Gunjan Bhawsinghka
D12B-04

Page No.
Date: / /

ASSIGNMENT NO. 2

1)

- SCM repository is the group of mechanisms and data structures which enable a software team to deal with modification in an efficient way.
- It offers the functionality of modern DBMS by the way of taking care of data integrity, sharing and integration.
- It provides a hub for purpose of integration of different available software tools.
- It helps to manage flow of software tools process & can implement uniform structure & format for software engineering related work products.
- To get these abilities, repository is described in terms of meta-model (used to identify way by which information is kept in repository, data can be accessed by tools).
- Repository that serves as a software engineering team should also:
  - incorporate with / directly support process management
  - support particular rules which govern SCM functions
  - provide interface to other tools which are related to SE.

2) RMMM plan -

- It is a part of software development plan or seperate document.
- It documents all work experience excuted as a part of risk analysis & used by project manages as a part of overall project plan.
- The risk mitigation & monitoring starts after project is started & documentation of RMMM is done.

Teacher's Signature.....

- In some software teams, risk is documented with help of risk information sheet (RIS).
- Basic steps in strategy are -
  - Risk mitigation
  - Risk monitoring
  - Risk management.

Risk mitigation - It means preventing risk to occur (Risk avoidance).
Steps - finding out risk
  - removing causes that are reason for risk creation
  - controlling corresponding documents time to time
  - conducting timely reviews to speed up work.

Risk monitoring - It is an activity used for project tracking. It has following primary objectives -
- to check if predicted risks occurs or not
- to ensure proper application of risk aversion steps defined for risk.
- to collect data for future risk analysis
- to allocate what problems are caused by which risks.

Risk management - It assumes that migitation activity failed risk is a reality.
- This task is done by project manager when risk becomes reality and causes severe problems.
- Main Objective of this is risk register which describes & focuses on predicted threats to project.

Teacher's Signature.....

Page No.
Date : / /

Categories of Risk -

1) Project risk: It is probable to slip if project risk is real, thus, cost of project will increase. It identifies potential schedule, resource, stakeholders & requirements problems & their impact on software project.
2) Technical Risk: If it is real, then implementation becomes impossible. It identifies potential design.
3) Business risk: If it is real, then it harms project. They are classified as - market risk, strategic risk, sales risk, management risk, budget risk.

3)

FTR -

- It stands for Formal Technical Review.
- It is an Umbrella activity.
- It is the meeting conducted by technical staff.
- It is one of the software Quality Assurance activities.

Objectives -

- FTR is used to uncover errors.
- It is used to verify that software under review meets its requirements.
- It ensures that predefined standard are used in software.
- It is used to make projects more manageable.
- It achieves software that is developed in uniform manner.

Steps in FTR -

1) Review Meeting - Every review meeting should be conducted by considering following constraints,
- Involvement of people (between 3 & 5 people should be involved in review)

Teacher's Signature.....

- advance preparation (It should occur)
- short duration (duration of review should be less than 2 hours).

Review meeting is attended by review leader, all reviewers and producers. One of the reviewers takes role of recorder. FTR begins with introduction of agenda & brief by producer. Producer then proceeds to walk through the work product, explaining material while reviewers raise issue based on their advance. At the end, attendees decide whether to,

- accept product without further modification
- reject product due to severe errors
- accept product provisionally.

2) Review reporting & record keeping - During FTR, a reviewer activity records all issues that have been raised. These are summarized at end of review meeting & review issues list is produced. In addition, formal technical review summary is completed.

3) Review guidance - Guidelines for conducting FTR must be established in advance. These guidelines must be distributed to all reviewer agreed upon & then followed. Eg, guidelines must be may include,

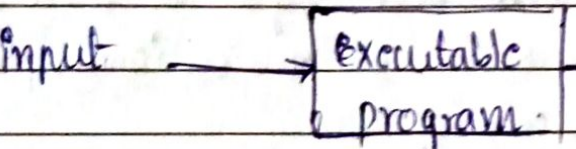
- review product not producer
- set an agenda & maintain it
- limit debate
- take written notes etc.

Teacher's Signature.....

Page No.
Date : / /

q)

- Software testing methods which test functionality of an application without knowledge, its internal structure, coding information & knowledge of internal paths of software is called black box testing.
- In this method, test cases are built based on what application is supposed to do.
- It is also known as behavioural testing or specification based testing.
- In Black Box testing, we concentrate on inputs provided to software product & output from software product without applying compulsion about knowledge of programming language used.



Black Box Testing

steps to perform black box testing -

- first requirements & specifications of system are analyzed
- testes select valid test data i.e. data for positive test scenario to verify whether application under test is able to process that data & give output as expected.
- Also, tester selects invalid test data for negative test scenario to verify the same.
- Tester state expected outputs for each test data.
- Software testes generates test cases with selected test data. Then, test cases are run.

Teacher's Signature.....

- Software tester performs comparison between actual output & expected output.
- Defect is detected, if actual result is not same as expected result.
- Then, detected defect is reported to developer.
- Developer can then fix that defect & application is retested by tester.

5)

Cohesion - It is a measure of degree to which elements of module are functionally related. It is degree to which all elements directed towards performing a single task are contained in component. A good software design with have high cohesion. Types - functional, sequential, communicational, procedural, temporal, logical, coincidental.

coupling - It is between two modules & is measure of degree of interaction or interdependent between two modules. A module having low coupling & high cohesion is said to be functionally independent of other modules. Types - data, stamp, control, external, common, content.

Teacher's Signature.....

c) Quality Metrics -
It can be defined as standards of measurement. It is used to measure quality of project. Simply, metric is a unit for describing an attribute. The goal of software metrics is to identify & control essential parameters that affect software development.
Different metrics for maintaining software quality:
1) Reliability - It is quality of performing consistency well. Software should be available for use & should satisfy following attributes: maturity, fault tolerance recoverability
2) Portability - It means system should be able to work in different hardware & software environments. It considers following attributes: installability, adaptability, conformance, replaceability
3) Efficiency - It means making optimal use of system resources & consider following attributes: time behaviour, resource behaviour
4) Maintainability - It is probability of performing a successful repair action within a given time. It should consider following: changeability, testability, stability
5) Functionality - It is range of operations that can be run on software. Following attributes are: interoperability, accuracy, security
6) Usability - It is degree to which application software is easy to use. Following attributes are: operability, understandability by end users.

Teacher's Signature.....

Page No.
Date

Software Reliability Metrics -

1) Project Management Metrics - A good management can result in creation of better products. Higher reliability can be achieved by using better risk management process, configuration management process & development process.

2) Process Metrics - Process metrics can be used to estimate, improve & monitor quality & reliability of software. It is proved that products quality is direct function of process.

3) Fault & Failure Metrics - Main motive of fault & failure metrics is ability to determine when software is approaching execution of failure free.

7) Integration Test -

- Helps in better test coverage tools improves test gaps
- Tests are more reliable & easy to isolate failures.
- Majorly helps to build real time use cases using end-to-end testing.
- Integration tests catch system-level issues, like broken database schema, mistaken cache etc.
- Test Automation develops need to think beyond writing test after product built but at early stage of development.
- Test are written from day one of development cycles.

Incremental Approach -

- In this, testing is done by joining two or more modules that are logically related.

Teacher's Signature.....

- Then other related modules are added & tested.
- Process continues until all of modules are joined & tested successfully.
- Incremental Approach is carried out by two different method :- Bottom up, top down approach.

Bottom-up : In this, each module at lower levels is tested with higher modules until all modules are tested. It takes help of drivers for testing.

Top-Down : In this, testing takes place from top to down following control flow of software system.

8) Quality Control -
- It is set of methods used by organizations to achieve quality parameters or quality goals & continually improve organization's ability to ensure that a software product meet quality goals.

Process :-
- Three class parameters that control software quality are products, processes, resources.
- Total quality control process consist, plan (where quality control processes are planned), Do (use defined parameters to develop quality), Check (stage to verify if quality of parameters), Act (take corrective action if needed & repeat work)

Quality Assurance -
- It is defined as auditing & reporting procedures used to provide stakeholders with data needed to make well-informed decisions.

Teacher's Signature.....

- It is process degree to which system meets specific requirements & customes expectations.

Quality Assurance Criteria :-
correctness, efficiency, flexibility, integrity, reliability, maintability, usability, testability.

9) Features of Good Test Cases -
- Easily identifiable with its name
- Simple & specific
- Reasonable probability of catching defect.
- Traceable to requirements
- Explicited pre-conditions
- Defined test data input
- Always list post condition
- Always list expected results
- Allows isolation with other test cases
- Assure quality
- Makes defects obvious

10) Boundary Value Analysis (BVA) -
- It is process of testing boundaries of input values.
- It is most commonly used technique in BBT.
- Basic idea is to select input values at their minimum, just above minimum, normal value, maximum value & just below maximum value.
- BVA always comes after equivalence class partitioning.

Teacher's Signature.....

Page No.
Date : / /

Eg. Suppose you have very important tool at office, accepts valid user name & password field to work on that tool & accepts minimum 8 characters & maximum 12 characters. Valid range 8-12, invalid range 7 or less than 7 & invalid range 13 or more than 13.

Equivalence class Partitioning -
- It reduces numbers of all possible inputs by dividing them into classes.
- It tests application thoroughly & avoids redundancy of input values.
- It can be applied at all levels of testing.
- It always performed before boundary value analysis technique is applied.
- BVA & equivalence partitioning are closely related & used together.

Eg, A text field permits only numeric characters. length must be 6-10 characters long. Partition should be like this,

0 1 2 3 4 5.	6 7 8 9 10	11 12 13 14
Invalid	Valid	Invalid

At times of testing, test 11 & 12 as invalid values & 7 as valid one. It is easy to test input ranges 0-10 but harder to test input ranges 2-600. Testing will be easy in case of lesser test cases but you should be very careful.

Teacher's Signature.....

11) Software Maintenance –

- Software doesn't wear out or get tired.
- However, it needs to be upgraded & enhanced to meet new user requirements.
- For such modifications in software systems, software maintenance is performed.
- Software maintenance is part of SDLC model.
- It is modification of software product after delivery.
- IEEE defines maintenance as a process of modifying of software system or component after delivery to correct faults to improve performance.
- Objective is to ensure that software is able to accommodate changes after system has been delivered & deployed.

Types –

1) Corrective Maintenance: It is used for normally 20% of all maintenance activities. It is type of maintenance in which errors are fixed when it is observed during use of software. The error may cause due to faculty software design, incorrect logic & improper code.
2) Adaptive Maintenance: It is used for normally 25% of all maintenance activities. It is implementation of changes in part of system. Changes may be hardware or OS environment. Eg. change in business rules, change in work, pattern have significant impact

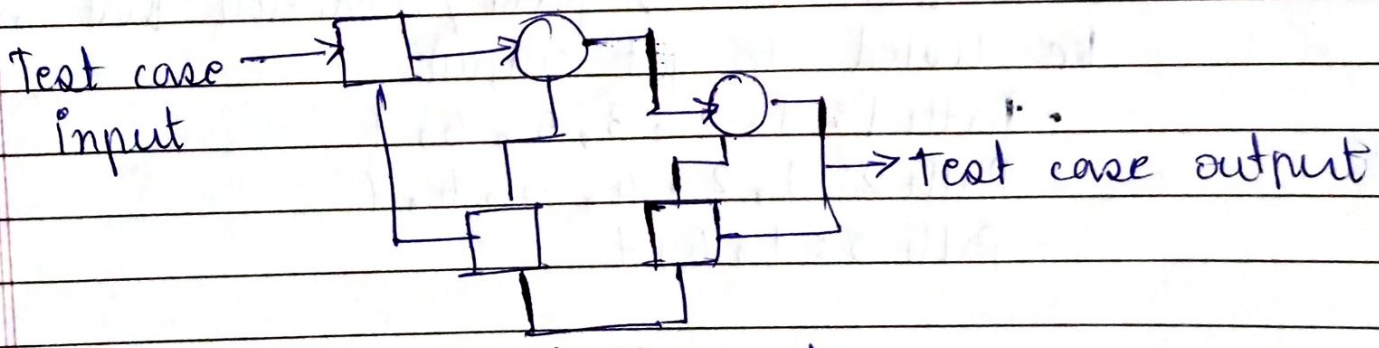
Teacher's Signature.....

3) Perfective Maintenance: It is used for normally 50% of all maintenance activities. It deals with implementing new or changed users requirements. In this, function & efficiency of code is improved. Eg. adding new repais in sales analysis system.

4) Preventive Maintenance: It is used for normally 5% of all maintenance activities. It involves performing activities to prevent occurence of errors. In this, complexity is minimized & quality of program is enhanced.

12) White Box Testing -

It is known as structural Testing or Glass Box Testing. It is performed to test program internal structure. To perform white box testing, testes should have enough knowledge of program internals along with purpose of developing software.



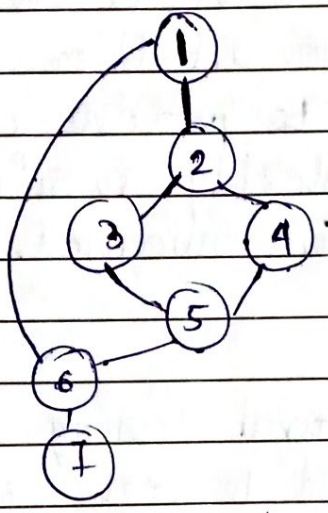
Basic Path Testing:

- In basis path testing test cases are generated.
- This technique is used to specify the basic set of execution paths.

Teacher's Signature.....

- To define basic set of execution path, complexity measure is used as guide.
- With increase in size of software, no. of execution path also increases.

Eg. graph matrices & flow graph notations.




```
If A=50
then if B>C
then A=b
else A=c
endif
endif
Print A
```

- In above example, we can see these are few conditional statement that is executed depending on what condition it suffice.
- Here, these are 3 path / condition that need to be tested to get output.
  - Path 1: 1, 2, 3, 5, 6, 7
  - Path 2: 1, 2, 4, 5, 6, 7
  - Path 3: 1, 6, 7