

Chapter-7

Test Management

Project Success:

Successful project should have the following:

- a well defined scope and agreed understanding of intended outcomes.
- active management of risks, and issues.
- timely decision making
- clean and short lines of reporting.
- ongoing commitment and support from senior management.
- trained & experienced project team along with project leader.
- Well defined and visibly managed process.
- thorough planning & strong leadership.
- good and healthy communication between test members.
- project should have realistic objectives.

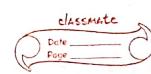
To increase the chances of success, projects should have the following:

- ✓ clear and well managed processes.
- ✓ a clearly defined purpose and limits.
- ✓ shared understanding of intended outcomes
- ✓ realistic objectives
- ✓ good management of risks and problems
- ✓ thorough planning
- ✓ timely decision making supported by short, clean lines of reporting.
- ✓ strong leadership.

- ✓ commitment and support from senior management.
- ✓ a senior person with overall
- ✓ a trained and experienced project manager who is suited to the particular project.
- ✓ a trained and experienced project team.
- ✓ clearly defined jobs and responsibilities.
- ✓ good communications.

The most common reasons for project failure are:

- ✓ following a method without thinking
- ✓ being too confident of success
- ✓ not having enough contribution from those with an interest.
- ✓ having unrealistic expectations
- ✓ partners having too little involvement.
- ✓ poor communication
- ✓ poor project specification
- ✓ not enough resources
- ✓ having the wrong person involved in the project
- ✓ having too much reliance on one person.
- ✓ not enough planning - unrealistic time and resource estimates - unclear or unmeasurable project objectives - changing project objectives during the project
- ✓ failure to manage risks and problems
- ✓ inexperienced project manager.
- ✓ unexperienced project team.
- ✓ weak leadership.



In house Project

- It is the project developed by company that is going to use it or created for generic organization.
- This type of project has soft deadline.
- This type of project does not have any fixed schedule and can be developed at any time when the development team is free.
- The product should be marketed since it is not developed for one specific client.

Advantages:

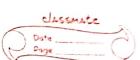
- ✓ security & control
- ✓ foresight context
- ✓ familiarity with company procedures
- ✓ commitment
- ✓ program evaluation is easy.

Disadvantages:

- ✓ limited experience
- ✓ knowledge stagnation
- ✓ fluctuating workload volume.

Project for clients

- It is the project that is developed for the specific customer based on their needs and demands and following their orders.
- Here, in this project, the organization and the customer first agree on the specification and contract is made.



which includes the budget, deadline and features to be added in the project.

- Based on the details provided by the contract, the development team develop the project. Here, they cannot include their own thoughts and extra features according to their convenience.

Outsourcing Project:

- Exporting project to other companies and pay them for building the project.

Project Complexity

Some of the causes of project complexity are:

- ① Details - number of variables and interfaces
- ② Ambiguity - lack of awareness of events and causality, its cause.
- ③ Uncertainty - inability to pre-evaluate actions.
- ④ Unpredictability - inability to know what will happen.
- ⑤ Dynamics - rapid rate of change.
- ⑥ Social structure - number & types of interactions.
- ⑦ Interrelationships - many interdependences and interconnections exist.

How to reduce complexity?

- Assign project leaders based on the project profile.
- Select the project cycle based on the project profile.
- Select appropriate management technique based on the project profile.
- Design and build adaptive business solution.

Imp Roles and Responsibilities of Test Leader.

- ① Estimation of testing resources. (Time, budget, manpower)
- ② Negotiate with management to acquire necessary resources.
- ③ Decide when to use test automation.
- ④ Lead the test analysis, design, implementation and execution of test cases and procedures.
- ⑤ Ensure proper test environment before executing test.
- ⑥ Schedule and prioritize test for execution.
- ⑦ Write summary report of test.

Imp Roles and Responsibilities of a Tester.

- ① Contribute in setting up the test environment.
- ② Contribute in making test plans, test cases, test procedure, test data etc.
- ③ Execute and log the test.
- ④ While testing, evaluate the result and document problems found.
- ⑤ Monitor testing and test environment.
- ⑥ Review each others' work.

Test Plan:

- Test plan is the project plan for the testing work to be done.
- It is not a collection of test cases or test procedures; in fact most of our test plans do not address that level of detail.

Purpose & importance of test plans:

- ① It guides our thinking → deal with:
 - forces to confront the challenges that await us and focus our thinking in important topics.
- ② Serves as means of communication with other members of the project team.
- ③ It helps us to manage change. As the project evolves, and situation change, we adapt our plans.

Things to keep in mind while planning tests

- ① What is in scope and what is out of scope for this testing effort?
- ② What are the test activities? objectives?
- ③ What are the important project and product risks?
- ④ What constraints affect testing? (e.g. budget, deadlines)
- ⑤ What is most critical component for this product and project?

classmate
Date _____
Page _____

classmate
Date _____
Page _____

- ⑥ Which aspects of the product are more (or less) testable?
- ⑦ What should be the test schedule and priority?
- ⑧ How to split testing work into various levels (e.g. unit, integration, system & acceptance)

Entry Criteria

Entry criteria is used to determine when a given test activity should start.

Entry criteria is defined in test plan during planning phase.

Example:

- Verify if the Test environment is available and ready for use.
- Verify if test tools installed in the environment are ready for use.
- Verify if Testable code is available.
- Verify if Test Data is available and validated for correctness of Data.
- Verify if the risks associated with testing are studied.
- Verify if test plans and test cases are reviewed and approved.
- Verify if all the testing resources (manpower, budget, time) are available.

~~10~~

Exit Criteria

- Exit criteria is used to determine whether a given test activity has been completed or not.
- Exit criteria is defined in the test plan during the planning phase.

Example:

- Verify if all test planned have been run.
- Verify if requirement coverage have been met.
- Verify if there are no critical defects left.
- Verify if all high risk areas are completely tested.
- Verify if software development activities are completed within estimated cost.
- Verify if software development activities are completed within estimated time.
- Verify if closure reports are signed off.

classmate
Date _____
Page _____

classmate
Date _____
Page _____

Test Estimation

- Testing is a process rather than a single activity. Hence, we need to breakdown a testing project using the fundamental test process like:

- planning & control
- analysis & design
- implementation & execution
- evaluating exit criteria & reporting
- test closure.

- Within each phase we identify activities and within each activity we identify tasks and perhaps subtasks.

Estimation techniques in software testing:

The ~~techniques~~ techniques for estimation are:

① Work Breakdown:

- break down test project into modules.
- " modules into sub modules.
- " submodules into functionalities.
- functionalities into subfunctionalities.
- figure out the number of tasks the team need to complete.
- Estimate the effort for each task.
- Estimate the duration for each task.

② Usecase Point Method.

- count the total no. of actors.
- count the total no. of usecases
- Estimate the effort and duration for each use case.

③ Percentage Distribution:

- all the phases of SDLC are assigned effort in percentage. It may be based on past data from similar project.
eg:

Phase	% of Effort
Requirement Analysis	5%
Design	10%
Coding	15%
Testing	25%
Documentation	25%
Deployment	5%
Maintenance	15%

Next, all the testing phases are assigned effort in percentage. example:

System Testing	% of effort
Functional Testing	65
Non-functional Testing	35

④ Experience based Testing Estimation Technique

- This estimation method assumes that metrics are collected from similar projects carried out in the past.
- Estimation is carried out based on those metrics by experts.

DSI PBM Lcm

Factors affecting test effort:

L1 DSP MBT
Late DSP man Bahadur Thapar

* Project Documentation

- A detailed documentation contains accurately specified test cases, which results in delays.

* Increasing size product.

- Large project requires large number of tests to be performed and increases difficulty to test.

* Life cycle Model

- The incremental model requires more regression testing than V model.

* Time Pressure

- If more tests are needed to be performed in less amount of time, then all the test cases cannot be executed.

- * People factor
 - The skills and experience of the people involved in testing determines the testing effort.

- * Budget
 - The amount of budget allocated for testing determines the testing effort.

- * Management

- The effectiveness of management of test leader & test manager determines the testing effort.

Test Strategy

- It is the most powerful factor in the success of the test effort and the accuracy of the test plans and estimates.

Test Strategy

- It is the most powerful factor in the success of the test effort and the accuracy of the test plans and estimates.

Major types of Test Strategies:

- ① Risk-Based strategy:
 - It involves performing the risk analysis using project documentation and stakeholders input.

② Model-Based strategy

- If the behaviour of the system under test conforms to the predicted by the model, the system is said to be working.

③ Dynamic Strategy

- It is concerned on finding as many defects as possible during test execution.

④ Consultative or directed

- To guide the testing effort, the suggestions and help is taken from non-testers like developers, customers etc.

⑤ Regression-based

- Here a set of automated procedures are used to allow them to detect regression defects.

⑥ Process or standard compliant

- Testing is carried out by maintaining the standards like IEEE.

Some of these strategies are more preventive and some are reactive.

Preventive approaches

- Here test basis is analysed prior to test execution to identify problems in test basis.

Reactive approaches

- Here, the strategies focus on test execution period to detect the error. We cannot find error until we execute ~~test~~ the system on actual environment.

Topic

Test Monitoring

Purposes:

1. To give the test team and the test manager feedback on how the testing is going, allowing opportunities to guide and improve the testing.
2. Provide the visibility of test result to the project team.
3. Measure the status of testing, test coverage and test items.
4. Evaluate exit criteria to determine whether the test work is done.
5. Gather data for use in estimating future test efforts.

For small projects, the test leader can gather test progress monitoring information manually using documents, spreadsheets and simple databases.

For large projects, distributed project and long-term test efforts, automated tools can be used for efficient and consistent data collection.

Ways to keep the record of test progress:

- ① Use table with the information about test runs rows and columns with status, date, bug ID, duration, comment etc.

classmate
Date _____
Page _____

classmate
Date _____
Page _____

- (2) use graph that plots the total number of defects opened and closed over the course of the test execution. It can show defect density / failure rate.
- (3) compare the number of unresolved defects with the size of the product. Once the number of defects unresolved defects falls below some predefined threshold, then the product is said to have met the ~~exit~~ defect density exit criteria.
(example of threshold → three defects per million lines of code)

Test Control

In test control, the actual progress is compared against the plan and the status of executing the plan including any deviation is seen, then it is reported.

- To bring back the deviated project, test control is used which helps to bring the project back under control by performing following activities:
• Prioritizing the testing efforts
• Revisiting the test schedules
• Reorganizing test environment
• Reprioritizing the test cases.

Example:

A portion of the software under test is scheduled to be delivered late but market condition dictate that release date must be earlier.

Here, test control comes into play to re-prioritize the tests so that the product could be released earlier.

Also rescheduling is done by test control by allocating resources for tests during weekends and nights.

7.5 Configuration Management.

- Configuration management is all about the items that make up the software or system. These items include source code, test scripts, hardware, data and both development and test documentation.
- Configuration management is also about making sure that these items are managed carefully and thoroughly during the entire project and product life cycle.
- Configuration management has a number of implications for testing.
- ✓ The configuration management allows the tester to manage their testware and test results.
- It also helps to deliver test releases into test environment.

- It also allows us to keep the record of what is being tested to the underlying files and components that make it up.

- The practices used for configuration management includes - ~~change control~~ baseline.

Baseline:

Baseline is a measurement that defines completeness of a phase in SDLC.

A phase is baselined when all activities involved in that phase are finished and well documented.

Configuration management takes care of occurrence of any change after a phase is baselined.

Revision control ~~change control~~.

Change control.

Configuration Management ensures that the changes made to the software system are consistent and made as per organizational rule and regulations.

A change in the configuration of product goes through following steps:

- I) Identification: The change request is identified and documented.
- II) Validation: Change request is validated.

- ④ Analysis: The impact of change request is analyzed.
- ⑤ Review: The request is accepted or rejected based on the impact on the system and seriousness of request.
- ⑥ Execution: Appropriate action is taken to execute the change request.
- ⑦ Close requests: Change is verified to be correctly implemented and merged with the rest of the system. The change is documented properly.

7.6. Risk and Testing: (Project Risk, Product Risk)

Risk / Threat

- Risks are the possible problems that might endanger the objectives of the project stakeholders.
- It is something that has not happened yet and may never happen.
- It is the possibility of a negative or undesirable outcome.

Types/categories

- ① Product Risk
- ② Project Risk

① Product Risk

- It is the possibility that the system or software might fail to satisfy some reasonable specification of the customer, user or stakeholders.

Example:

- i) If the software skips some key function that the customer specified.
- ii) If the software is unreliable and frequently fails to work.
- iii) If software fail in ways that cause financial or other damage to a user.
- iv) If the software has problems related to a particular quality characteristic, which might not be functionality but rather security, reliability, usability, maintainability or performance.

So, product risk analysis should be done:-

During analysis:-

- First, consider both likelihood of occurrence of the risk and the impact of the risk.
- Second, perform risk analysis as early as possible and follow up on it.

① Project Risks

- These are the risk involved during the development phase before the delivery of the product.
- Some of the project Risks are:
 - i) late delivery of test items
 - ii) Delay in testing test items.
 - iii) Insufficient Budget for testing.
 - iv) Insufficient time for testing.
 - v) Incomplete test cases
 - vi) Improper selection of testing techniques to be used. (not following leading practice)
 - vii) Tester Ego.
 - viii) lack of experienced knowledge in tester.
 - ix) Improper management for testing
 - x) Improper testing environment.
 - xi) skill, training and staff shortage.
 - xii) improper communication.
 - xiii) low quality of design, code, test data and tests.
- The actions that we can take against project & product risk are:-
 - i) Mitigate - take steps in advance to reduce the possibility and impact of the risk.
 - ii) Contingency - Have a plan in place to reduce the possibility of the risk to become an outcome.
 - iii) Transfer - convince some other member to reduce the probability or accept the impact of risk.

- iv) Ignore: Ignore the risk when the impact of risk is low in the project.

Risk Based Testing:

- It is the idea to organize our testing efforts in a way that reduces the product risk when the system is deployed

How to perform risk based testing?

1. Make a prioritized list of risks
2. Perform testing that explores each risk.
3. As a risk is mitigated, new ones emerge, so adjust the test effort to stay focused.

7.7 Incident Management:

Incident in software testing:

- while executing a test, we might observe that the actual results may vary from expected results. When the actual result is different from the expected result then it is called as incidents, bugs, defects, problems or issues.
- To be specific, an incident is basically any situation where the system exhibits questionable behaviour; but we refer to an incident as a defect only when incident cause some problem in the item we are testing.
- The causes of incidents are:
 - i) misconfiguration or failure of test environment
 - ii) corrupted test data
 - iii) invalid expected results.
 - iv) tester mistakes.

Role of incident management?

(i) Incident logging:

- We log incident so that we can keep the record of what we observed and can follow up the incident and what can be done to correct it.
- Also, we log incidents to manage incidents found during development and reviews because it gives useful information about the early and cheaper defect detection and removal activities.

(ii) Incident Reporting:

Reporting is done to fix the issues ASAP ^{to} the respective owners in order to make the system qualitative and gain user's trust.

objectives:

- i) Provide developers and other parties with feedback about problem
- ii) Provide test leaders a means of tracking the quality of the system ^{understand the} progress of testing.
- iii) Provide ideas for test process improvement.