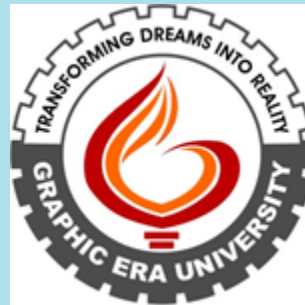


Agile Software Development (TCS 855)

Unit-IV Agile Testing
Regression Test and Risk Based Testing



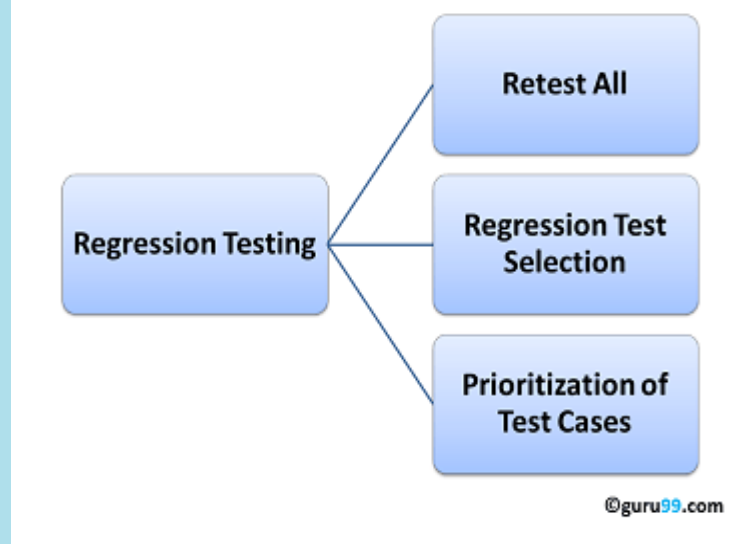
Prof.(Dr.) Santosh Kumar
Department of Computer Science and Engineering
Graphic Era Deemed to be University, Dehradun

Regression Testing: Definition, Benefits, and Challenges of Regression Testing

Definition of Regression Testing

- It is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.
- Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.
- This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.
- The **Need of Regression Testing** mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not.
- Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

- In order **to do Regression Testing** process, we need to first debug the code to identify the bugs.
- Once the bugs are identified, required changes are made to fix it, then the regression testing is done by selecting relevant test cases from the test suite that covers both modified and affected parts of the code.
- Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary.
- Regression Testing can be carried out using the following techniques:



Retest All

- This is one of the methods for Regression Testing in which all the tests in the existing test bucket or suite should be re-executed. This is very expensive as it requires huge time and resources.

Regression Test Selection

- **Regression Test Selection** is a technique in which some selected test cases from test suite are executed to test whether the modified code affects the software application or not. Test cases are categorized into two parts, reusable test cases which can be used in further regression cycles and obsolete test cases which can not be used in succeeding cycles.

Prioritization of Test Cases

- Prioritize the test cases depending on business impact, critical & frequently used functionalities. Selection of test cases based on priority will greatly reduce the regression test suite.

Selecting test cases for regression testing

- It was found from industry data that a good number of the defects reported by customers were due to last minute bug fixes creating side effects and hence selecting the **Test Case** for regression testing is an art and not that easy.
- Effective Regression Tests can be done by selecting the following test cases –
 - i. Test cases which have frequent defects
 - ii. Functionalities which are more visible to the users
 - iii. Test cases which verify core features of the product
 - iv. Test cases of Functionalities which has undergone more and recent changes
 - v. All Integration Test Cases
 - vi. All Complex Test Cases
 - vii. Boundary value test cases
 - viii. A sample of Successful test cases
 - ix. A sample of Failure test cases

Regression Testing Tools

- If your software undergoes frequent changes, regression testing costs will escalate.
- In such cases, Manual execution of test cases increases test execution time as well as costs. Automation of regression test cases is the smart choice in such cases.
- The extent of automation depends on the number of test cases that remain re-usable for successive regression cycles.
- Following are the most important tools used for both functional and regression testing in software engineering:

1) Avo Assure

Features

- Autogenerate test cases with a 100% no-code approach
- Test across the web, desktop, mobile, ERP applications, Mainframes, associated emulators, and more with a single solution.
- Enable accessibility testing
- Execute test cases in a single VM independently or in parallel with Smart Scheduling
- Integrate with Jira, Jenkins, ALM, QTest, Salesforce, Sauce Labs, TFS, etc.
- Define test plans and design test cases through the Mindmaps feature

2) Eggplant

- Eggplant's AI-driven test automation streamlines regression testing through prioritization of test cases and minimization of test maintenance.

Features:

- AI-driven test execution enables Eggplant to test the most important areas of each release.
- Reuse testing models and scripts to test multiple versions with one set of assets.
- Reduce the burden of test maintenance through self-healing functional tests.
- Understand and focus in on problematic areas of your application that put your release at risk.
- Discover bugs that you would otherwise miss through automated exploratory testing.
- Reduce the time required to test key functionality of applications after updates.

3. Selenium: This is an open source tool used for automating web applications. Selenium can be used for browser-based regression testing.

- **Quick Test Professional (QTP):** HP Quick Test Professional is automated software designed to automate functional and regression test cases. It uses **VBScript** language for automation. It is a Data-driven, Keyword based tool.
- **Rational Functional Tester (RFT):** IBM's rational functional tester is a Java tool used to automate the test cases of software applications. This is primarily used for automating regression test cases and it also integrates with Rational Test Manager.

- **Regression Testing and Configuration Management**
- Configuration Management during Regression Testing becomes imperative in Agile Environments where a code is being continuously modified. To ensure effective regression tests, observe the following :
- Code being regression tested should be under a configuration management tool
- No changes must be allowed to code, during the regression test phase. Regression test code must be kept immune to developer changes.
- The database used for regression testing must be isolated. No database changes must be allowed

Difference between Re-Testing and Regression Testing:

- Retesting means testing the functionality or bug again to ensure the code is fixed. If it is not fixed, Defect needs to be re-opened. If fixed, Defect is closed.
- Regression testing means testing your software application when it undergoes a code change to ensure that the new code has not affected other parts of the software.

Challenges in Regression Testing:

- Following are the major testing problems for doing regression testing:
- With successive regression runs, test suites become fairly large. Due to time and budget constraints, the entire regression test suite cannot be executed
- Minimizing the test suite while achieving maximum **Test coverage** remains a challenge
- Determination of frequency of Regression Tests, i.e., after every modification or every build update or after a bunch of bug fixes, is a challenge.

Risk Based Testing: Approach, Matrix, Process & Examples

Risk Based Testing

- **Risk Based Testing (RBT)** is a software testing type which is based on the probability of risk. It involves assessing the risk based on software complexity, criticality of business, frequency of use, possible areas with Defect etc. Risk based testing prioritizes testing of features and functions of the software application which are more impactful and likely to have defects.
- Risk is the occurrence of an uncertain event with a positive or negative effect on the measurable success criteria of a project. It could be events that have occurred in the past or current events or something that could happen in the future. These uncertain events can have an impact on the cost, business, technical and quality targets of a project.
- Risks can be positive or negative.
 - ✓ **Positive risks** are referred to as opportunities and help in business sustainability. For example investing in a New project, Changing business processes, Developing new products.
 - ✓ **Negative Risks** are referred to as threats and recommendations to minimize or eliminate them must be implemented for project success.

Risk based Testing Implementation

- Risk based testing can be implemented in
- Projects having time, resource, budget constraints, etc.
- Projects where risk based analysis can be used to detect vulnerabilities to SQL injection attacks.
- Security Testing in Cloud Computing Environments.
- New projects with high risk factors like Lack of experience with the technologies used, Lack of business domain knowledge.
- Incremental and iterative models, etc.

Risk Management Process

- Let's now understand the steps involved in Risk Management Process

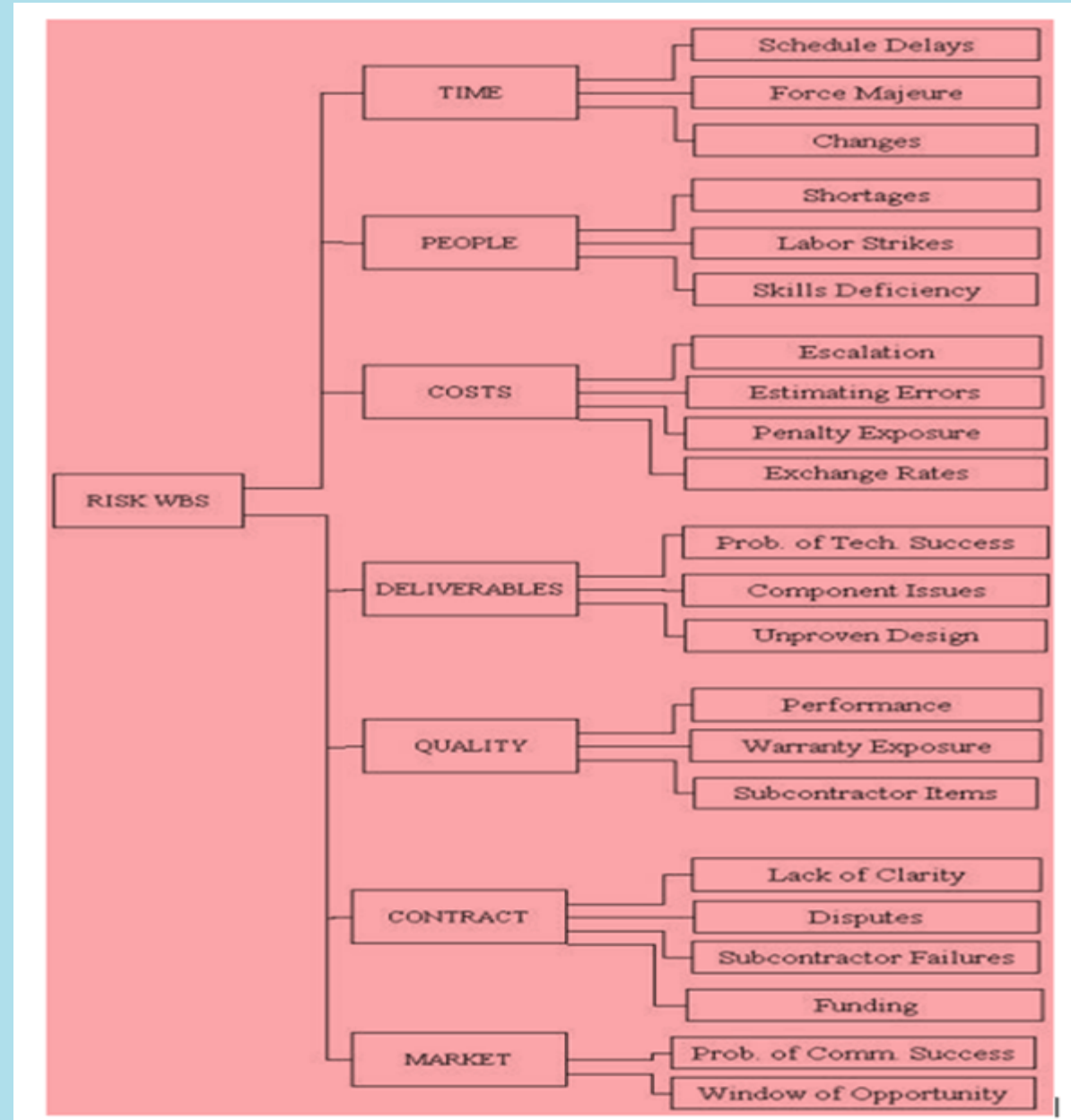
Risk Identification

- Risk identification can be done through risk workshops, checklists, brainstorming, interviewing, Delphi technique, cause and effect diagrams, lessons learnt from previous projects, root cause analysis, contacting domain experts and subject matter experts.

Risk Register is a spreadsheet which has a list of identified risks, potential responses, and root causes. It is used to monitor and track the risks (both threats and opportunities) throughout the life of the project.

- Risk response strategies can be used to manage positive and negative risks.
- Risk breakdown structure plays an important role in risk planning.
- The Risk Breakdown structure would help in identifying the risk prone areas and helps in effective evaluation and risk monitoring over the course of the project.
- It helps in providing sufficient time and resources for risk management activities. It also helps in categorizing many sources from which the project risks may arise.

- Risk Breakdown structure sample

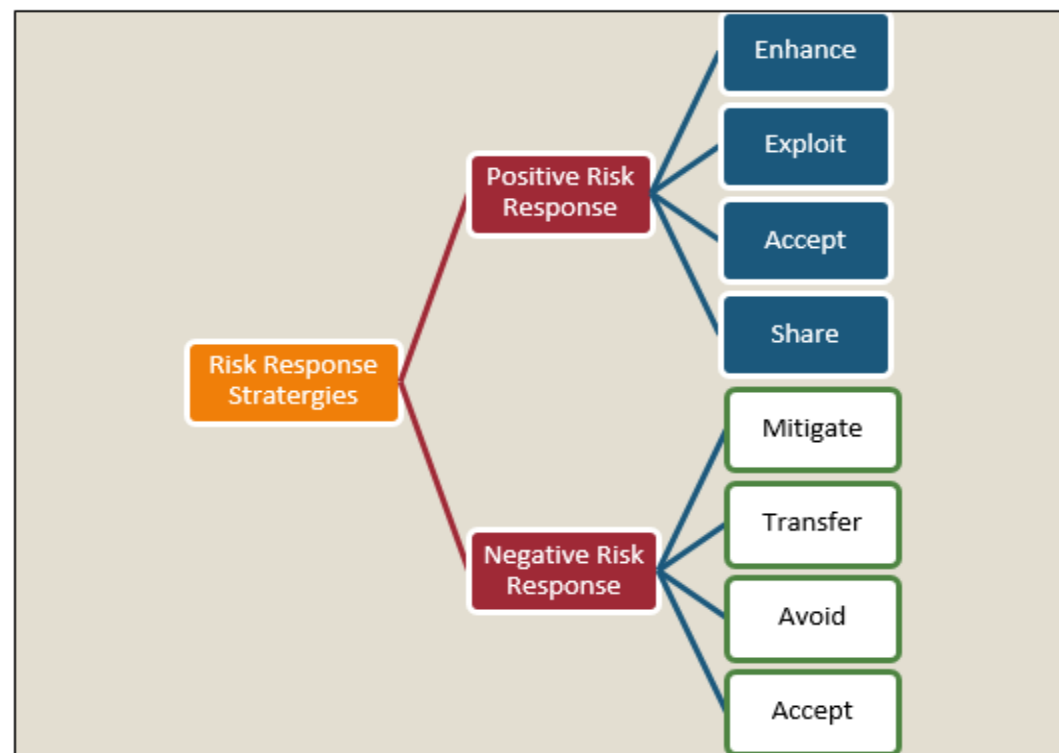


Risk Analysis (Includes Quantitative and Qualitative Analysis)

- Once the list of potential risks has been identified, the next step is to analyze them and to filter the risk based on the significance.
- One of the qualitative risk analysis technique is using Risk Matrix (covered in the next section).
- This technique is used to determine the probability and impact of the risk.

Risk Response planning

- Based on the analysis, we can decide if the risks require a response.
- For example, some risks will require a response in the project plan while some require a response in the project monitoring, and some will not require any response at all.
- The risk owner is responsible for identifying options to reduce the probability and impact of the assigned risks.
- **Risk mitigation** is a risk response method used to lessen the adverse impacts of possible threats.
- This can be done by eliminating the risks or reducing them to an acceptable level.



Risk Contingency

- Contingency can be described as a possibility of an uncertain event, but the impact is unknown or unpredictable.
- A contingency plan is also known as the action plan/back up plans for the worst case scenarios. In other words, it determines what steps could be taken when an unpredictable event materializes.

Risk Monitoring and Control

- Risk control and monitor process are used to track the identified risks, monitor residual risks, identify new risks, update the risk register, analyze the reasons for the change, execute risk response plan and monitor risk triggers, etc. Evaluate their effectiveness in reducing risks.
- This can be achieved by risk reassessments, risk audits, variance and trend analysis, technical performance measurement, status update meetings and retrospective meetings.

Risk Based Testing Approach

1. Analyze the requirements.
2. Documents (SRS, FRS, Usecases) are reviewed. This activity is done to find and eliminate errors & ambiguities.
3. Requirements sign-off's is one of the risk-reduction technique for avoiding the introduction of late changes into the projects. Any changes to requirements after the document are baselined would involve a change control process and subsequent approvals.
4. Assess the risks by calculating the likelihood and impact each requirement could have on the project taking the defined criteria's like cost, schedule, resources, scope, technical performance safety, reliability, complexity, etc. into consideration.
5. Identify the probability of failure and high-risk areas. This can be done using risk assessment matrix.
6. Use a risk register to list the set of identified risks. Update, monitor and track the risks periodically at regular intervals.
7. Risk profiling needs to be done at this stage to understand the risk capacity and risk tolerance levels.
8. Prioritize the requirements based on the rating.
9. Risk-based test process is defined
10. Highly critical and medium risks can be considered for mitigation planning, implementation, progress monitoring. Low risks can be considered on a watch list.
11. Risk data quality assessment is done to analyze the quality of the data.
12. Plan and define test according to the rating
13. Apply appropriate testing approach and test design techniques to design the test cases in a way that the highest risks items are tested first. High-risk items can be tested by the resource with good domain knowledge experience.
14. Different test design techniques can be used for e.g. using the decision table technique on high-risk test items and using 'only' equivalence partitioning for low-risk test items.
15. Test cases are also designed to cover multiple functionalities and end to end business scenarios.
16. Prepare test data and test conditions and test bed.
17. Review the Test plans, Test Strategy, Test cases, Test reports or any other document created by the testing team.

18. Peer review is an important step in defect identification and risk reduction.
19. Perform dry runs and quality checks on the results
20. Test cases are executed according to the priority of the risk item.
21. Maintain traceability between risk items, tests that cover them, results of those tests, and defects found during testing. All testing strategies executed properly will reduce quality risks.
22. Risk-based testing can be used at every level of testing, e.g. component, integration, system, and acceptance testing
23. At the system level, we need to focus on what is most important in the application. This can be determined by looking at the visibility of functions, at frequency of use and at the possible cost of failure.
24. Evaluation of exit criteria. All high-risk areas fully tested, with only minor residual risks left outstanding.
25. Risk-based Test Results reporting and metrics analysis.
26. Reassess existing risk events and new risk events based on Key Risk Indicators.
27. Risk register updation.
28. Contingency plans- This works as a fallback plan/emergency plans for the high exposure risks.
29. Defect analysis and defect prevention to eliminate the defects.
30. Retesting and Regression testing to validate the defect fixes based on pre-calculated risk analysis and high-risk areas should be most intensively covered.
31. Risk-based automation testing(if feasible)
32. Residual Risk calculation
33. Risk Monitoring and Control
34. Exit Criteria or completion criteria can be used for different risk levels. All key risks have been addressed with appropriate actions or contingency plans. Risk exposure is at or below the level agreed to as acceptable for the project.
35. Risk profiling reassessment and customer feedback.

Risk Based Testing Approach to the System Test

- **Technical System Test** –This is referred to as environment test and integration test. Environment test includes testing in development, testing, and the production environment.
- **Functional System Test**– Testing of all functionalities, features, programs, modules. The purpose of this test is to evaluate if the system meets its specified requirements.
- **Non-functional System Test**-Testing the non-functional requirements performance, load tests, stress-test, configuration tests, Security tests, backup and recovery procedures and documentation (system, operation and installation documentation).

Risk Based Testing: Complete Process

- This section covers, Risk based Test Process
- Risk Identification
- Risk Analysis
- Risk Response
- Test Scoping
- Test Process definition

