

Part 1

Note - Our answers to the multiple choice questions are marked in **bold**

1. According to the ISTQB Glossary, regression testing is required for what purpose?

- a) To verify the success of corrective actions.
- b) To motivate better unit testing by the programmers.
- c) To prevent a task from being incorrectly considered completed.
- d) To ensure that defects have not been introduced by a modification.**

2. Pair the following processes with their corresponding activities:

- | | |
|------------------------|--|
| 1. Test analysis | A. Group tests into scripts |
| 2. Test plan | B. Write or extract a test summary report for the stakeholders |
| 3. Test implementation | C. Transform the test objectives into test conditions and test cases |
| 4. Test reporting | D. Establish the scope, objectives and risks of testing |

Order of corresponding activities is: 1 - D, 2 - C, 3 - A , 4 - B

3. Which of the following is a non-functional quality characteristic?

- a) Accuracy
- b) Reliability**
- c) Security
- d) Regression

4. Which of these is a functional test?

- a) Measuring how long it takes to upload a file to a cloud-based solution.
- b) Checking the effect of high volumes of traffic in a call-center system.
- c) Counting if the number of outputs is as expected.**
- d) Checking how easy is to use the software system.

5. Acceptance testing is not the responsibility of the development team. It is the responsibility of the customers, but the development team can assist in the process.

- a) True**
- b) False

6. Static code analysis typically identifies all but one of the following problems. Which is it?

- a) Unreachable code
- b) Memory leaks**
- c) Violations of programming standards
- d) Variables that are declared, but never used

7. In the follow-up phase of a review activity, the moderator checks if the defects reported by the reviewers have been indeed fixed by the document author.

8. Should preconditions and postconditions be part of a test case?

- a) Yes
- b) No

9. Equivalence partitioning is applied as test design technique when the inputs and the outputs of a software under test can be grouped in a way that exhibits similar behavior.

10. Pair the following roles with their typical activities:

Tester

- Writes automated tests
- Evaluates the results of the execution of tests: pass or fail
- Acquires and prepares test data

Test leader

- Writes test summary reports for management
- Introduces metrics for measuring the test progress
- Gives recommendations to continue or stop the testing, based on the test execution results.

11. Which of the following metrics would be most useful to monitor during test execution?

- a) Number of testers versus number of developers in the team.
- b) Number of defects found and fixed.**
- c) Number of test environments that should have been set up, but were abandoned.
- d) Percentage of requirements for which a test has been written.

12. Is it allowed to use low-fidelity prototyping when designing a user-centric software system?

- a) Yes
- b) No

13. What does it mean that a user interface is operable?

- I. All functionality can be operated by the same user.**
- II. All functionality is available from a keyboard.**
- III. All user interface is navigable with a keyboard.**

- a) I, II
- b) I, III
- c) II, III**
- d) I, II, III

Part 2

What are the five fundamental test activities? Briefly describe their respective tasks from planning to closure.

1. Planning and control: The test leader makes sure that the developers understand the goals and objectives of the project, including the stakeholders. It's based on this understanding that we set our goals and the objectives of the testing itself. The most important being to define an exit criteria, such that we can assure that our tests have satisfied the objective.
2. Analysis and design: The general test objectives which were specified in the planning and control phase is transformed by the developers into actual test designs including the setup for the test environment.
3. Implementation and execution: The developers derive test conditions from the test designs and write test cases and procedures. This also includes setting up the environment in which the tests will be executed in.
4. Evaluating exit criteria and reporting: The result of each executed test in every test level is assessed against the objectives by developers. If the results satisfies our objectives and the exit criteria, the test leader writes a test summary report which is then given to the management.
5. Test closure activities: The test leader gathers data from completed test activities, but also evaluate how the testing went and lesson to be learned for future releases and projects.

2. Explain briefly the following terms: regression testing and confirmation testing

- Regression testing: When modifying any part of the system, the developer has to make sure that those modifications didn't cause any unintended changes. This is done by executing tests after the modifications. If all tests successfully finishes, the developer can be assured that the modification didn't cause unintended changes.
- Confirmation testing: After identifying and fixing a defect in the system, the developer execute the same corresponding tests cases to confirm that the defect has been fixed.

3. Identify and describe three types of non-functional software characteristics.

- Accessibility: How easily can people with and without disabilities access and benefit from the given system.
- Availability: To which degree (days/hours) is the system in a functioning state.

- Reliability: How well can the system function during a specific moment or time interval.

4. How does testing depend on the development life-cycle for the software under test? (sequential and iterative-incremental)

- Sequential: For this given model of development, all tests are executed at the final stages of the life-cycle. If a defect is found in one of the given development stages, to correct that given defect the current development stage of the project has to return to the phase in which the defect occurred. Because testing is done so late in the life-cycle, the chance that the defect has nested itself into several phases in the project is higher, such that the actual testing phase happens at a later time and costs increases.
- Iterative-incremental: At each given stage of the life-cycle tests will be executed concurrently. This reduces the chance of a defect occurring at any given phase, because each associated phase has it's own test phase such that the costs decreases.

5. Define and explain the purpose of entry criteria and exit criteria in software testing.

- Entry criteria: The criteria that have to be meet before the testing can be performed, eg: Platform is developed.
- Exit criteria: The criteria that have to be meet before the testing can finish, eg: 100% decision coverage

6. Summarize the potential benefits and potential risks a company may face when using test tools.

Benefits:

- Higher chance of finding new defects
- Possibility of catching different types of defects
- Higher chance to find defects which are hard to find
- Automate repetitive work
- One standard of testing which increases the consistency and lower cognitive load for the developers
- Streamlined information and progress on execution results

Disadvantages:

- The personnel is untrained, there's no experience with such tools in the organization
- The tools most likely isn't compatible with different platforms/tools
- Trust that the tool actually finds defects

7. What is a decision table? Provide an example.

A decision table is a specification-based technique, the primary goal is to test business rules. The way of doing so is by combining a set of business rules with a random a set of conditions and asserting that the combination conditions yields the correct outcome.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Personal id has been entered	T	F	T	T	T	F	F	F
Amount of credit to borrow has been entered	T	T	F	T	F	T	F	F
Amount is below allowed credit balance	T	T	T	F	F	F	T	F
Outcome								
Grant credit	T	F	F	F	F	F	F	F

In the example above the conditions is our set of business rules. Each combination of conditions yields us the result of either granting credit or denying it. By using a decision table we can clearly see what combination of rules has to be satisfied in order to yield the right outcome. The benefit of using decision tables is that it aids in selecting effective test cases and catching possible ambiguities in the specification.

8. Explain the role of the personas in the study of accessibility. Provide an example of such a persona.

Personas are used to give a realistic view of the people that will be users of the system. The benefits of using personas is to have a clear and consistent view on the possible accessibility challenges, ideal users of the system will meet, but also how to solve those challenges and meet their needs.

Benjamin, 18

“All cool websites uses flashy colors, but i’m colorblind so it all looks weird to me”

- Lives at home
- Attends high school
- Uses laptop at home, school and always carries a smartphone

Characteristic: Uses color contrast adjustment to improve user experience

Aptitude: Skilled user of technology

Attitude: Digital native and early adopter of technology

Assistive technology: Color contrast adjustment and various screen dimming software.

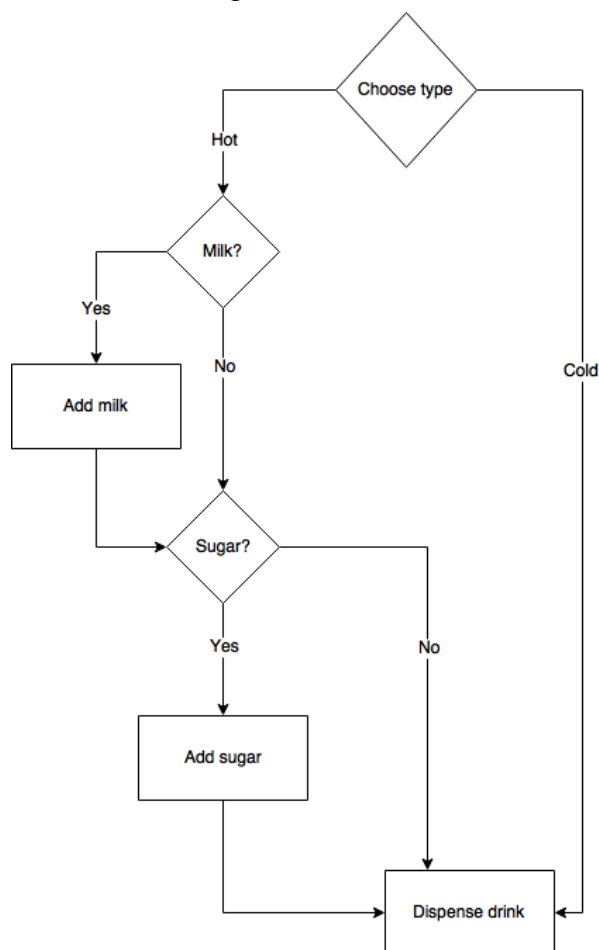
Part 3

Problem 1

Statement coverage for each test:

- Test.1: 33%
- Test.2: 66%

Control flow diagram



The test case: "Hot drink with milk and sugar" would yield a 100% statement coverage.

Problem 2

Equivalence partitions with boundary values would be the best technique to use to test for interest-rates on different accounts.

Minimum required tests for this case would be 14 tests to test for both boundary-values in each equivalence partition.

1. test spar-account with balance under 0, expected: error
2. test spar-account with balance 0, expected: 2% interest
3. test spar-account with balance between 0 and 1000, expected: 2% interest
4. test spar-account with balance 1000, expected: 2%
5. test spar-account with balance 1001, expected: 3%
6. test spar-account with balance between 1000 and 10 000, expected: 3%
7. test spar-account with balance 10 000, expected: 3%
8. test spar-account with balance 10 001, expected: 4%
9. test spar-account with balance more than 10 000, expected: 4%
10. test super-spar with balance less than 0, expected: error
11. test super-spar with balance between 0 and 5000, expected: 5% interest
12. test super-spar with balance 5000, expected: 5%
13. test super-spar with balance 5001, expected: 10% interest
14. test super-spar with balance more than 5000, expected: 10% interest.

Problem 3

The issue with the use case scenario above is that there's a contradiction between the precondition, which states: **"The user has to be logged-in to the system"** and point.4 in the main branch which states **"The users must be logged out of the website."**

Severity: High

Impact: High

Date: Non applicable

Tester: Mr.Rose

Program: Requirement document

Summary: The user is immediately logged out after adding an item to cart

Detailed description: The user is logged in and has a few items in the cart. The user can browse the website, add items to their shopping cart and the in-house stock is updated accordingly. After the user adds an item to their cart, the user is immediately logged out.

Input: Logged in user adds an item to their cart

Output: User is automatically logged out.

Expected result: The user is redirected back to the main website with an updated cart.

Part 4

Test automation:

An automated test is a program that can run tests automatically on different parts of the system that is being tested.

Test automation is used to make regression-tests and confirmation-tests more effective. In many cases automated tests can be reused in other scenarios and on other systems. Test-automation is cognitively easier for the people creating the tests.

It's important because humans have an error-bias meaning that humans make mistakes when they do repetitive tasks. It's the best way for testing only parts of code without running the whole program. It also allows you to test greater amounts of code in less amount of time.

1. Graphical user interface
2. and Application Program interfaces

GUIs can test the frontend of the system(usability and accessibility). And APIs can test the backend of the system(implementation etc.).

Benefits:

- It's more effective in testing the implementation of the system
- It's more reliable
- It's easier to test for changes
- It's easier to replicate scenarios
- It's better for testing individual units in the system
- It's better for diagnosing where the defects arise
- Less cognitive load

Disadvantages:

- It's less effective testing non-functional requirements like usability and adherence to rules and regulation.
- Not everything can be automated, and not everything should be automated.
- It's great for testing simple units, but bad for testing complex scenarios.

Approaches:

1. Capture and replay
2. Test-driven development
3. Behavior-driven development
4. Data-driven development (DDT)

Major successfactors:

The success factors are mostly linked to the implementation of the automated test.

So the factors would include:

- Efficient use of resources.
- Making the tests simple.
- Having the test perform a very specific task.
- Granularity and an streamlined overview of progression

Manual testing of an ATM

Description:

- Basic ATM: The ATM is required to have a graphical user interface to facilitate the user for their chosen operation. The pin keyboard should include the numbers from 0-9 and the option to accept, reject and cancel their current operation.

The graphical user interface has to support the functionalities of withdrawal and reporting of the account balance. The ATM should also have a constraint on the number of times the user is allowed to enter the incorrect pincode. If the user exceeds the given limit, the ATM should consume the card and place it into a reject box.

It is required that the ATM has a money dispenser to support withdrawal of money and and a receipt printer to state the response current operation. The ATM is required to have necessary memory and computing power to be able to communicate with the bank's servers in order to update and retrieve the correct information.

- Advanced ATM: Same as the basic ATM, with the exception of a few additional functions. A deposit-box and deposit functionality, and functionality to report account information.

Actors:

- A diverse set of personas and their specific accessibility issues.
- A normal user

Manual tests:

1. Testing for normal withdrawal of different amounts
2. Testing for deposit
3. Testing for account information
4. Testing for usability and accessibility

5. Testing for bank depositing money and reports of not sufficient cash in the ATM
6. Testing for multiple wrong pin-codes and eating of credit-cards.

Prioritized list from most important to least important:

5, 1, 6, 3, 2, 4

We've left out security, because it's a whole different set of tests, that are both physical and abstract in nature.

Also tests for implementation and lower levels are left out, because they're a whole different part of the problem and should have their own specific tests.

Also the communication between the bank account and the atm to check to see if the amount was withdrawn from the account, because then we would have had to test both the atm and the banksystem. The bank system is not our test-responsibility.