Debre Markos University
Institute of Technology
School of Computing, Software Eng A/program

Software Testing and Quality Assurance(SEng4051)

Chapter Seven

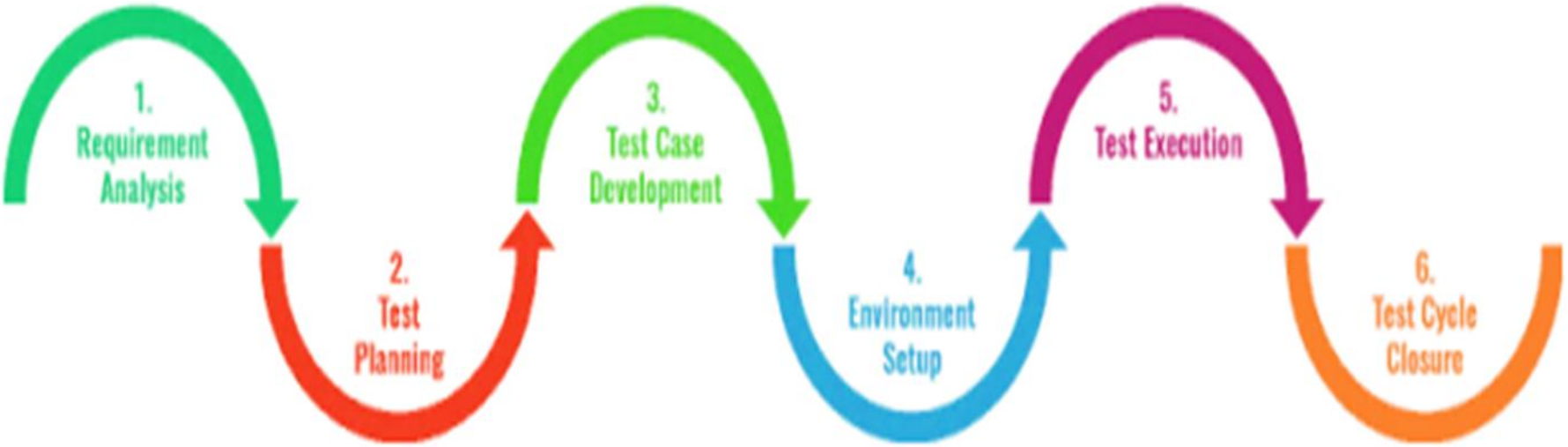
Test Process management and Improvement

Objective

- **After successful completion of this chapter, students will be able to understand**
  - ✓ Software Testing Process (revision)
  - ✓ Test Management
  - ✓ Test Estimation

Software Testing Process (revision)

- A process is a **series** of activities performed to fulfill a purpose and produce a tangible output based on a given input.
- **Software testing life cycle** defines the stages in testing of software.



Test Management

- **Test Management** is a process of managing the testing activities in order to ensure high quality and high-end testing of the software application.
- The method consists of organizing, controlling, ensuring **traceability** and **visibility** of the testing process in order to deliver the high quality software application.
- It ensures that the **software testing process** runs as expected.

Test Management Process

- **Test Management Process** is a procedure of managing the software testing activities from start to the end.
- The test management process provides planning, controlling, tracking and monitoring facilities throughout the whole project cycle.
- The process involves several activities like test planning, designing and test execution.

Cont...

- There are two main Parts of Test Management Process: –

➢ **Planning**

  - Risk Analysis
  - Test Estimation
  - Test Planning
  - Test Organization

➢ **Execution**

  - Test Monitoring and Control
  - Issue Management
  - Test Report and Evaluation

I, Planning: 1. Risk Analysis and Solution

- **Risk** is the probability of occurrence of an undesirable event.
- Risk should be identified and corresponding solutions should be determined before the start of the project.
- Risk Analysis is the first step which Test Manager should consider before starting any project. Because all projects may contain risks, early risk detection and identification of its solution will help Test Manager to **avoid** potential loss in the future & save on project cost.

Cont...

- **How to Perform Risk ANALYSIS?**
- It's a 3-Step process
  - Identify the Risks
  - Analyze Impact of each Identified Risk
  - Take counter measures for the identified & Analyzed risk

Step 1) Identify Risk

- Risk can be identified and classified into 2 types in software product

I)Project Risk

- Project risk can be defined as an **uncertain** event or activity that can impact the project's progress.
- There are primarily 3 categories of Project Risks
    - **Organizational Risk**
    - **Technical Risk**
    - **Business Risk**

Cont...

❑ Organizational Risk

- It is a risk related to your **human resource** or your Testing team. **For example**:-
  - ➢ in your project, lack of technically skilled members is a risk.
  - ➢ Not having enough manpower to complete the project on time is another risk.
- To identify the Organizational Risk, you should make a list of few **questions** and answer them as self-exercise.
  - ➢ Is this a well-organized Team?
  - ➢ Does each team member has the skill to do his/her job?
  - ➢ Compare to project size and schedule, do we have enough human resource to finish this project at the deadline?

Cont...

Technical Risk

- Technical Risk is the probability of loss incurred during the **execution** of a technical process such as wrong testing procedure.

Example:-

- The Testing **environment** may not be setup properly like real business environment.

Business Risk

- The risk involves an **external** entity. It is the risk which may come from your company, your customer but **not** from your project.
- Example:-
  - Your project **budget** may cut by half because of business situation

II) Product Risk

- **Product risk** is the possibility that the system or software might fail to satisfy or fulfill the expectation of the customer, user, or stakeholder.

Example:-

- The software skips some **key** function that the customers specified in the users' requirement
- The software is **unreliable** and frequently **fails** to work.
- Software **fail** in ways that cause **financial** or other damage to a user or the company that uses the software.
- The software has problems related to a particular quality characteristic such as **security**, **reliability**, **usability**, **maintainability** or **performance**.

Step 2) Analyze the impact of the risk occurring

- Each risk should be classified on the basis of following two parameters
  - ✓The **probability** of occurrence
  - ✓The **impact** on the project

Priority = Probability * Impact

	Probability	**Impact**
High (3)	Has very high probability to occur, may impact to the whole project	Cannot continue with project activity if it is not solved **immediately**
Medium (2)	50% chance to occur	Cannot continue the project activity if it is not solved
Low (1)	Low probability of occurrence	Need to solve it but it is possible to take alternative solution for a while

Cont...

- Based on the above priority you can take the Risk Mitigation in Testing or counter measures.

Priority	**Risk Management**
High 6-9	*Take mitigation action immediately and monitor the risk every day until its status is closed.*
Middle 3-5	*Monitor the risk every week at internal progress meeting*
Low 1-2	*Accept the risk and monitor the risk on milestone basis*

Step 3) Take COUNTER MEASURES to mitigate the risk

- This activity is divided into 3 parts.

❑ Risk response

- The project manager needs to choose strategies that will reduce the risk to minima
  - ➢ Avoidance
  - ➢ Reduction
  - ➢ Sharing
  - ➢ Accept

❑ Register Risk

- All the risk must be recorded, documented and acknowledged by project managers, stakeholder and the project member.

Monitor and Control Risk

- Risks can be monitored on a continuous basis to check if any changes are made.
- New risk can be identified through the constant monitoring and assessing mechanisms.

2. Test Estimation

- Test Estimation is approximately determining
  - **how long** will this testing take?
  - **How much** will it cost?
- Accurate test estimates lead to better planning, execution and monitoring of tasks under a test manager's attention.
- Allow for more accurate scheduling and help realize results more confidently.

3. Test Planning

- A test plan gives **detailed** testing information regarding an upcoming testing effort, including:
  - Test Strategy
  - Test Objective
  - Exit /Suspension Criteria
  - Resource Planning
  - Test Deliverables

4. Test Organization

- **Test Organization in Software Testing** is a procedure of defining roles in the testing process.
- It defines who is responsible for which activities in testing process. Test functions, facilities and activities are also explained in the same process.
- The competencies and knowledge of the people involved are also defined however everyone is responsible for quality of testing process.

II, Execution: 1. Test Monitoring and Control

- Test Monitoring and Control is the process of overseeing all the metrics necessary to ensure that the project is running well, on schedule, and not out of budget.
- **Monitoring** is a process of collecting, recording, and **reporting** information about the project activity that the project manager and stakeholder needs to know
- To Monitor, Test Manager does following activities
  - **Define** the project goal, or project performance standard
  - **Observe** the project performance, and compare between the actual and the planned performance expectations
  - **Record** and **report** any detected problem which happens to the project

Cont...

- Project **Controlling** is a process of using data from monitoring activity to bring actual performance to planned performance.
- In **Controlling**, the Test Manager takes action to correct the deviations from the plan.
- In some cases, the plan has to be **adjusted** according to project situation.

2. Issue Management

- **Issue Management** is the process to make others aware of the problem and then resolve it as fast as possible

3. Test Report & Evaluation

- The project has already completed. It's now time for look back what you have done.
- "Test Evaluation Report" describes the results of the Testing in terms of Test coverage and exit criteria.

Test Estimation

- Estimation can be used to predict how much effort with respect to time and cost would be required to complete a defined task.
- Here's a list of Estimation Techniques for testing software:
  - Work Breakdown Structure (WBS)
  - 3-Point product Estimation Test
  - Function point and testing point breakdown
  - Wideband Delphi method
  - Use-case Methodologies etc.

Work Breakdown Structure (WBS)

- The essence of this technique is to divide a complex test project into small components.
- The project is broken down into sub-modules;
- Each sub-module, in turn, is divided into functionalities, which are split into sub-functionalities.

Three-Point Estimation

- This is a statistical method, but it also breaks down the task into subtasks (in this it is similar to WBS).
- Then, three possible scenarios should be estimated for each sub-task.
- **The best case:** assuming that:
  - you have a talented team,
  - all of the necessary resources,
  - no problem occurs and everything goes right
- you can complete the task, for example, in *100 man-hours (B)*. This is an optimistic scenario.
- **The most likely case:** assuming that:
  - you have a good team
  - enough resources
  - almost everything goes right, although some problems may occur,
- you can complete the task in *150 man-hours (M)*. This is a normal scenario.

Cont...

- **The worst case:** assuming that:
  - your team is not experienced
  - everything goes wrong
- you have to solve numerous problems, you can complete the task in *200 man-hours (W)*. This is a pessimistic scenario.
- Thus, you have three values:
  - B = 100
  - M = 150
  - W = 200

Cont...

- Now, you can calculate the average value for the test *estimation (E)* using the following formula:
- **E = (B + 4M + W) / 6**
- E = (100 + 4 x 150 + 200) / 6 = 150 man-hours
- As the average value may fluctuate a little bit, to be more accurate, you need to calculate *standard deviation (SD)* — the limits within which E may change. The formula is as follows:
- **SD = (W – B) / 6**
- SD = (200 – 100) / 6 = 16.7 man-hours
- You can present the final estimate as this: the team needs 150 +/- 16.7 (133.3 up to 166.7) person-hours to accomplish the sub-task.

Function Point Analysis (FPA)

- It estimates the value of the **total effort** that can be considered as **time or cost** of the task.
- FPA is based on specification documents, such as SRS document or Design.
- Again, as with WBS, the project is split into modules.
- Each module — depending on its complexity — is assigned a *functional point (FP)*.
- Simple tasks get lower points, difficult tasks get higher points.

Total Effort = Total FP x Estimate Defined per FP

Cont...

- Let's consider the total effort with respect to cost and take the estimate defined per FP as equal to **$100/points**.
- The whole project is divided into three groups of modules:
- Complex modules (FP is 5) — 3 pieces
- Medium modules (FP is 3) — 8 pieces
- Simple modules (FP is 1) — 4 pieces

Complexity	**Number of Modules**	**FP for each module**	**Total FP for each module**
Complex	3	5	15
Medium	8	3	24
Low(Simple)	4	1	4
			Total FP = 43

- Total Effort = 43 * $100 = $4,300.

Wideband Delphi method

- This is one of the most widely used testing estimation techniques based on surveys of the experts involved in the testing process.
- The essence is that a panel of experts discuss the given task under the guidance of a manager and make anonymous personal forecasts (how many man-hours this task will take), providing the reasons for their opinions.
- As a rule, after the first round, the range of answers is quite wide. Then, the experts are encouraged to revise their answers taking into account other members' judgments.
- Several rounds may take place until the range of answers decreases and the average value can be calculated.

Use-Case Point Method

- **Use-Case Points** (**UCP**) method is an estimation technique used for measuring software with different use cases.
- The Use-Case Points counting process has the following steps –
  - Calculate unadjusted UCPs
  - Adjust for technical complexity
  - Adjust for environmental complexity
  - Calculate adjusted UCPs

THE END

THANK YOU!!!