**Mini-Project 24: Software Test**

**By Esmond Burke**

1. Read the application details again. What concerns might you have right off the

bat?

* External Threats: The first primary concern is that the app, which is available to any authenticated user, tracks drivers' location while carrying hazardous materials like explosives and radioactive materials. If a bad actor can get a user's authentication, they now have access to dangerous materials.
* Access Control: Any authenticated user having unlimited access to the app is concerning because in principle this access goes against the least privilege principle that creates a secure environment for users and the company.
* Data Protection and Privacy: Though the IT operations are in India, the app tracks activity in the United States. Hence, each state in the United States has privacy laws and regulations that the app must adhere to.
* Insider Threat: Someone who is a company employee or contractor may utilize their access to sell private data to outside sources.

1. What is the difference between test cases and test cycles (aka “test runs”)?

A test cycle is a focused set of test cases that are grouped to achieve specific testing goals (Zephyr Scale Cloud, 2020). Test cycles can be broken down in a variety of ways including specific features, specific functionality, or even specific team members (Fine, 2020). Test cases are defined routines that are executed repeatedly to ensure the product is working (Fine, 2020). In our scenario, adding the features of location, speed, direction, and cargo will undergo a test cycle for each feature. When completing the location feature test cycle, we would have different test cases to ensure that the app location features are working properly. For example, a test case can check the validity of latitude and longitude.

1. What things should be considered when writing test cases?

When writing test cases, it is important to consider the following:

* The specific features and functionality of the software that need to be tested:
* The different types of data that can be entered into the software
* The different ways that users can interact with the software. In the case of the app, use interact with a user interface with
* The different types of errors that can occur in the software

We can outline a few best practices when creating a test case (PractiTest, n.d.):

* Consider risks and priorities: As an example, a high-risk feature that is scheduled for delivery in 6 weeks might be of higher priority than a test for a low-risk feature due to be released in 1 week
* Use the 80/20 rule as a guide: 20% of tests will cover 80% of the application
* Make other people complete test cases: Test cases can be outsourced.
* The “good enough” test case: test cases are not written in one iteration so getting the case to a point of being good enough should be the main objective
* Create reusable test cases: Create tests that will be relevant in future sprints/builds/releases.
* Create a shortlist of tests before writing the details: Create a list of topics and their priority based on risk. This will help keep focus on what you need or want to test
* Classify test cases based on Business Scenarios and functionality: This process clarifies what tests can be run based on the needs of the future test plan.
* Test length: 60 to 90 minutes should be ideal for running a test.
* Test-drive the tests: Be sure to run the test for correctness.
* Keep test relevant: Continuously running tests will check them for relevancy and allow for updates.

1. What kind of documentation would you expect to come out of this project (e.g.

training docs, change requests)?

Here are some documentation that will be included in this project:

* Training documentation: Providing user manuals, guides, or training materials for end-users and administrators.
* Project plan: Outlining the objectives, scope, deliverables, and timeline of the project
* Change requests: Document any changes or enhancements requested during the project, including the rationale, impact analysis, and approval process.
* Requirements documentation: Describing the functional and non-functional requirements of the application.
* Test plan: Detailing the overall testing strategy, including test objectives, test environments, test approach, and test deliverables.
* Test cases: Documenting individual test cases, including their purpose, steps, expected results, and test data.
* Test Cycles: Documenting the different features, including cycle length, steps, and priority

1. If an online/cloud-based project management solution is being used, what

concerns might you have regarding that?

* Data security: When employing a cloud-based project management solution, we have to ensure that appropriate security measures are employed such as encryption, access controls, and regular security updates, to protect sensitive project information.
* Data privacy: Verifying that the project management solution complies with relevant data protection regulations and does not expose sensitive data to unauthorized parties.
* Vendor reliability: Evaluating the reputation, track record, and reliability of the online/cloud-based project management solution provider to ensure the long-term availability and support of the solution.
* Compliance: The cloud provider may not be compliant with all of the same laws and regulations as the company. Additionally, since operations are in 150 locations globally, emphasis needs to be placed on compliance for the respective global locations.
* Disaster recovery: If the cloud provider experiences a data breach or other disaster, the company could lose access to its data.
* Integration and compatibility: Ensuring that the project management solution integrates smoothly with other systems or tools used in the organization's IT infrastructure.

1. You find out that a couple of developers have full access to Dev, QA, and Prod. Is

this a security problem? Why or why not? Is it a compliance problem? Why or why

not?

Granting full access to developers in all environments (Dev, QA, and Prod) poses both security and compliance problems:

* Security problem: Full access to all environments increases the risk of unauthorized changes, data breaches, or malicious activities. Potentially, a developer can approve QA checks bypassing standard operating procedures. Developers should have limited access rights based on the principle of least privilege, ensuring they can only perform their necessary tasks.
* Compliance problem: Compliance standards, such as those related to data protection (e.g., GDPR) or industry-specific regulations, often require segregation of duties and proper access controls. Granting full access to developers violates these principles and can lead to compliance issues.

1. What is regression testing and why is it important?

Regression testing is a type of test performed on software that has recently undergone a code or program change to ensure the software and its features are unaffected by the coding changes (Indeed Editorial Team, 2023). Regression testing allows developers to make continuous improvements to applications without disturbing the end user's experience by validating the functionality of new code ensuring its compatibility with existing features (Guide, 2022).

1. What are the pros and cons of regression testing?

Pros of regression testing:

* Detecting defects: Regression testing helps identify any unintended consequences or defects introduced by changes made to the application, ensuring that the system remains functional and reliable.
* Ensuring stability: By retesting existing functionalities, regression testing ensures that the software continues to function as expected and does not regress due to changes made in the application.

Cons of regression testing:

* Time-consuming: Depending on the size and complexity of the application, regression testing can be time-consuming, especially when executed manually.
* Resource-intensive: Regression testing requires adequate resources, including personnel, test environments, and test data, which can pose challenges in terms of cost and availability.

1. What are some “political” concerns that could arise if issues are found that could delay the project?

* Stakeholder expectations: Project delays can lead to dissatisfaction among stakeholders, such as management, users, or clients, who may have already set expectations based on the planned go-live date.
* Budgetary constraints: Delays in the project schedule may result in increased costs, such as additional development or testing efforts, which can impact the project's budget.
* Interdepartmental conflicts: If issues are found during testing that require additional development or fixes, conflicts may arise between different teams or departments involved in the project, such as development, testing, and management.

1. What are the two most common project management methodologies? What are

the main differences between them?

* Waterfall: In the waterfall methodology, projects progress linearly through distinct phases, including requirements gathering, design, development, testing, deployment, and maintenance. Each phase is typically completed before moving on to the next, with minimal opportunity for iteration or change (Hoory & Bottorff, 2022).
* Agile: Agile methodologies, such as Scrum or Kanban, involve iterative and incremental development. Projects are divided into small, manageable work units called sprints, and requirements can evolve throughout the project. Continuous collaboration, flexibility, and adaptation are key principles of Agile (Guide, n.d.).

Waterfall is fixed and more rigid, the Agile method is adaptable and is better suited for a client-facing work process (Guide, n.d.).

1. In the middle of QA testing, management requested that another feature be included in the application. What is this called, and how should the project team evaluate it?

When management requests another feature in the application during QA testing, it is called a change request. The project team should evaluate the change request by considering factors such as:

* Impact analysis: Assessing the potential impact of the change on the project timeline, budget, and existing functionalities.
* Priority and urgency: Evaluating the importance and urgency of the requested feature compared to other project priorities.
* Resource availability: Determining if the project team has the necessary resources, including time and personnel, to accommodate the change request without compromising the overall quality of the application.
* Risk assessment: Identifying any potential risks associated with implementing the requested feature, such as introducing new defects or destabilizing the system.

Based on the evaluation, the project team can determine whether to include the new feature in the current release or plan it for a future iteration or release.

1. When the application went live, some bugs were found. At what stage of the process should the bugs have been caught and fixed? Why?

The bugs should ideally have been caught and fixed during the testing phase, specifically in the Quality Assurance (QA) environment. QA testing is conducted to identify and resolve defects before the application is deployed to the Production (Prod) environment.

Catching bugs in the QA environment allows for proper debugging, fixing, and retesting before the application goes live. This helps minimize the risk of releasing a defective or unstable application to end-users, ensuring a higher level of quality and reliability.

Test Plan per Rubric:

Test Plan

* Purpose: This test plan describes the testing activities that will be performed to ensure the quality of the new application that will track the movement of trucks that carry hazardous materials.
* Scope: The test plan covers all aspects of the application, including the user interface, the database, and the back-end logic.
* Schedule: The testing will be conducted in three phases:
  + Unit testing: This will be performed by the developers to ensure that the individual units of code are working as expected.
  + Integration testing: This will be performed by the QA team to ensure that the different units of code are working together correctly.
  + System testing: This will be performed by a team of users to ensure that the application meets the requirements and that it is easy to use.
* Resources: The following resources will be needed for the testing:
  + Developers: The developers will be responsible for unit testing the code.
  + QA team: The QA team will be responsible for integration testing and system testing.
  + Users: A team of users will be responsible for system testing.
* Deliverables: The following deliverables will be produced as a result of the testing:
  + Test reports: These reports will document the results of the testing.
  + Bug reports: These reports will document any defects that are found during the testing.
* Approvals: The following approvals are required for the testing:
  + The development manager must approve the test plan.
  + The QA manager must approve the test reports and bug reports.
* Risks: The following risks have been identified for the testing:
  + The application may not meet the requirements.
  + The application may not be easy to use.
  + Defects may be found after the application has gone live.
* Mitigation strategies: The following mitigation strategies have been developed to address the risks:
  + The requirements will be reviewed by a team of users to ensure that they are complete and accurate.
  + The application will be user-tested to ensure that it is easy to use.
  + A process will be in place to track and fix defects that are found during the testing.

Test Cases:

Test Case 1:

* Description: Test the ability to track the movement of trucks.
* Steps:
  + Log in to the application as a user.
  + Enter the details of a truck, including its license plate number, start location, destination, and cargo type.
  + Click on the "Track" button.
  + Verify that the application displays the current location of the truck.
  + Verify that the application updates the location of the truck in real-time.

Test Case 2:

* Description: Test the ability to filter the results of a search.
* Steps:
  + Log in to the application as a user.
  + Click on the "Search" button.
  + Enter a search term in the search box.
  + Click on the "Filter" button.
  + Select the criteria that you want to filter the results by.
  + Click on the "Search" button again.
  + Verify that the application only displays the results that match the criteria that you selected.

Test Case 3:

* Description: Test the ability to view the details of a truck.
* Steps:
  + Log in to the application as a user.
  + Click on the "Trucks" tab.
  + Click on the name of a truck.
  + Verify that the application displays the details of the truck, including its license plate number, start location, destination, cargo type, and current location.

Test Cycle:

Test Cycle 1:

* Goal: To verify the basic functionality of the application.
* Activities:
  + Unit testing by developers.
  + Integration testing by the QA team.
  + System testing by a team of users.
* Expected Results: All basic functionality of the application should work as expected.

Test Cycle 2:

* Goal: To verify the performance of the application.
* Activities:
  + Load testing by a third-party company.
  + Stress testing by the QA team.
* Expected Results: The application should be able to handle the expected load and stress without any performance issues.

Test Cycle 3:

* Goal: To verify the security of the application.
* Activities:
  + Security testing by a third-party company.
  + Penetration testing by the QA team.
* Expected Results: The application should be secure and should not be vulnerable to common security attacks.

**References**

Fine, M. (2020, March 23). *Software Testing Foundations: Test Management*. LinkedIn.com. https://www.linkedin.com/learning/software-testing-foundations-test-management/test-plan-test-cases-test-cycles?u=36492188

Guide, S. (n.d.). *What is Agile?* Atlassian. Retrieved June 14, 2023, from https://www.atlassian.com/agile

Guide, S. (2022, February 8). *Why is Regression Testing Important?* Copado. Retrieved June 14, 2023, from https://www.copado.com/devops-hub/blog/why-is-regression-testing-important-crt

Hoory, L., & Bottorff, C. (2022, March 25). *What is Waterfall Methodology?* Forbes. Retrieved June 14, 2023, from https://www.forbes.com/advisor/business/what-is-waterfall-methodology/

Indeed Editorial Team. (2023, February 3). *What Is Regression Testing? (With Benefits and Example)*. Indeed. Retrieved June 14, 2023, from https://www.indeed.com/career-advice/career-development/regression-testing

PractiTest. (n.d.). *How To Write Test Cases? 10 Important Guidelines*. PractiTest. Retrieved June 14, 2023, from https://www.practitest.com/qa-learningcenter/best-practices/write-a-test-case/

Zephyr Scale Cloud. (2020, October 29). *Test Cycles (Overview) | Zephyr Scale Cloud Documentation*. SmartBear Support. Retrieved June 14, 2023, from https://support.smartbear.com/zephyr-scale-cloud/docs/test-cycles/test-cycles-overview.html