COURSE NAME

SOFTWARE QUALITY AND TESTING

CSC 4133

(UNDERGRADUATE)

CHAPTER 1

BASICS OF SOFTWARE QUALITY ASSURANCE

WHAT IS A SOFTWARE?

- ❑ Software is:
  - More than just a computer program!
  - Composed of computer programs, procedures, and possibly associated documentation and data related to the operation of a computer system.

- ❑ Two major types of Software:
  - Generic (Buy) – Stand alone, Sold on open market (OTH)
  - Customized (Build) – For specific customer or business
  - BUY & BUILD – Most software that you buy for personal use will likely be 'commercial off-the-shelf' (COTS), means the same application will be run by thousands of other users. It may not work exactly how you'd like it to or do everything you'd want it to whereas a customized (bespoke) software solution is built exactly to the business' specific requirements.

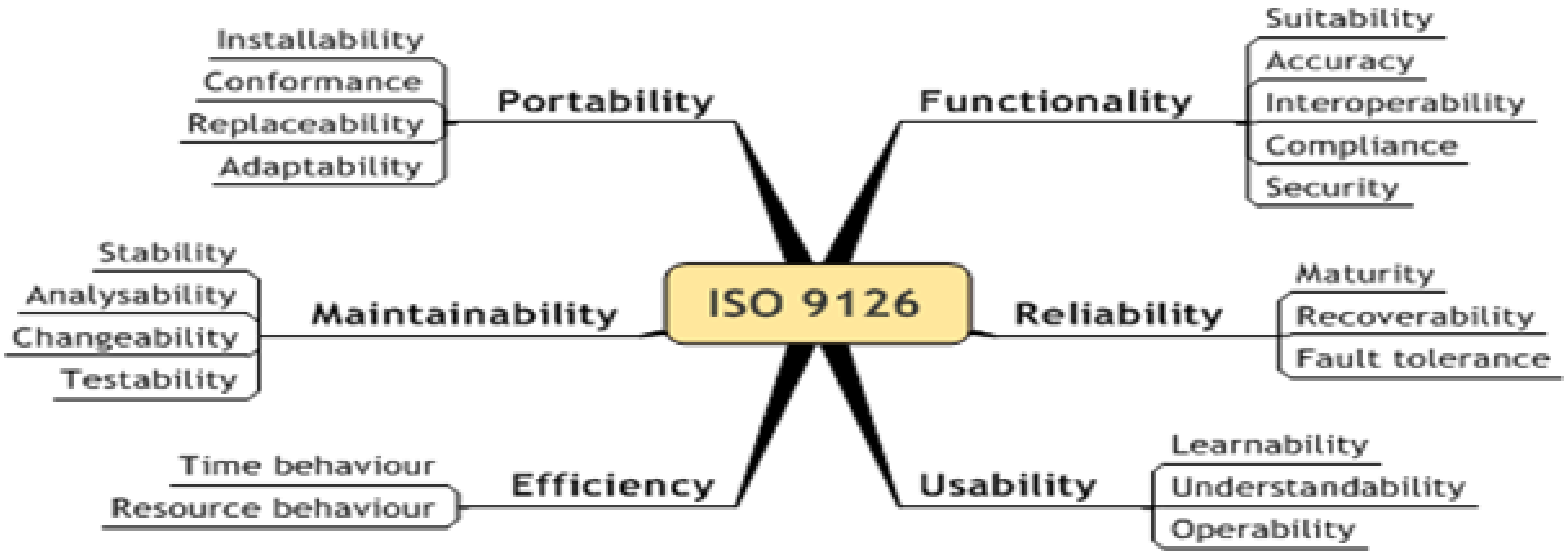
WHAT IS SOFTWARE QUALITY?

- ❑ Software Quality (as per ISO/ IEC 9126):

  The totality of functionality and features of a software product that contributes to its ability to satisfy stated or implied needs. It can be customized for organizations.

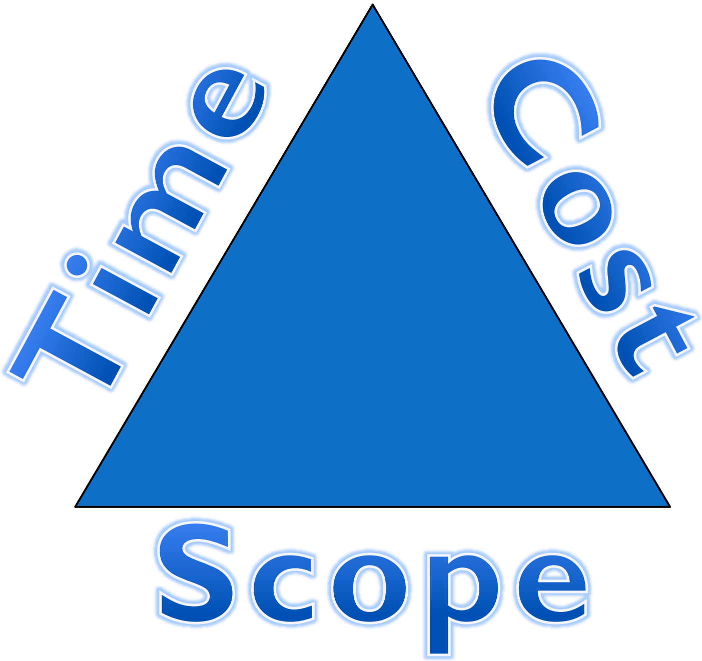
- ❑ Software Quality (as IEEE Std 610):

  The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations.

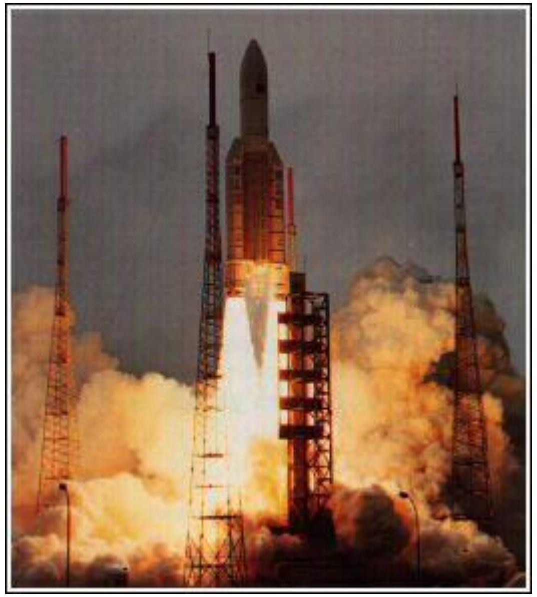


CHALLENGES IN SOFTWARE PROJECTS

- ❑ The primary goal of project management activity is to deliver the product on time, on budget and expected quality
- ❑ What’s the main challenges of software development now-a-days?
  - Time: late, difficult to deliver on time
  - Cost: high cost, over budget
  - Scope: low quality with faults



EXAMPLE OF SOFTWARE DEFECTS



❑ Flight Ariane 5

- Most Expensive Computer Bug in History
- On June 4, 1996, the rocket Ariane 5 tore itself apart 37 seconds after launch because of a malfunction in the control software making the fault most expensive computer bug in history.

❑ Lethal X-Rays :Therac-25 system

Therac-25 was a radiation therapy machine produced by Atomic Energy of Canada Ltd (AECL) in 1986. But initially lot of people died because of massive overdose of radiation. And this is happened because of a software bug.

SOFTWARE TESTING

- Software Testing is the process of executing a system or component under specified conditions with the intent of finding defects/bugs and to verify that it satisfies specified requirements.

- Goals of Software Testing
  - Main goal is to detect issues

    - Requirement conformance

    - Errors in system operation

    - System performance (TCAS in Autopilot system)

  - Have different levels of testing operation

    - Unit testing in developing each module

    - Integration testing to combine different module and check incompatibility

    - System testing is done at system combination of software and hardware

ROLE OF TESTING

Software quality assessment divide into two categories:

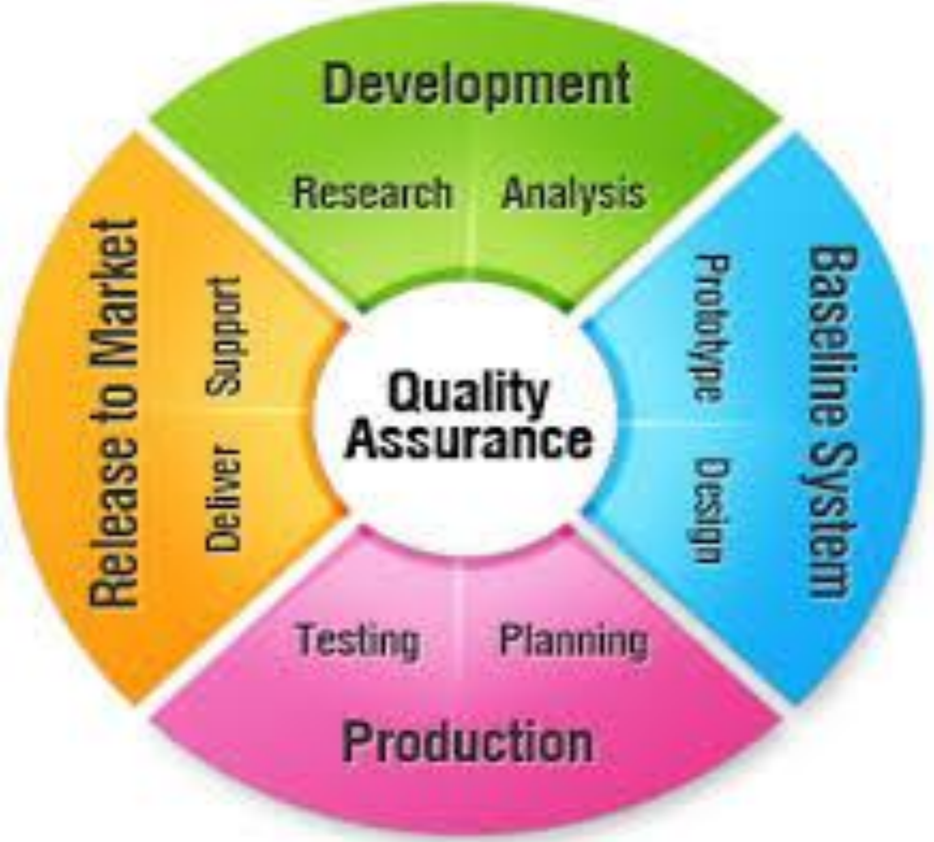
- Static analysis (manual testing)
  - It examines the code/document and reasons over all behaviours that might arise during run time
  - Examples: code review, inspection, and algorithm analysis
  - Limitation: takes longer time in review process
- Dynamic analysis (automated testing)
  - Actual program execution to expose possible program failure
  - One observe some representative program behaviour, and reach conclusion about the quality of the system
  - Limitation: It can not think out of the box (supervised learning) finds only syntax error
- Static and Dynamic Analysis are complementary in nature. Focus is to combine the strengths of both approaches

Static Testing

Dynamic Testing

SOFTWARE QUALITY ASSURANCE (SQA)

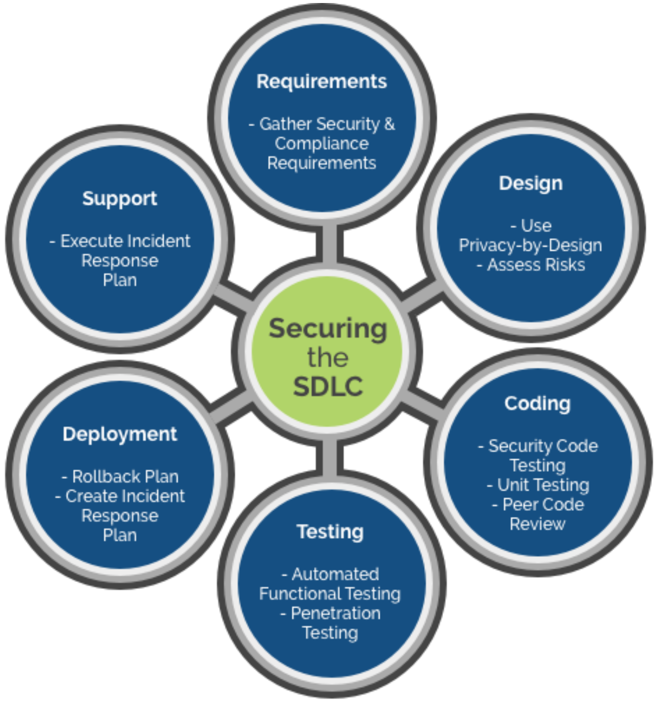
- Develop an effective plan for software development
- The function of software quality that assures that the standards, processes and procedures are appropriate for the project and are correctly implemented.
- Defined as a planned and systematic approach to the evaluation of the quality of and adherence to software product standards, processes and procedures.
- A set of activities designed to ensure that the development and/or maintenance process is adequate to ensure a system will meet its objectives.



- An umbrella activity that is applied throughout the software process.
- Consists of a means of monitoring the software engineering processes and methods used to ensure quality. An effective approach to produce high quality software.

SOFTWARE QUALITY CONTROL (SQC)

- Execute the software development plan effectively
- The function of software quality that checks that the project follows its standards processes, and procedures, and that the project produces the required internal and external (deliverable) products.
- A set of activities designed to evaluate a developed work product. It includes the following activities:
  - Requirement Review (requirements document)
  - Design Review (system design)
  - Code Review (system module)
  - Deployment Plan Review (agile delivery process)
  - Test Plan Review
  - Test Cases Review



Securing the SDLC

Requirements
- Gather Security & Compliance Requirements

Design
- Use Privacy-by-Design
- Assess Risks

Coding
- Security Code Testing
- Unit Testing
- Peer Code Review

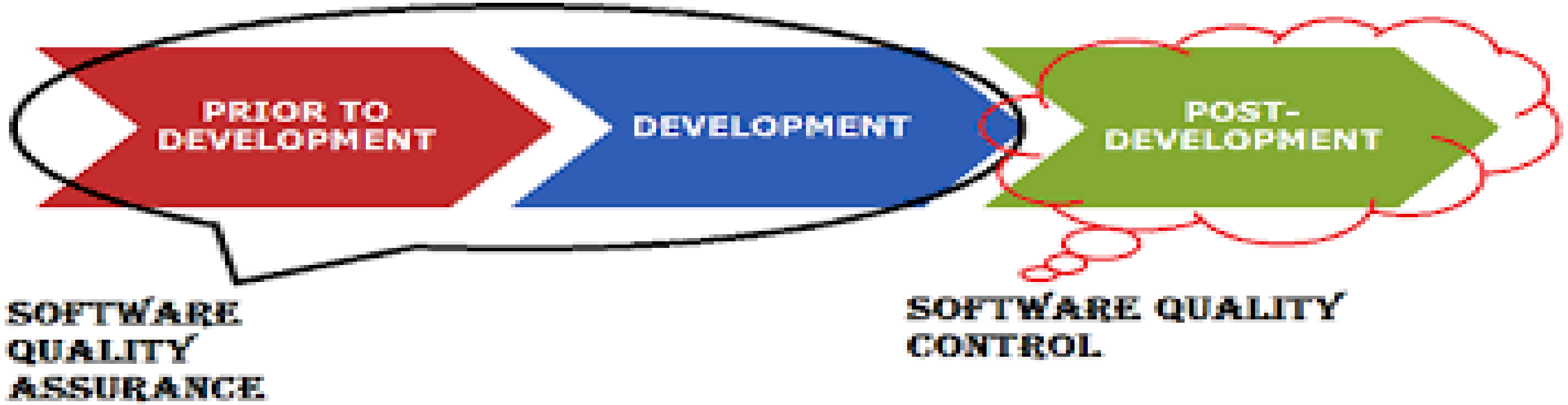
Testing
- Automated Functional Testing
- Penetration Testing

Deployment
- Rollback Plan
- Create Incident Response Plan

Support
- Execute Incident Response Plan

DIFFERENCE BETWEEN SQA & SQC

SQA	SQC
A process which deliberate on providing assurance that quality request will be achieved	A process which deliberates on fulfilling the quality request
Aim is to prevent the defect	Aim is to identify and resolve the defects
Is the technique of managing the quality	Is the method to verify the quality
Does not involve executing the program	Always involves executing the program
All team members are responsible	Only testing team is responsible



DIFFERENCE BETWEEN SQA & SQC

SQA	SQC
Verification - Makes sure you are doing the things right (is process oriented)	Validation - Makes sure the results of what you've done are what you expected (product oriented)
Planning for doing a process	Action for executing the planned process
Defines standards and methodologies to followed in order to meet the customer requirements	Ensures that the defined standards are followed while working on the product
The process to create the deliverables	The process to verify that deliverables
Responsible for full software development life cycle (SDLC)	Responsible for software testing life cycle
Detects weakness	Detects defects
Is failure prevention system	Is failure detection system

VALIDATION AND VERIFICATION

Validation

- Software systems must do what they are supposed to do; they must do the right things
- Validation is a process that ensures the software product meets the customer requirements.
- Building the correct product

Verification

- Software systems must perform the specific tasks correctly; they must do the things right
- Verification is a process that ensures the software product works properly.
- Building the product correctly

DIFFICULTIES ON QUALITY EXPECTATIONS

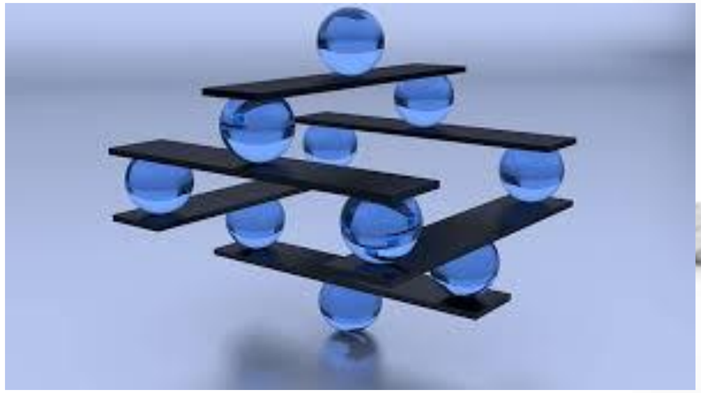
- ❑ Difficulties in achieving good quality:
  - Size (LOC in the function code)
    Manual testing would be difficult to find logical errors in bulky source code
  - Invariant Complexity in the Product and Project
    Complexity in S/W is more because it deals with logic where any error in logic may cause significant impact such as autopilot software (e.g. set landing mode)
  - Environmental stress/constraints
    Test the system with it maximum load (e.g. result publish website)
  - Flexibility/adaptability expected
    Frequent change because of flexibility in S/W (e.g. regression testing after each change)
  - Cost and market conditions
    Testing cost with automated tools and license

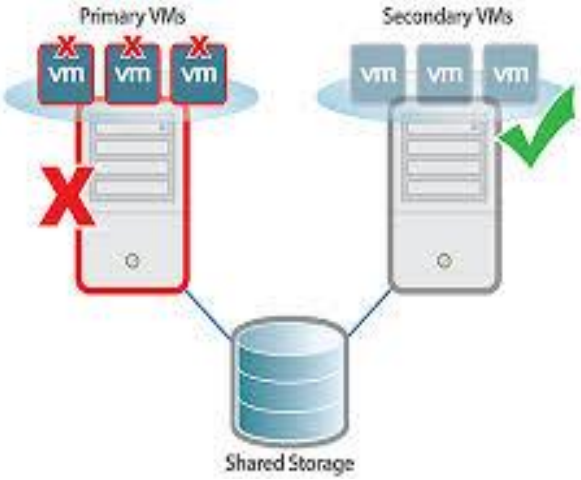
SOFTWARE QUALITY ENGINEERING (SQE)

- ❑ To ensure software quality through the related validation and verification activities.
- ❑ To be carried out by the people and organizations responsible for developing and supporting these software systems in an overall quality engineering process:
  - Planning: Quality planning
  - Execution: Execution of selected QA plan or software validation & verification activities
  - Decision: Decide pass/fail criteria of a test case. Measurement and Analysis to provide convincing evidence to demonstrate software quality to all parties involved
- ❑ Customers and users need to have the assurance that their quality expectations are satisfied by the delivered software systems.

SQE ACTIVITIES

- ❑ Testing: remove defect & ensure quality
- ❑ Other QA alternatives to testing
  - Defect Prevention
    (plan for avoid any defect)
  - Formal Verification
    (e.g. Inspection/Review/Walkthrough)
  - Fault Tolerance
    (e.g. data backup, autosave document, system reliability upto 5 failure)





ERROR, FAULT, FAILURE, AND DEFECT

Error

- A human action that produces an incorrect result
- Missing or incorrect human action resulting in certain fault(s) being injected into a software

Fault

- An incorrect step, process, or data definition in a computer program
- An underlying condition within a software that causes certain failure(s) to occur

Failure

- The inability of a system or component to perform its required functions within specified performance requirements
- A behavioural deviation from the user requirement or the product specification

Defect

- Failures, faults, and errors are collectively referred to as defects in literature.

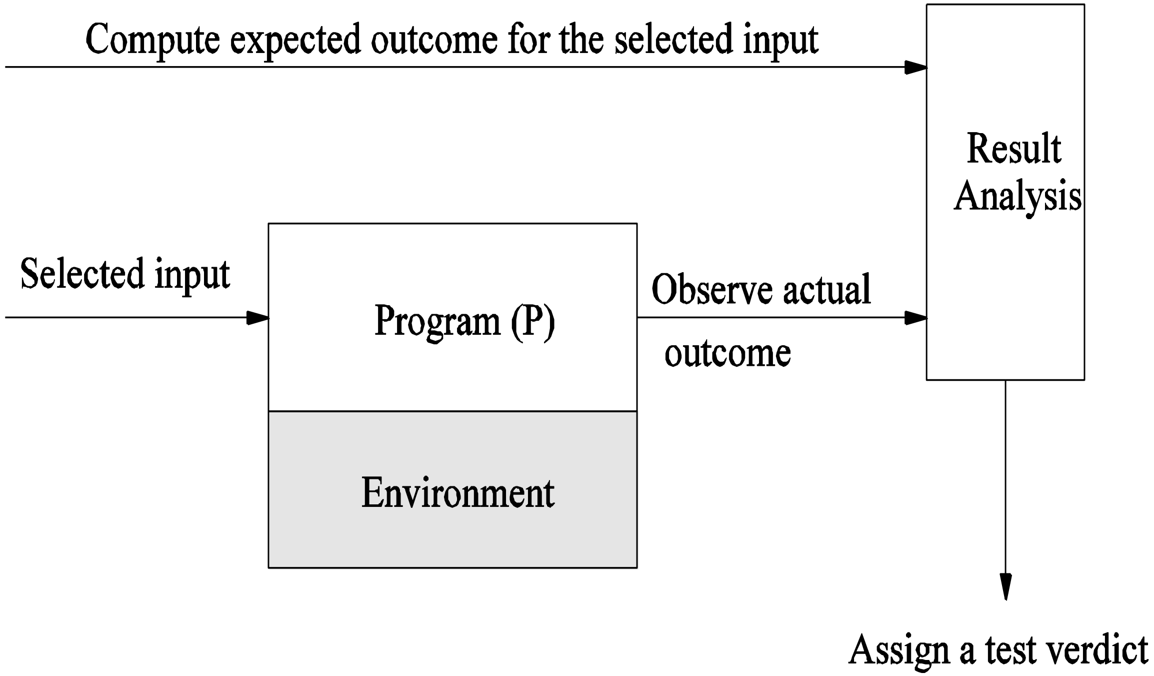
Note: Software problems or defects, are also commonly referred to as “bugs”

COMPLETE TESTING

- The objectives of Testing is to Detect bugs
  - Reduce the risk of defects
  - Reduce the cost of testing

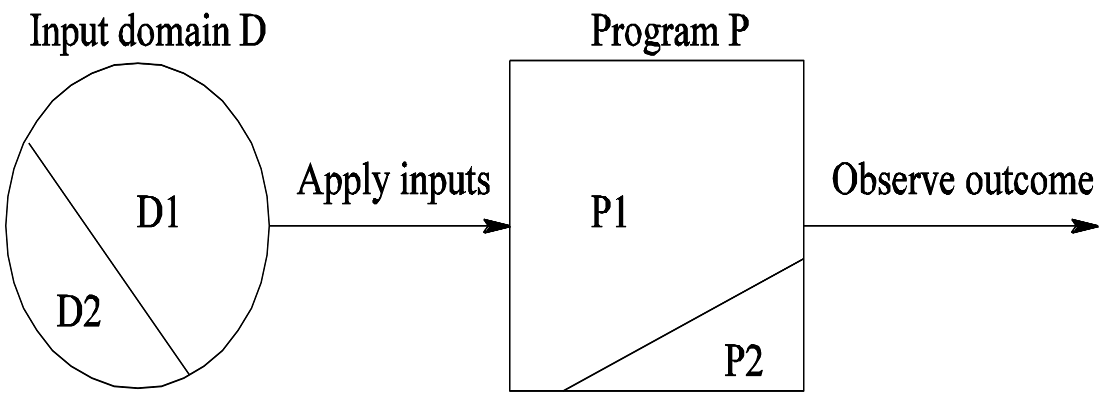
- Complete/Exhaustive testing
  - There are no undisclosed faults at the end of test phase
- Complete testing is nearly impossible for most of the systems, Because..
  - The domain of possible inputs of a program is too large (valid inputs and invalid inputs)
  - The design issues may be too complex to completely test. And testing effort/budget may very high for testing the system completely
  - It may not be possible to create all possible execution environments of the system (create all possible simulation of an auto pilot testing)

TESTING ACTIVITIES



- Identify the objective to be tested (prioritized)
- Select inputs from input domain
- Compute the expected outcome
- Set up the execution environment of the program
- Execute the program
- Analyse the test results with expected outcome

- A subset of the input domain exercising a subset of the program behaviour
- Divide the input domain D into D1 and D2
- Select a subset D1 of D to test program P
- It is possible that D1 exercise only a part P1 of P



REFERENCES

- Software Testing And Quality Assurance – Theory and Practice - Kshirasagar Naik & Priyadarshi Tripathy
- Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement - Jeff Tian