GENIUS | Page No.
Date / /

Unit - IV (Testing Strategies and Quality Management)

(#) Testing Strategies for Software

Software testing is the process of evaluating a software application to identify if it meets specified requirements and to identify and defects.

- Software Testing is a type of investigation to find out if there are any defect or error present in the software so that the errors can be reduced or removed to increase the quality of the software and to check whether it fulfils the specified requirement or not.

- The main objectives of software testing is to design the test in such a way that it systematically finds different types of error without taking much time and effort so that less time is required for the development of the software.

Common Testing Strategies are :-

(1) Black Box Testing

Black box testing is a type of software testing in which the tester is not concerned with the software internal knowledge or implementation details but rather focuses on validating the functionality based on the provided specification or requirement.

Types

↳ Functional Testing

Functional testing is defined as a type of testing that verifies that each function of the software application works with the requirement or specification.

↳ Regression Testing

Regression testing is the process of testing that modifies parts of codes and the parts that might get affected due to the modification.

↳ Non-functional Testing

Non-functional testing is defined as a type of testing to check non-functional aspects of the software. It is design to test the readiness of system

Advantages

↳ Efficient for implementing the test in larger project
↳ Test are easily reproducible
↳ Used to find contradiction in functional specification

Disadvantages

↳ Difficult to implement
↳ Sometime test failure cannot be detected
↳ Not reveal the error in control structure

② White Box Testing

- White Box testing is a software testing technique that involves testing the internal structure and working of software application
- The tester has access to the source code and uses this knowledge to design test case that can verify the correctness of the software at the code level
- white box testing is also known as structural testing or code based testing and is used to test software internal logic, flow and structure.
- White Box testing focuses on
  - ↳ Path checking
  - ↳ Output validation
  - ↳ Security testing
  - ↳ Loop testing
  - ↳ Data flow testing

Types

① Unit testing

checks if each part of or function of the application works correctly ensures the application meets design requirement during development.

② Integration testing

Examines how different parts of application work together. Ensures the component work well alone and together

③ Regression testing

Verifies that changes or update don't break existing functionality.

Advantages

- ↳ Early detection of defect
- ↳ Detection of complex defect
- ↳ Code optimization
- ↳ throughout testing

Disadvantage

- ↳ Testers need to have programming knowledge
- ↳ Testers need to have deep knowledge of code
- ↳ Missing functionalities cannot be detected.
- ↳ High chances of error in production.

③ Unit Testing

A unit test is a small piece of code that checks if a specific function or method in an application works correctly. It will work as function inputs and verifying the output

- Unit test should run one by one, it means that they do not depend on other system part.

- Unit testing strategies

↳ Logic check

Verify if the system perform correct calculation and follows the expected path with valid input

↳ Boundary Check

Test how the system handles typical, edge case and invalid input

↳ Error handling

Check the system properly handles the error

Types

① Manual Testing

Manual testing is like checking each part of a project by hand, without using any special tools. Developer do manual testing to see if their code works correctly.

② Automated Testing

Automated testing is a way of checking if software works correctly without needing lots of human effort. We use special tools made by people to run these test automatically.

Techniques of Unit Testing are :-

- ↳ Black Box testing
- ↳ White Box testing
- ↳ Gray Box testing

Advantages
- ↳ Early Detection of issue
- ↳ Improved code Quality
- ↳ faster development
- ↳ Reduce Time and Effort for later testing

Disadvantage
- ↳ Require time and efforts
- ↳ Dependence on developers.
- ↳ Difficulty in testing Interaction
- ↳ Maintenance overhead.

(4) Integration Testing

Integration testing is a software testing technique that focuses on verifying the interaction and data exchange between different component or modules of a software application.

- The goal of integration testing is to identify any problem or bugs that arise when different components are combined and interact with each other.
- Integration testing is typically performed after unit testing and before system testing.
- Integration testing is important because it verifies that individual software module or component work together correctly as a whole system.

- Integration testing approach
  - ↳ Big - Bang Integration testing
  - ↳ Bottom - up Integration testing
  - ↳ Top - down Integration testing
  - ↳ Mixed Integration testing

Application
- Identify the component
- Create a test plan
- Set up test environment
- Execute the test
- Analyze the results
- Repeat testing

Difference between Unit Testing & Integration Testing

Unit Testing
- Each module of software is tested separately
- Internal design of software is known
- It is performed first of all testing process.
- Unit testing is white box testing
- It is performed by developers.

Integration Testing
- All module of software is tested combined
- Internal design of software is not known.
- It is performed after unit testing and before system testing

- Integration testing is black-box testing
- It is performed by testers

⑤ System Testing

System testing is a type of software testing that evaluate the overall functionality and performance of a complete and fully integrated software solution.

- It test if the system meets the specified requirement and if it is suitable for delivery to the end-users.
- This type of testing is performed after Integration testing and before the acceptance testing.
- System testing detects defect within both the integrated units and the whole system.

Process
- ↳ Setup test Environment
- ↳ Generate test cases
- ↳ Generate testing data
- ↳ Execute test case
- ↳ Defect Reporting
- ↳ Regression testing
- ↳ Log defects
- ↳ Retest

Advantages
- ↳ Testers do not require more knowledge about programming
- ↳ Test environment is similar to real time production
- ↳ Verifies the overall functionality of the system
- ↳ Improves system reliability and Quality

Disadvantages
- ↳ This testing is time consuming.
- ↳ The cost of testing will be high
- ↳ Time consuming and expensive
- ↳ Requires proper planning, coordination and execution

⑥ Verification and Validation

Verification and validation is the process of investing whether a software system satisfies application and standard and fulfilled the required purpose. Barry Bohem described Verification and Validation

- Verification is the process of checking that software achives its goal without any bugs. It is the process to ensure whether the product that is developed is right or not.
- Validation is the process of checking whether the software product is up to the mark or in other words product has high-level requirement.
- Verification is also known as Static Testing and Validation is also known as Dynamic Testing.

Debugging Approaches

Debugging is the process of finding and resolving defect or problem within a computer program that prevent the correct operation of computer software or a system.

- following are the approaches for debugging :-

① Brute Force Method

This is the foremost common techniques of debugging. During this approach the program is loaded with print statement to print the intermediate values with hope that a number of written values can facilitate to spot the statement in error.

② Backtracking

During this approach, starting from the statement at which an error symptom has been discovered, the source code is derived backward till the error is discovered.

③ Cause Elimination method

In this approach, a listing of cause that may have contributed to the error symptom is developed and test are conducted to eliminate every error.

④ Program slicing

This technique is analogous to backtracking. Here search house is reduced by process slices. This is done to reduce the complexity of the code, making it easier to analyze, debug or understand.

Software Reliability & Quality Management

Software Reliability

Software reliability is defined as the probability of failure-free operation of a software system for a specified time in a specified environment.

- Software Reliability start with many fault in the system when first created.
- Software reliability cannot be predicted from any physical basis, since it depends completely on the human factors in design.
- The system needs to be tested to reduce faults.
- The complexity of software reliability is low.

Quality Concept

Software Quality shows how good and reliable a product is. It performs all function as laid out in the SRS document.

- factors of software Quality
  - ↳ Portability
  - ↳ Usability
  - ↳ Reusability
  - ↳ Correctness
  - ↳ Maintainability
  - ↳ Reliability
  - ↳ Efficiency

Software Quality Assurance

- Software Quality Assurance (SQA) is simply a way to assure quality in software. It is a set of activities that ensure process, procedure as well as standard are suitable for the project and implemented correctly.

- Software Quality Assurance is a process that works parallel to software development. It focuses on improving the process of development of software so that problems can be prevented before they become major issue.

- Software Quality Assurance Include
  - ↳ A Quality management approach
  - ↳ Formal technical reviews
  - ↳ Multi testing strategy
  - ↳ Effective software engineering technology
  - ↳ Measurement and reposting mechanism

Advantages
- ↳ SQA provide high quality software
- ↳ Saves time and cost for high quality application.
- ↳ It is benificial for better reliability
- ↳ Improves the quality of software

Disadvantage
- ↳ SQA requires skilled personnel.
- ↳ Cannot guarntee the elimination of all bugs
- ↳ SQA process can be complex.

Software Review

Software Review is a systematic inspection of a software by one or more individuals who works together to find and resolve error and defect in the software during the early stages of the Software Development Life Cycle (SDLC).

- Software review is an essential part of Software Development Lifeyc (SDLC) that help software engineering in validating the quality, functionality and other feature and component of software.

Objectives
- ↳ To improve the productivity of the development team
- ↳ To make the testing process time and cost-effict
- ↳ To make the final software with fewer defect
- ↳ To eliminate the inadequancies

Types

① Software Peer Review

Peer review is performed in order to examine or resolve the defect in software, whose quality is also checked by other member of team

② Software Management Review

Software Management Review evaluates the work status. In this section decision regarding downstream activities are taken.

③ Software Audit Review

Software Audit Review is a type of external review in which, who are not part of development team, organize an independent inspection of software product.

④ Formal Technical Review

Formal technical Review (FTR) is a software quality control activity performed by software engineers.

- It is an organized, methodical procedure for assessing and raising the stand of any technical paper, including software object.
- Finding flaws, making sure standard, improving the product or document are the main objectives of formal technical review (FTR).

Objectives

- ↳ Detect Identification
- ↳ Quality assurance
- ↳ Risk mitigation
- ↳ Knowledge Sharing
- ↳ Consistency
- ↳ Learning and Training

⑤ Software Configuration Management

Software configuration management (SCM) that involves managing and controlling changes to software product throughout their development and maintenance lifecycle.

- The primary goal of Software configuration management (SCM) is to ensure the integrity and consistency of software product as it evolves.

Process Involved in SCM

① Identification and Establishment

Identifying the configuration item from product and establishing relationship between items creating a mechanism to control and manage multiple levels.

② Version Control

Creating version / specification of the existing product to build new product with the help of SCM System.

③ Change Control

Controlling changes to configuration item. A change request (CR) is submitted and evaluated to asses technical merit

④ Configuration auditing

A software configuration audit complement the formal technical review of process and product.

⑤ Reporting

Providing accurate status and current configuration data to developers, tester through admin guides, users guide.

- Software configuration management (SCM) is a software engineering practise that focuses on managing the configuration of software system and ensuring software component are properly controlled, tracked and stored.

ISO 9000

The International standard organization (ISO) is a standard which serves as a contract between interdependent parties. It specifies guideline for development of quality system.

- ISO 9000 is a set of international standard that focuses on quality management system (QMS). It provides a framework for organization to ensure that they consistently meet customer requirement.

Features

↳ Document Control

All document concerned with development should be properly managed and controlled.

↳ Planning

Proper plans should be prepared and monitered.

↳ Review

All phases should be independently checked and reviewed.

↳ Testing

The product should be tested against specification.

↳ Organizational aspect

Various organizational aspect should be addressed.

Capability Maturity Model

Capability maturity model (CMM) was developed by Software Engineering Institute (SEI) at Carnegie Mellon University in 1987.

- It is not software process model. It is a framework that is used to analyze the approach and technique followed by any organization to develop software product.

Levels of CMM

There are 5 levels of Capability maturity model

- Initial
- Repeatable
- Defined
- Managed
- Optimizing

Level 5	Process change management Technology change management Defect Prevention	Optimizing
Level 4	Software Quality management Quantitative management	Managed
Level 3	Peer Reviews Inter group coordination Training program	Defined
Level 2	Project planning Configuration management Requirement management Software Quality Assurance	Repeatable
Level 1	NO KPA's ↳ Key process area	Initial

Level 1 - Initial
- NO KPI's defined
- Process followed are not well defined
- Very few or no process are described and followed.
- It is also called chaotic level

Level 2 - Repeatable
- Fundamental project management practises like tracking cost and schedule are established
- Size cost estimation method like COCOMO method are used.
- focuses on establishing basic project management.

Level - 3 - Defined
- Documentation of standard guideline and procedure take place
- Process and product qualities are not measured.
- focuses on the enhancement of knowledge and skills of team

Level 4 - Managed
- Quantitative goals are set for the organization
- Two kinds of metrices are composed
  - ↳ Product metrices
  - ↳ Process metrices.

Level 5 - Optimizing
- Process and product measurement data are evaluated for continous process improvement
- This is the highest level of process maturity in CMM

⑧ Risk management

Risk management is a systematic process of recognizing, evaluating, and handling threats or risk that have an effect on the finance, capital and overall operation of an organization.

- The main goal of risk management is to predict risks and find solution to deal with them successfully.

- Risk management is important because it helps organization to prepare for unexpected circumstance that can vary from small issue to major crisis.

Process

↳ Risk Identification

It refers to the systematic process of recognizing and evaluating potential threats that could negatively impact an organization

↳ Risk Analysis

It is the process of evaluating and understing the potential impact of identified risk on an organization. It helps to determine how serious a risk is.

↳ Risk Planning

It involves developing strategies and action to manage and identified risk effectively.

↳ Risk monitoring

It involves continuously tracking and overseeing identified risk to assess their status, changes of migitation strategies.

Advantage

↳ Help protect against potential loss
↳ Remove unexpected expenses
↳ Safeguard company reputation

Disadvantage

↳ Implementing risk management is expensive
↳ Some risk are hard to pridict
↳ Managing risk takes lots of time and resource.