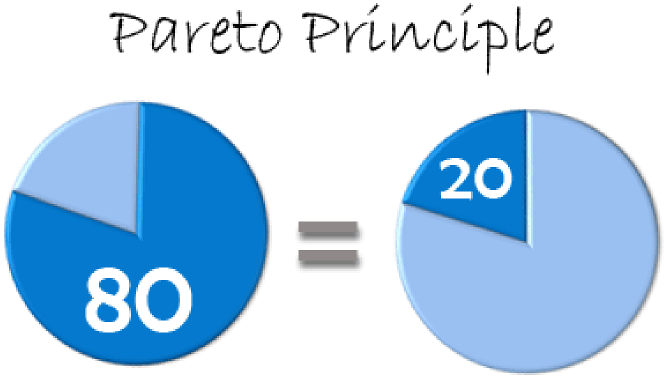
   - The cost of a bug tends to increase as the project continues.

   - Testing should start as early as possible (test design) and shall be focused on defined objectives (cheaper bug fixing, preventing bugs).

7 Testing Principles

4. Defect Clustering

- Bugs are not distributed evenly in the SW.
- Testing effort shall be focused proportionally to the expected and later discovered defect density of the modules.
- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most of operational failures.



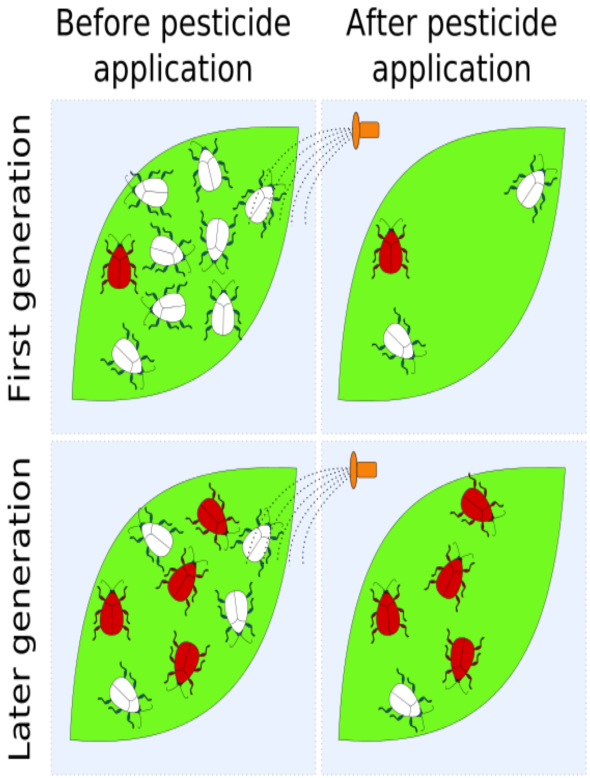
7 Testing Principles

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 defects.

- To overcome, the test cases need to be regularly reviewed and revised.

- New and different tests need to be written to exercise different parts of the software or system to potentially find more defects.



7 Testing Principles

6. Testing is Context Dependent

- Testing is done differently in different contexts.
  - Safety critical applications
  - Websites
- Different projects, organizations, and products have different testing needs.
- Best testing practices exist but you need to tailor them to your project.
- Failure to adapt the test team and its methods to these needs is a common result of dissolution of test teams.

7 Testing Principles

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.
- Finding and fixing many bugs does not guarantee user, customer, and/or stakeholder satisfaction.
- Many low-defect products have failed in the market place.
- Successful projects balance competing forces in terms of features, schedule, budget, and quality.

The End Of The 7 Testing Principles

Any questions

...?

Fundamental Test Process

Fundamental Test Process

- The Fundamental Test Process Consists Of:

1. Test planning and control
2. Test analysis and design
3. Test implementation and execution
4. Evaluating exit criteria and reporting
5. Test Closure activities

Fundamental Test Process

❖ Test Planning and Control

- Test planning is the activity of defining objectives of testing and the specification of test activities in order to meet the objectives and mission.

- Test control is the ongoing activity of comparing actual progress against the plan, and reporting the status , including the deviations from the plan.

Fundamental Test Process

❖ Test Analysis and Design

- Test analysis and design is the activity during which general testing objectives are transformed into tangible test conditions and test cases.

Fundamental Test Process

❖ Test Analysis and Design

- Reviewing the test basis (such as requirements, risk level, risk analysis, architecture, design, interface specifications)
- Evaluating testability of the test basis and test objects
- Identifying and prioritizing test conditions based on analysis of test items, the specification , behavior and structure of the software.
- Designing the test environment setup and identifying any required tools

Fundamental Test Process

❖ Test Implementation and Execution

- Developing, implementing, and prioritizing test cases & sequence
- Creating test data
- Preparing automation scripts
- Creating test suites
- Verifying test environment setup
- Updating traceability matrix
- Execution, Logging, Incidents reporting
- Re-testing and regression testing

Fundamental Test Process

❖ Evaluating Exit Criteria & Reporting

- Checking test logs against the exit criteria specified in test planning.
- Assessing if more test are needed or if the exit criteria specified should be changed.
- Writing a test summary report for stakeholders.

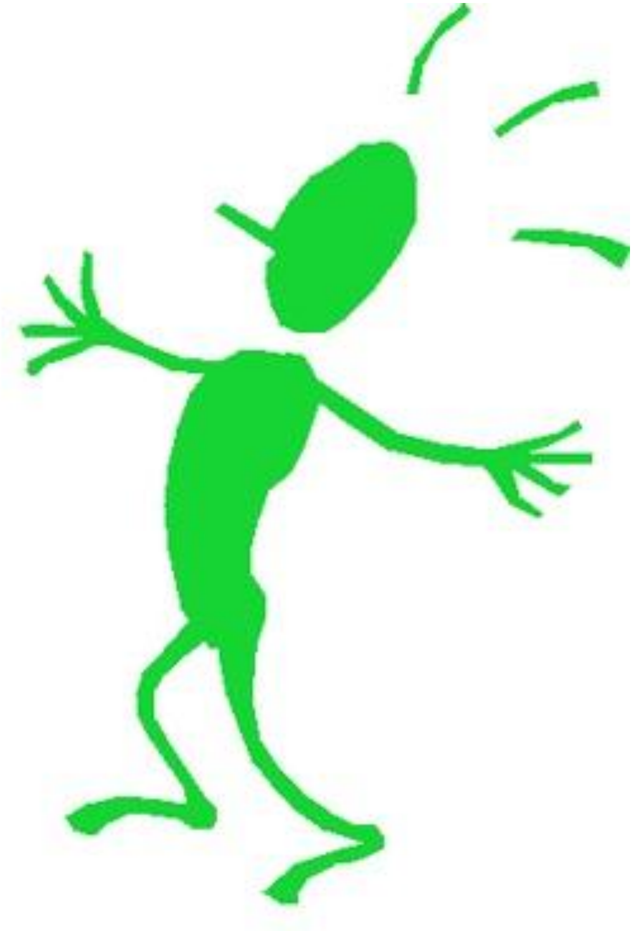
Fundamental Test Process

❖Test Closure Activities

- Checking deliverables
- Closing incident reports or raising any records for open incidents
- Documenting the acceptance of the system
- Archiving all test assets
- Handing all test assets to maintenance teams
- Analyzing lessons learned to improve test maturity

The End Of Fundamental Test Process

Any questions …?



IMT

SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/