1. **Designing**

You has responsibility to create a website for Person who finding a new job and website must fill their informations of Educations, Skills. Companies that use this website must fill types of jobs that they are finding candidates. Each job has title, requirement, benefit, salary and address of working. Please design Entity Relationship Diagram of this project in details.

A diagram of a computer system

Description automatically generated with medium confidence

1. **Project**

process In software development project, how many branches in GIT does the Technical manager should use? What are the branches used for? If you have a new module to develop, which steps do you should prepare to develop from starting the module to releasing the module?

In a typical software development project, the Technical Manager should aim to use a few key Git branches:

Main Branch: This is the primary, production-ready branch that contains the latest stable version of the codebase. Changes are typically merged into the main branch after testing and review and pull request. This is for product release

Development Branch: used as the integration point for ongoing development work.after developmer merge all pull request, using this for tester review all bug before release.

Feature Branches: For each new feature or functionality, developers should create a dedicated feature branch. this allows to work on the feature independently, without affecting the main codebase.

Hotfix Branches: These are used to quickly address and resolve critical bugs or issues in release product of main branch. Hotfixes are then merged back into the main branch.

If I have new module, there are step I should prepare before release to main branch:

1. Plan how the module work
2. Create a Branch for that module

3.create the Module

4. Test the Module

5. pull request for merge the Module to development branch.

6. resolve conflix in development branch

7.testing development branch

8.merge to main branch

9.release

1. **Testing**

What are types of testing strategies? What is Unit Testing used for? What is Integration Testing used for? If a tester has a new module to test, which steps does he/she should test from starting the module to releasing the module.

Types of Test Strategy in Software Testing

In the intricate world of software development, the adoption of a suitable testing strategy is pivotal for ensuring the quality and reliability of the final product. Each testing strategy offers a unique approach and caters to different aspects of the software development lifecycle. Let’s delve into some of the prominent types of Test Strategy in Software Testing that have significantly shaped modern software testing practices.

1. Risk-Based Testing: Software Test Strategy

This is a Software Test Strategy that prioritizes testing activities based on the risk assessment of various software components. In this approach, components that are deemed high risk – due to factors like complexity, business importance, or past defects – are tested more rigorously and earlier in the testing cycle. This strategy is effective in optimizing resources and time, ensuring that the most critical parts of the application are robust and reliable.

2. Consultative Test Strategy and Test Plan

This Consultative Test Strategy and Test Plan involves a collaborative approach where stakeholders, including business analysts, developers, and users, are actively involved in the testing process. This strategy leverages the diverse perspectives and expertise of all stakeholders to identify key areas of focus, potential risks, and the most appropriate testing methodologies. It fosters a shared understanding of the project goals and ensures that the testing aligns with business requirements and user expectations.

3. Exploratory Testing

This is an approach where testers dynamically explore and test the software without predefined test cases or scripts. This strategy relies heavily on the tester’s experience, intuition, and creativity. It is particularly effective in uncovering issues that may not be easily identified through structured testing approaches. Exploratory testing is often used in conjunction with other testing strategies to enhance test coverage and discover unexpected or hidden issues.

4. V-Model Testing Strategy

This is a development methodology that emphasizes the parallel relationship between development stages and their corresponding testing phases. Each development phase, such as requirements specification or design, has a corresponding testing phase, like acceptance testing or system testing. This model ensures that testing is integrated throughout the development process, facilitating early detection and resolution of defects.

5. Behavior-Driven Development (BDD) Testing

This Test Strategy in Software Testing focuses on the behavior of the software from the user’s perspective. In BDD, testing begins with the definition of expected behavior and user stories. Tests are then designed to validate that the software behaves as expected in various scenarios. This approach bridges the gap between technical and non-technical stakeholders, ensuring that the software meets the intended user requirements.

6. Test-Driven Development (TDD)

This is an approach where test cases are written before the actual code is developed. Developers first write a failing test case that defines a desired improvement or new function, then produce the minimum amount of code to pass the test, and finally refactor the new code to acceptable standards. TDD promotes simple designs and inspires confidence in the software’s functionality.

7. Model-Based Testing Strategy

Model Based Test Strategy

involves creating models that represent the desired behavior of the system under test. These models are used to generate test cases automatically. This approach is particularly useful for complex systems, as it helps in visualizing the requirements and generating comprehensive test scenarios.

Each of these Test Strategies in Software Testing offers distinct advantages and can be chosen based on the specific requirements, complexity, and context of the software project. The key is to understand the nuances of each strategy and apply them judiciously to enhance the quality, efficiency, and effectiveness of the software testing process.

Unit Testing: Unit testing is a type of Software Testing focus on individual units or components of a software system. The purpose of unit testing is to confirm that each software unit works as intended and meets the requirements. Developers often perform unit testing, and this is done early in the development process before the code is integrated and tested as a whole system.

Integration Testing: Integration Testing is the process of testing the interface between two software units or modules. It focuses on determining the accuracy of the interface. The purpose of integration testing is to detect errors in the interaction between integration units. Once all modules have been unit tested, integration testing will be performed.

steps a tester should follow when testing a new module:

Review the Module Specifications: Thoroughly understand the requirements and functionality of the new module.

Create a Test Plan: Develop a comprehensive test plan that covers unit tests, integration tests, and any other necessary testing strategies.

Perform Unit Testing: Test the individual components or units of the new module to ensure they function correctly in isolation.

Perform Integration Testing: Test how the new module integrates and interacts with other components or modules in the application.

Execute System Testing: Test the entire, fully integrated system to ensure the new module works as expected and does not introduce any regressions.

Conduct Acceptance Testing: Test the new module from the end-user's perspective to ensure it meets their needs and expectations.

Perform Regression Testing: Run a suite of tests to ensure that the new module has not broken any existing functionality.

Document the Test Results: Thoroughly document the test results, including any issues or defects discovered, and share them with the development team.