

Testing Concepts

Lesson 1: Fundamentals of Testing



Lesson Objectives

To understand the following topics:

- Some Facts
- Introduction to Software Testing
- Software Testing - Definitions
- Need of Software Testing
- Error-Failure-Defect
- Causes of Software Defects
- Cost of Software Defects
- What does Software Testing reveal
- Importance of Software Testing
- Importance of Testing Early in SDLC Phases
- Testing and Quality
- Quality Perceptions
- Seven Testing Principles
- Economics of Testing
- How Testing is conducted?





Lesson Objectives

To understand the following topics:

- Software Testing – Then (Past)
- Software Testing – Now (Present)
- Scope of Software Testing
- Factors influencing the Scope of Testing
- Risk Based Testing
- Project Risks
- Product Risks
- Need of Independent Testing
- Activities in Fundamental Test Process
- Attributes of a good Tester
- Psychology of Testing
- Code of Ethics for Tester
- FS SBU: Focus on Testing
- Testing Roles in iTEAMS
- Limitations of Software Testing



Some Facts !!



Ariane 5 explosion - data conversion of a 64-bit no to 16-bit

Patriot missile - rounding error....Kills 28, injures 100

Mars Climate Orbiter lost - (Mixture of pounds and kilograms, 1999)

F16 autopilot - flipped plane upside down whenever it crossed the equator

Microsoft's anti-Unix site crashes News: vnunet.com

Yahoo glitch strikes again
03/20/2002

Some Facts !!



Excel gives $77.1 \times 850 = 100000$ instead of 65535

Y2K problem in Payroll systems designed in 1974

Disney's Lion King - 'Simba'



Introduction to Software Testing

Software Testing is the process of executing a program with the intent of finding errors as early as possible in SDLC

It is a process used to help identify the correctness, completeness and quality of a developed computer software

Software Testing helps in Verifying and Validating if the Software is working as it is intended to be working

Testing is a process that helps in finding out how well the product works :

- Aimed at finding defects
- Aimed at demonstrating lack of quality
- Aimed at demonstrating the gap between specifications and actual product
- Aimed at building faith in the end product that gives advice on quality and risk



Software Testing - Definitions

The process of executing a program (or part of a program) with the intention of finding errors - G. J. Myers

Software testing is the process of testing the functionality and correctness of software by running it

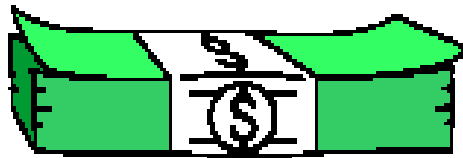
Testing is the process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements - IEEE 83a

The process of analyzing a system to detect the difference between existing and required conditions and to evaluate the feature of the system - IEEE/ANSI, 1983 [Std 829-1983]



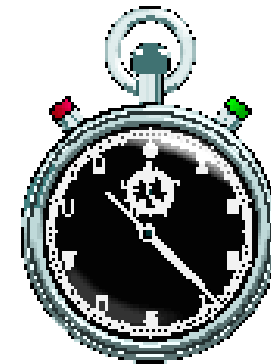
Need of Software Testing

To find greatest possible number of errors with manageable amount of efforts applied over a realistic time span with a finite number of test cases



Cost effective

Time
limited



Because software is likely to have faults

Because failures can be very expensive

To contribute to the delivery of higher quality software product

To detect the faults in the Software before User finds it

➤ **To know the reliability of the software**

Undetected errors are costly to detect at a later stage

To satisfy users and to lower the maintenance cost



Error-Failure-Defect

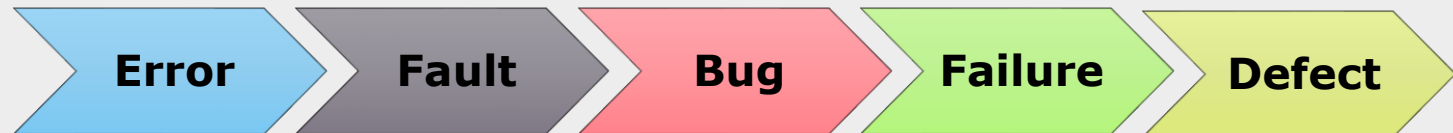
Error(Mistake): A human action that produces an incorrect result

Fault: A stage caused by an error which leads to unintended functionality of the program

Bug: It is an evidence of the fault. It causes the program to perform in unintended manner. It is found before application goes into beta version

Failure: Inability of the system to perform functionality according to its requirement

Defect: It is a mismatch of the actual and expected result identified while testing the software in the beta version





Causes of Software Defects

Software is written by human beings

- Who know something, but not everything
- Who have skills, but aren't perfect
- Who do make mistakes (errors)

Under increasing pressure to deliver to strict deadlines

- No time to check but assumptions may be wrong
- Systems may be incomplete

Environmental conditions

- Radiations, Magnetism, Electronic fields and pollution can cause faults in firmware or influence execution of software by changing hardware conditions

Minimal or no proper documentation of Business Requirements

Insufficient time window for development

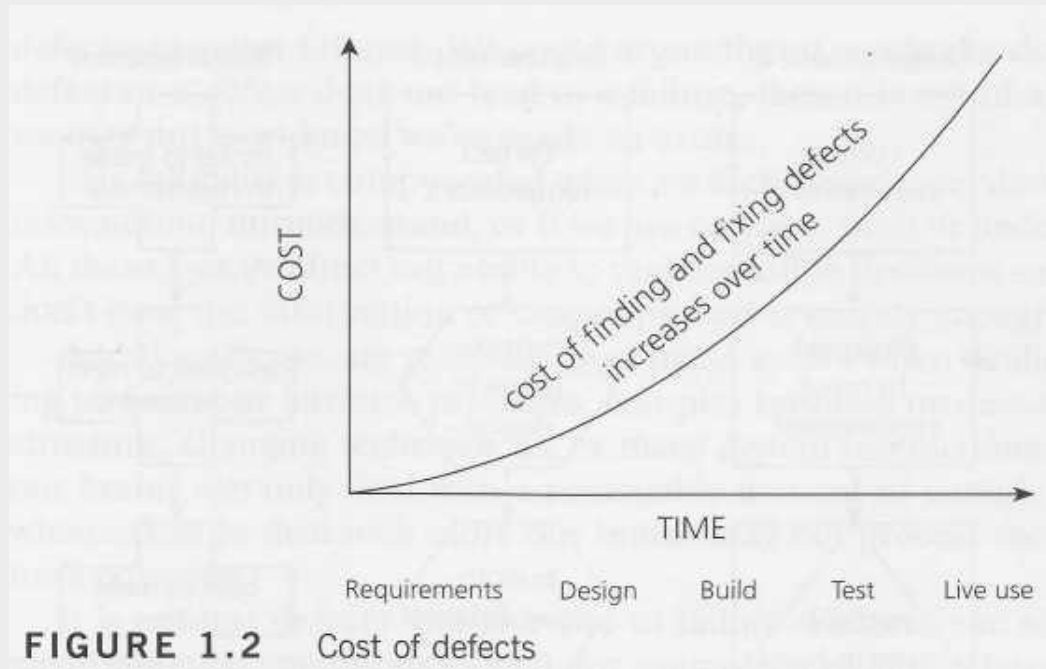
Lack of domain knowledge

Programming Language constraints

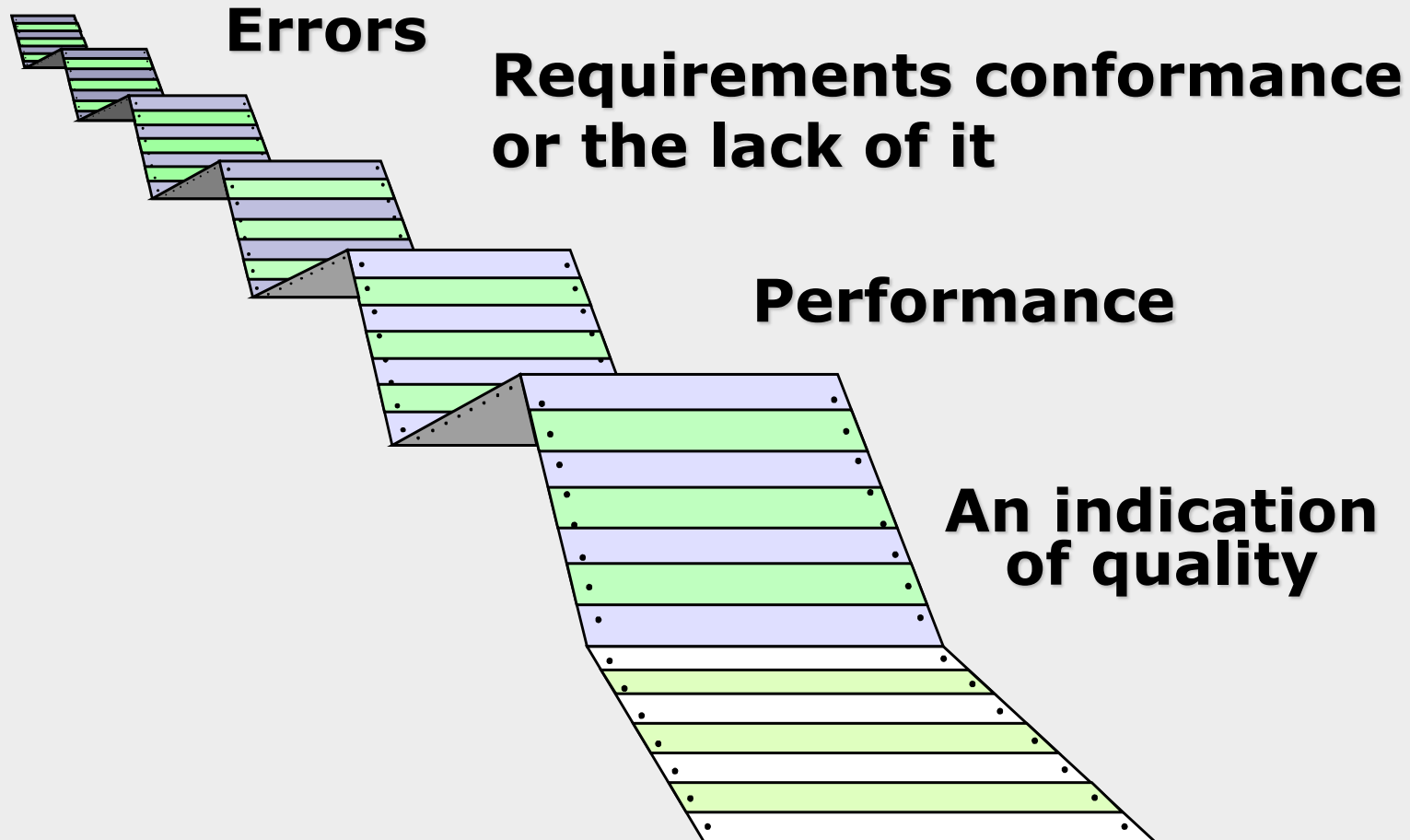
Cost of Software Defects



It is Easy to find and fix defect in early stages rather than in the later phases of software.



What does Software Testing reveal





Importance of Software Testing

Ensures that the product is usable

Ensures that Customer's Objectives are met

Early detection of errors to prevents breakdown at a later stage

Ensures that the software is reliable

Builds Confidence in software

Increases Customer Satisfaction

Ensures effective execution in the given environment

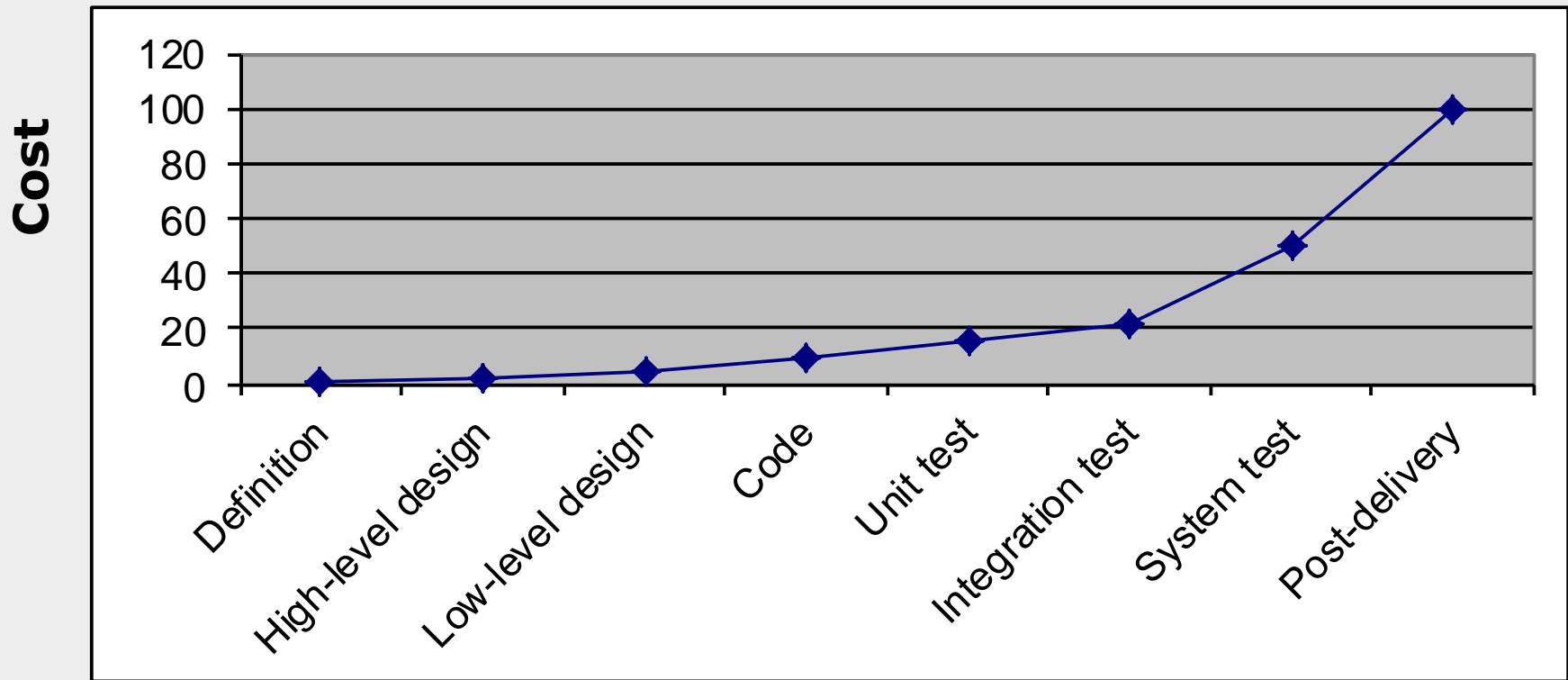
Reduces overall cost of software

Reduces time for going live (production)

Importance of Testing Early in SDLC Phases



Error removal cost over SDLC





Importance of Testing Early in SDLC Phases

Prevents future Problems thus lowering the cost

Testing will not be a bottleneck anymore

Testers become more familiar with the software, as they are more involved with the evolution of the product.

Reduces the chances of failure

The test environment can be prepared in advance

The risk of having a short time for testing is greatly reduced

Maintains “quality culture” in the organization



Testing and Quality

Testing helps to measure the Quality of software in terms of

- Number of defects found
- important information regarding the Non functional attributes like Reliability, Security, Performance, etc.

Quality Definition

- Conformance to explicitly stated functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected from all professionally developed software.



Quality Perceptions

Engineer may judge :

- User satisfaction
- Portability
- Maintainability
- Robustness & Efficiency

Customer may judge :

- Cost

User may judge :

- Reliability
- usability



Seven Testing Principles

Principle 1 - Testing shows presence of defects but cannot prove that there are no defects

Principle 2 - Exhaustive testing is impossible

Principle 3 - Early testing

Principle 4 - Defect Clustering

Principle 5 - Pesticide Paradox

Principle 6 - Testing is context dependent

Principle 7 - Absence of Errors fallacy



Seven Testing Principles : cont

Principle 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.

Principle 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.

Principle 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.

Principle 4: Defect clustering

A small number of modules contain most of the defects discovered during pre-release testing or show the most operational failures.



Principle 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.

Principle 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.

Principle 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.



Economics of Testing

Economics of Testing

- It is both the driving force and the limiting factor

Driving - Earlier the errors are discovered and removed in the lifecycle, lowers the cost of their removal.

Limiting - Testing must end when the economic returns cease to make it worth while i.e. the costs of testing process significantly outweigh the returns



How Testing is conducted?

- By examining the users' requirements
- By reviewing the artifacts like design documents
- By examining the design objectives
- By examining the functionality
- By examining the internal structures and design
- By executing code



Software Testing – Then (Past)

Testing led to blame each other and giving faulty justifications and excuses which would arise quarrels among the team members

Not much re-work was welcomed

Developers misunderstood that increased testing would increase the project cost unnecessarily

It was understood that software has quality if it is just easy to maintain, reusable and flexible.

Testing was conducted only at project execution phase

Ad-hoc & need driven

Totally manual

Testing jobs perceived as low scale as compared to other discipline



Software Testing – Now (Present)

Organized body of knowledge

- Independent Testing teams
- Testing Models
- Testing knowledge Groups

A specialized engineering discipline in great demand

Defects are highlighted and brought to surface so as to conduct corrective measures and achieve user satisfaction

Testing lead to cooperative solutions so as to prevent the failures

Documentation is considered as essential to note down the lessons learnt so that mistakes are not repeated

Developers know that testing increases the business profit

Describes software quality in terms of integrity, reliability, usability and accuracy

Testing is conducted as the project initiates



Scope of Software Testing

Bad news : You can't test everything

Good news : There is such a thing as "good enough"

Bad news : Good enough may cost too much

What do we do : Increase focus via systematic process of elimination

What you might test?

- Those areas which are within the scope of your project

What you should test?

- The critical system functionality which effects the customers & users experience of quality

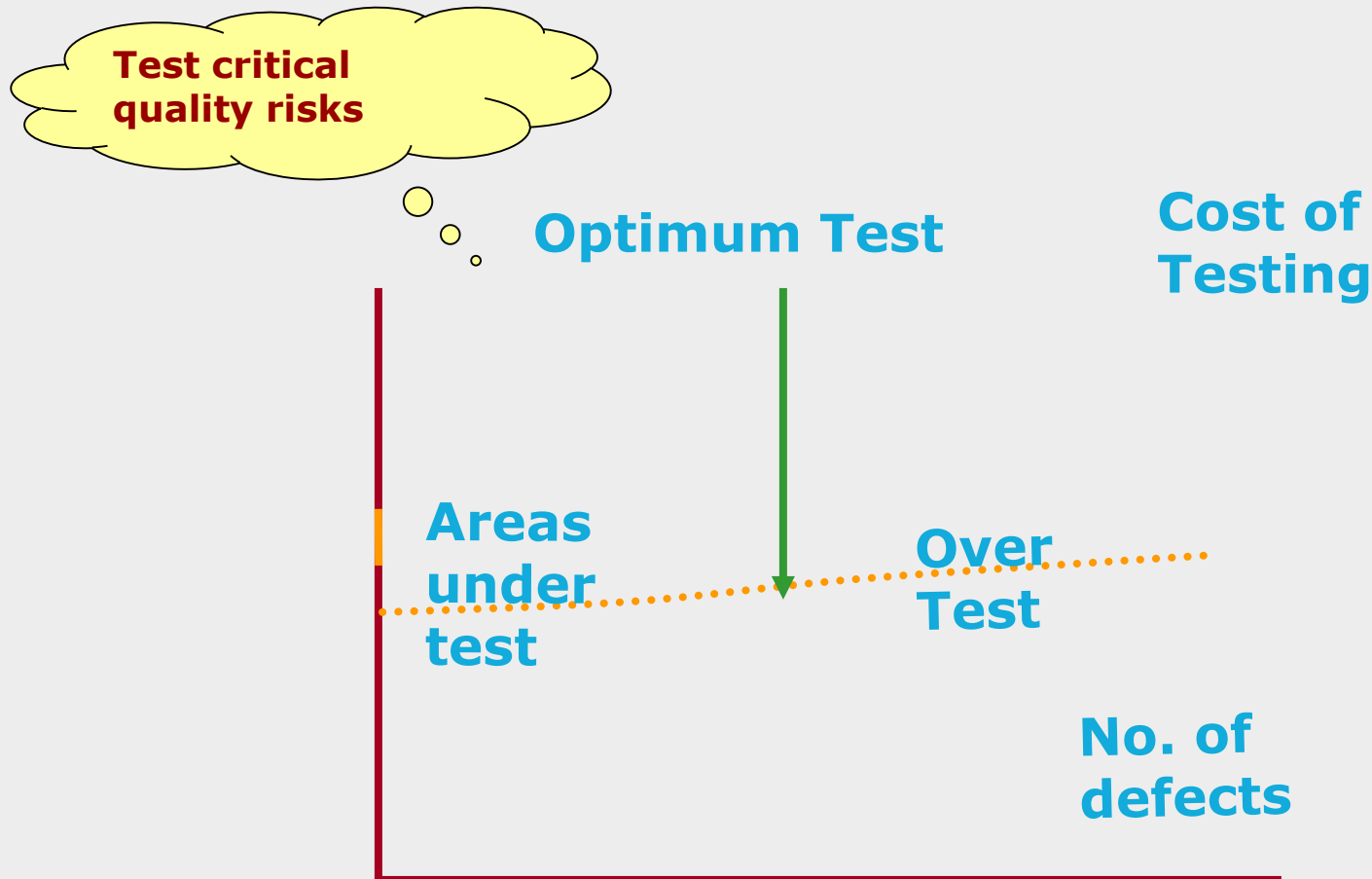
What you can test?

- Estimate time & resource for risk driven effort

Scope of Software Testing



“Understanding risk is the key to Optimum testing”





Factors influencing the Scope of Testing

Contractual requirements

Legal requirements

- Privacy related laws
- Non-disclosure of identity

Industry-specific requirements

- Aircraft safety equipment

Scope of Testing is about identifying the correct test cases for automation

The steps involved are:

- Identify various factors that form the basis of identifying the candidate test cases
- Apply 'Divide & rule' strategy : Break the application into smaller modules
- Analyze each module to identify the candidate test cases
- Calculate ROI

Factors influencing the scope of testing :

- In small projects
 - Test case writing
 - Test case execution
 - Regression testing
- In Large projects
 - Setting up test bed
 - Generating test data, test scripts, etc.



Risk Based Testing

Risk:

- A factor that could result in negative consequences; usually expressed as impact and like hood
- Risks are used to decide where to start testing and where to test more.
- Risked Based testing:
- Testing oriented towards exploring and providing information about product risks

Risk based Testing is used to reduce risk of adverse effect occurring or to reduce the impact of adverse effect

- It draws on the collective knowledge and insight of the project stakeholders to determine the risk and the level of testing required to address those risks.



Project Risks

Project Risk

- A risk related to management and control of the (test) project is called as Project Risk

Project risk are:

Organizational factor

- Skill, training and staff shortage
- Personnel issues
- Political issues
- Improper attitude toward or expectation of testing

Technical issues

- Problem in defining right requirements and quality of the design code, test data and test
- The extent to which requirement cannot be met given existing constraints
- Test environment not ready on time or late data conversion, migration planning and development and testing data conversion/migration tools

Supplier issues

- Failure of a third party
- Contractual issues



Product Risks

Product Risk: it is directly related to the test object

Risks related to quality of a product:

- Failure-prone software delivered
- The Potential that the software/hardware could cause harm to an individual or company
- Poor software characteristics
- Poor data integrity and quality
- Software that does not perform its intended functions



Need of Independent Testing

- Unbiased testing is necessary to objectively evaluate quality of a software
- Developer carrying out testing would not like to expose defects
- Assumptions made are carried into testing
- People see what they want to see.
- More effective in terms of Quality & Cost
- It is conducted by an independent test team other than developer to avoid author bias and is more effective in finding defects and failures
- The tester sees each defect in a neutral perspective
- The tester is totally unbiased
- The tester sees what has been built rather than what the developer thought
- The tester makes no assumptions regarding quality



Activities in Fundamental Test Process

Test planning and control

- It defines the objectives and specification of test activities
- Test control is the on going activity of comparing actual progress against the plan

Test analysis and design

- Testing objectives are transformed into tangible test conditions and test cases
- Reviewing the test basis
- Evaluating testability of the test basis and test objects
- Identifying, Designing and Prioritizing test conditions based on analysis of test item, the specification, behaviour and structure of the software
- Designing and Prioritizing high level test cases
- Identifying necessary test data
- Designing test environment setup and identifying required infrastructure and tools
- Creating bi-directional traceability between test basis and test cases



Activities in Fundamental Test Process

Test implementation and execution

- Test procedures (scripts) are specified by combining test cases in a particular order and the environment is set up and tests are run
- Finalizing, implementing and prioritizing test cases
- Developing and prioritizing test procedures, creating test data and writing automated test scripts
- Creating test suites from test procedures for efficient test execution
- Verification if Test environment is setup correctly
- Verifying and updating bi-directional traceability between test basis and test cases
- Executing test procedure using tool or manually according to the planned sequence
- Logging the outcome of the test execution and recording the identities and version of the software under test tools and test ware
- Comparing actual result with expected result
- Reporting discrepancies and analysing their root cause
- Repeating test activities as a result of action taken for each discrepancy



Activities in Fundamental Test Process

Evaluating exit criteria and reporting

- Test execution is assessed against the defined objectives
- Test logs are checked against the exit criteria specified in test planning
- Assessment is done if more tests are needed
- Test summary report is written

Test closure activities

- Data from completed test activity is collected to consolidate experience, facts and numbers
- This is done at milestones



Attributes of a good Tester

- A good test engineer has a 'test to break' attitude
- Need to take a different view, a different mindset (“What if it isn’t?”, “What could go wrong?”)
- An ability to understand the point of view of the customer
- A passion for quality and attention to detail
- Notice little things that others miss/ignore (See symptom not bug)
- Ability to communicate fault information to both technical (developers) and non-technical (managers and customers)
- Tact and diplomacy for maintaining a cooperative relationship with developers
- Work under worst time pressure (at the end)
- “Patience”



Psychology of Testing

- Test Engineers pursue defects not people
- Don't assume that no error(s) will be found
- Test for Valid and Expected as well as Invalid and Unexpected
- The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section
- Testing is extremely creative and intellectually challenging

Code of Ethics for Tester



Involvement in software testing enables individuals to learn confidential and privileged information. A code of ethics is therefore necessary, among other reasons to ensure that the information is not put to inappropriate use.

PUBLIC - Tester shall act consistently with public interest

CLIENT AND EMPLOYER - Tester shall act in best interests of their client and employer

PRODUCT - Tester shall ensure that the deliverables they provide meet highest professional standards possible

JUDGMENT - Tester shall maintain integrity and independence in their professional judgment

MANAGEMENT - Tester managers and leaders shall subscribe to and promote an ethical approach to manage software testing

PROFESSION - Tester shall advance the integrity and reputation of the profession consistent with the public interest

COLLEAGUES - Tester shall be fair to and supportive of their colleagues, and promote cooperation with software developers

SELF - Tester shall participate in lifelong learning and shall promote an ethical approach to the practice of the profession



FS SBU: Focus on Testing

Fastest growing service line is FS SBU (Financial Services Strategic Business Unit)

Team of 1600+ test engineers

Was CMM level 5 assessed within a year of being established

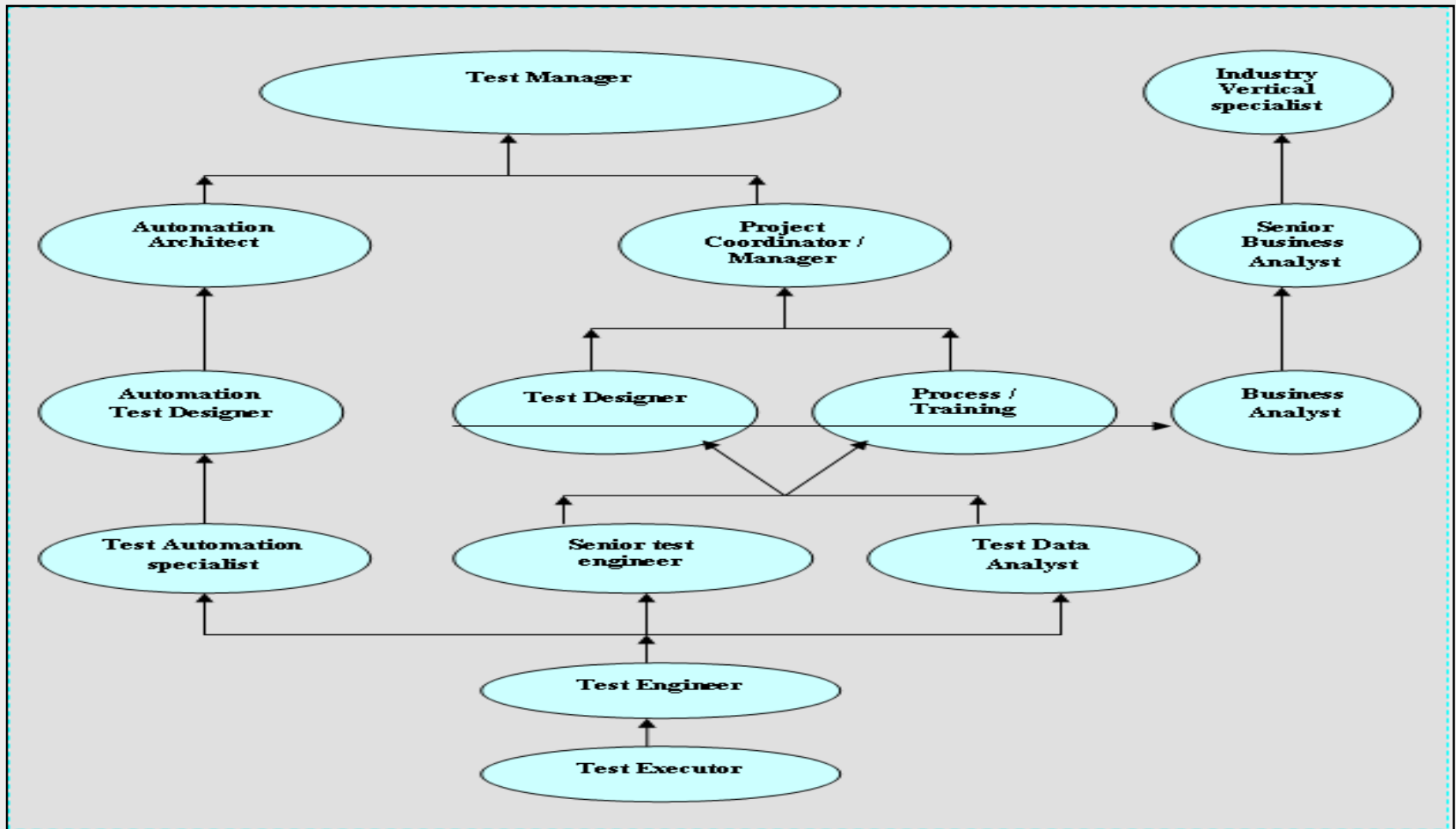
Presence in most of the existing FS SBU accounts

High Focus on competency building

Do almost all types of testing including Performance monitoring and Public Website testing

Offering Security Testing

Testing Roles in iTEAMS





Limitations of Software Testing

Even if we could generate the input, run the tests, and evaluate the output, we would not detect all faults

Correctness is not checked

- The programmer may have misinterpreted the specs, the specs may have misinterpreted the requirements

There is no way to find missing paths due to coding errors

Summary



In this lesson, you have learnt:

- Testing is an extremely creative & intellectually challenging task
- No software exists without bug
- Testing is conducted with the help of users requirements, design documents, functionality, internal structures & design, by executing code
- Scope of testing
- The cost of not testing is potentially much higher
- Testing is in a way a destructive process
- A successful test case is one that brings out an error in program
- Various principles of testing





Review Question

Question 1: What is visible to end-users is a deviation from the specific or expected behavior is called as

- Defect
- Bug
- Failure
- fault

Question 2: _____ is a planned sequence of actions

Question 3: Pick the best definition of Quality :

- Quality is job done
- Zero defects
- Conformance to requirements
- Work as designed

Question 4: One cannot test a program completely to guarantee that it is error free

- Option: True / False

Question 5: One can find missing paths due to coding errors

- Option: True / False





Review Question: Match the Following

1. Economics of limiting
2. Testing
3. A good test case
4. Use every possible input condition as a test case

A. Driving
B. Exhaustive testing
C. Limiting
D. Test cycle
E. Comparing outputs with specified or intended
F. Maximize bug count

