VALENTINE
Page No.:
Date: / /

Unit-1

Q1. Explain the testing principle.

Ans There are seven principle of testing :-

1. Testing shows presence of defects.
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. pesticide paradox
6. Testing is context dependent
7. Absence - of - error fallacy

1. Testing shows presence of defects.

→ testing can show that defects are present, but can not prove that there are no defect.

⇒ Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

VALENTINE
Page No.:
Date: / /

2. Exhaustive testing is impossible

- Exhaustive means fully comprehensive — all combinations of inputs and preconditions
- And testing everything is not feasible except for small cases.
- Instead of exhaustive testing, risk analysis and priorities should be used to focus testing efforts.

3. Early testing :-

- Testing activity should start as early as possible in the software or system development life cycle and should be focused on defined objectives.

4. Defect Clustering

- A small number of modules contain most of the defects discovered during pre-released testing or are responsible for the most operational failures.

5. Pesticide paradox

- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- To overcome this, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

6. Testing is context dependent

- testing is done differently in different contexts.

7. Absence-of-error fallacy

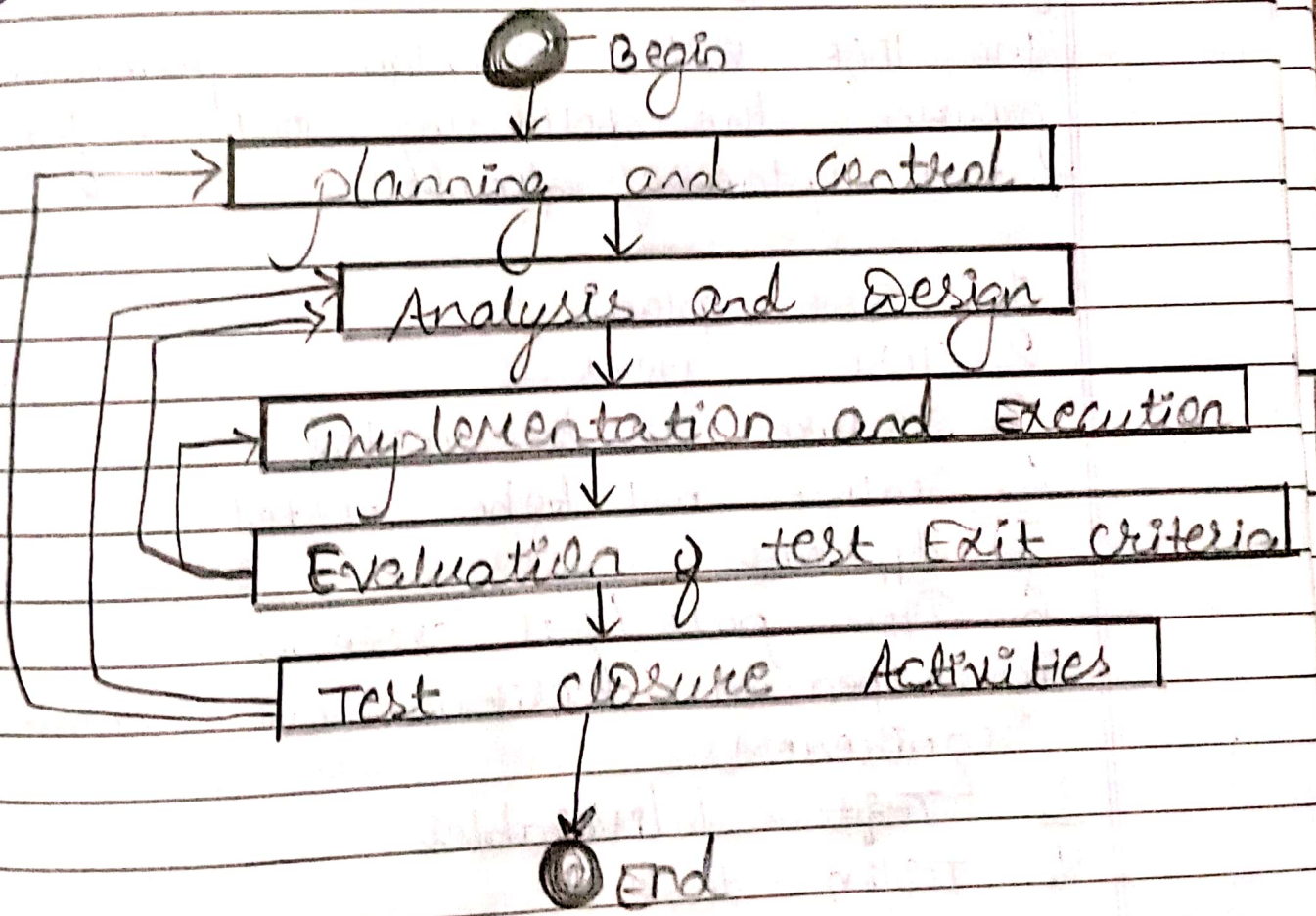
- finding and fixing defects does not help if the system built is unusable and does not fulfill the users needs and expectations.

VALENTINE
Page No.:
Date: / /

Q2 Write test process with all the steps.

Ans Test analysts and technical test analysts are a critical part of a successful test process implementation.

There are five steps to generic test process.



VALENTINE
Page No.:
Date: / /

- Test planning and test control

- at the planning stage, the test manager is determining the testing approach, planning the resource, setting the strategy, creating the test schedule, and determining the metrics that will be needed for control and monitoring of the project.

The IEEE 829 Test plan specification provides the following outline for the test plan document

1. Test plan
2. Test items
3. Features to be tested
4. Features not to be tested
5. Approach
6. Item pass/fail criteria
7. Suspension criteria and resumption requirements.
8. Test deliverables
9. Testing task
10. Environmental needs
11. Responsibilities
12. Staffing and training
13. Schedule
14. Risk & Contingencies
15. Approvals.

VALENTINE
Page No.:
Date: / /

- Test analysis and Design

At this phase, we are considering this details of testing project.

This is where we are figuring out what to test, how much effort to expand, what type of testing we should do and any tool that will be required for this effort.

This specify what we are going to test. This document consist of
1. Test design specification
2. features to be tested
3. Approach refinement
4. Test identification
5. Features pass/fail criteria

- Test implementation and Execution

Logical test cases are the high level test description that do not define the data that is to be used for test.
Concreted test case includes the actual data that is to be used during testing.

VALENTINE
Page No.:
Date: / /

The IEEE 829 test case specification is commonly used in industry that specify following information :-

1. Test case specification identifier
2. Test items
3. Input specification
4. Output specification
5. Environment needs
6. Special procedural requirements
7. Intercase dependencies.

The IEEE 829 test item transmitted report provides the information we need to locate and install the software we will be testing :-

(1) Transmitted report identifiers
(2) Transmitted items
(3) Locations
(4) Status
(5) Approvals.

As we run the test case, we want to be sure we are recording the execution information in the IEEE 829 test log.

① test log identifier
② Description
③ Activity and event entries.

- We can fill out an IEEE 829 test incident report that contains the following information:-

① Test incident report identifier
② Summary
③ The incident description
④ Impact.

- Evaluation of exit criteria and reporting

The test summary report can be prepared periodically as a type of test progress report at the conclusion of a level of testing but it is normally done at the conclusion of the testing efforts.

The test summary document includes the following information:-
① Test Summary report identifier
② Summary
③ Variances
④ comprehensiveness assessment
⑤ Summary of activities
⑥ Summary of result
⑦ Evaluation
⑧ Approvals

- Test closure Activity

- The test closure activities occur after the release has been shipped out.

- These test closure activities are often under budgeted and often receives inadequate attention, because the next project is already waiting.

Inadequate time spent on the test closure activities also reduces the effort we can spend looking for ways to improve our process.

Q3. Give any two differences between a logical test case and a Concrete test case.

- Logical test caseses are the high-level test description that do not define the data that is to be used for the tests.

- Concrete test cases include the actual data that is to be used during testing.

VALENTINE
Page No.:
Date: / /

Q4 Why the cost of the system is increased?

Ans The cost of testing which may increase considerably due to:

- High level of complexity
- The time and effort needed to localize defects.
- More integration testing may be required
- Higher management overhead
- Lack of overall control

Q5. Give any two differences between project risk and product risk.

- project risk :- they are oriented towards anything that could cause the overall project to fail to meet its objectives.

project risk includes personal issues :-
① vacations
② training
③ availability / vendor issues or third party issues.

VALENTINE
Page No. :
Date : / /

- product risk :~ These are the risk within the product itself, such as unfound defects. Testing and following good qualities practices are ways we mitigate product risk.

Unit - 2

Q1 Describe the benefits and drawback of structure-based Techniques.

- Benefits :- Their ability to supplement the tests designed black-box techniques to increase levels of coverage and increase testing effectiveness.

- Drawbacks :-
  - Availability of structural information
  - Skills levels
  - Effort required to achieve required coverage levels
  - Inability to detect faults which are sensitive to data
  - Calling order
  - Testing focuses only on structural issues.

Q2 Explain the procedure of orthogonal array with an example.

Ans These test design techniques are all oriented towards determining the representative sample. Combinations to be tested.

- Testing in a complex environment often results in test combinatorial in

the hundreds or even thousand.

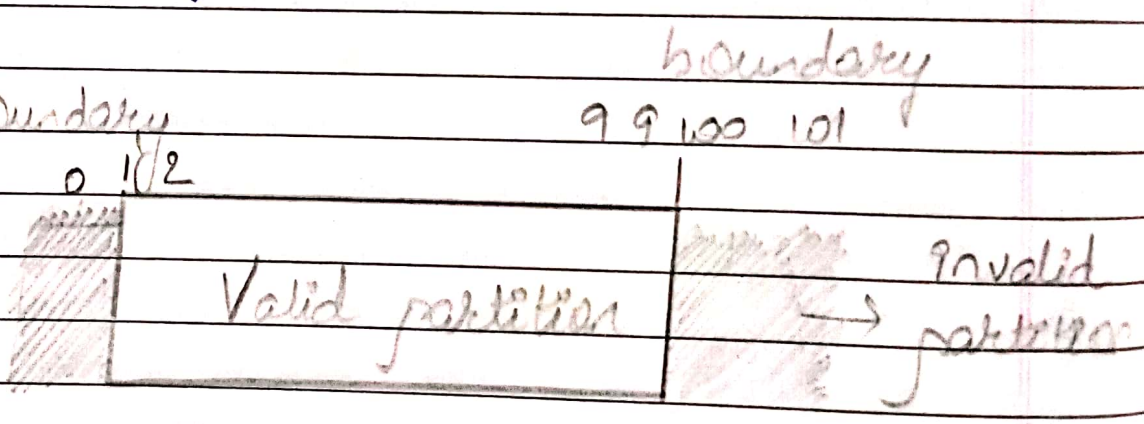
- All-pairs testing works at taking pairing of the options, eliminating the combinations that are impossible or unlikely to occur, and testing all realistic pair combinations
- Example, operating System, browser, language
- Given these parameters and their values we can build up a data tables. That has a lot of combinations.
- Orthogonal arrays are another method used to deal with large numbers of combinations of parameters. Every parameter is compared with every parameter in the neighboring column. We can be sure that problems with one particular value are detected. This is also known as a single-mode fault.
- we can also detect the interaction between two parameter, known as a double-mode fault.

- Example :-
- Application will run in windows and Mac both.
- all browser will supporting
- all language will support
- Database will support
- Unix, linux, solaris, hpux server will supporting.

Q3. Explain boundary value analysis with its strength and weakness in detail.

⇒ Boundary value analysis is a refinement of equivalence partitions. Once we have defined our partitions, we can then employ BVA to be sure we create test cases for the boundary values.

Boundary values are those values or conditions that occur on the edges of the partitions.



- BVA - Strengths
- It's very easy to forget to test the boundaries.
- Create test for those boundaries to reduce the bugs.

- BVA - weakness.
- The risk of putting too much emphasis on the edges and not enough on the rest of the factuality.
- if we had done better boundary value analysis in the experience report below, we would have caught the problem.

Q4 How does boundary value analysis improve the performance of the equivalence partitioning technique.

Ans Boundary value analysis is better than Equivalence partitioning as it consider both positive and negative values along with maximum and minimum value. So, when compared with Equivalence partitioning, Boundary value analysis proves to be a better choice in assuring the quality.

VALENTINE
Page No.:
Date: / /

- Software testing techniques allow you to design better test cases. There are five primarily used techniques.
- Boundary value analysis is testing at the boundaries between partitions.
- Equivalent class partitioning allows you to divide set of test conditions into a partition which should be considered the same.

VALENTINE
Page No.:
Date: / /

Q5 Write a short note on Defect-Based Testing technique.

Ans This is useful technique for focusing on and detecting particular types of defects.

- Based on taxonomies defect will be determine. It may be any one from following:-
  - Root Cause's
  - Defects
  - failure
- Coverage criteria for defect-based test do not imply that the set of test is completed but that is sufficient detection will be provided for the target defects.

- Taxonomies -

- There are published taxonomies or classification of bugs that we can use to help us identify possible areas for defect-based test.

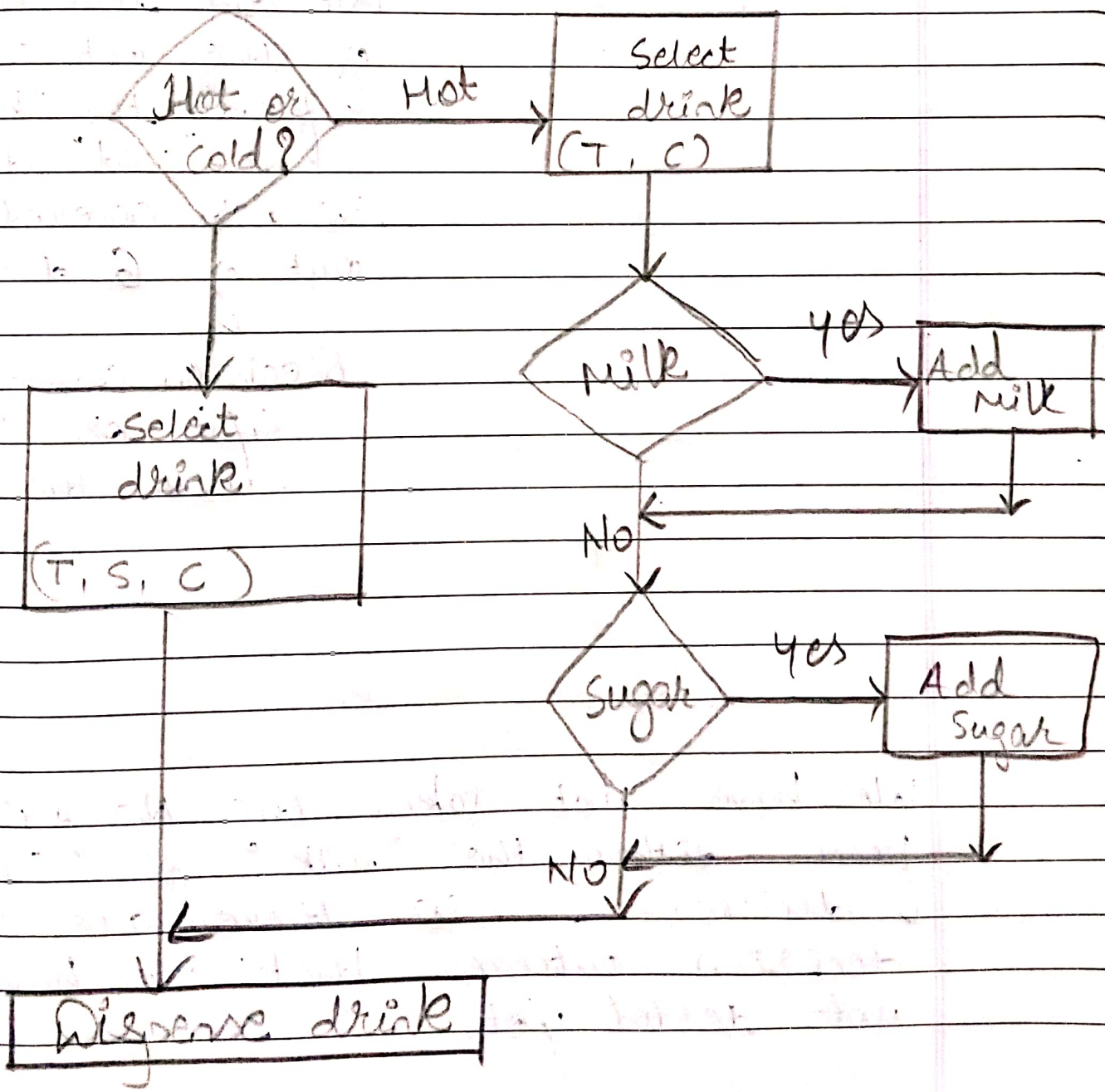
Q6. Discuss the criteria used for selecting the structure-Based techniques.

Ans. During the discussion on individual structure-based techniques. Structure-based test design techniques are a good way to help ensure more breadth of testing. To measure what percentage of code has been exercised by a test suite, one or more coverage criteria is used. A coverage criterion is usually defined as a rule or requirement, which a test suite needs to satisfy.

Page No.:
Date: / /

Q. A vending machine dispenses either hot or cold drinks. If you choose a hot drink, it asks if you want milk, then it asks if you want sugar, then your drink is dispensed.

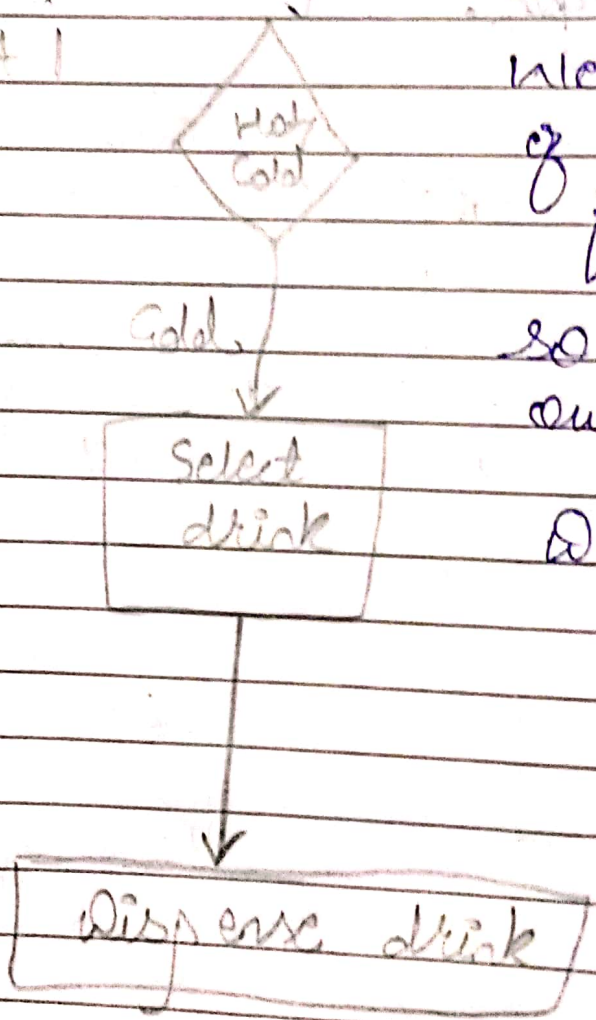
- Draw a control flow diagram for this example.



- Given the following tests, what is the decision coverage achieved?

Test 1 : Cold drink

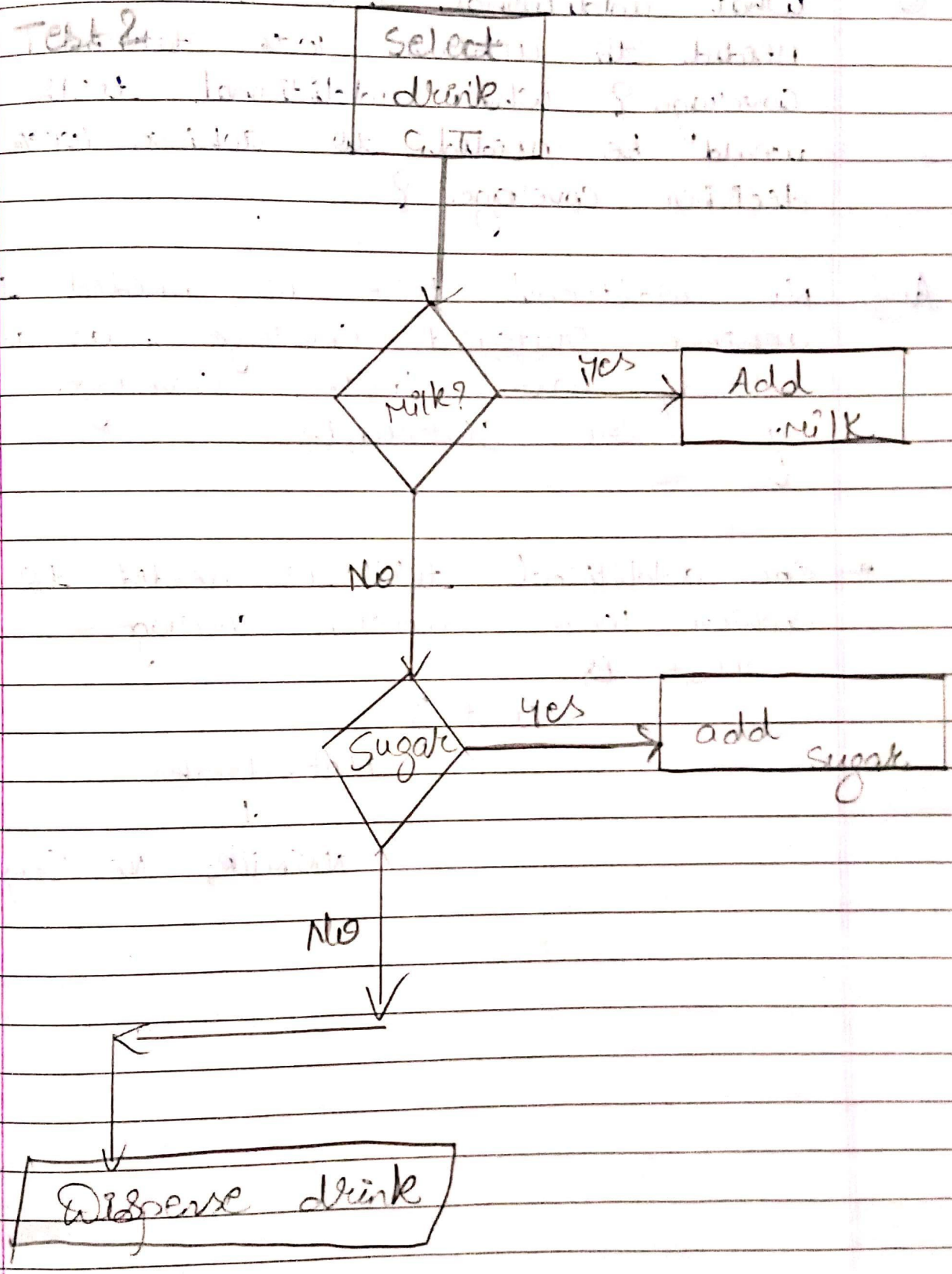
Test 2 : Hot drink with milk and sugar.



We did test both of the out comes from the "hot" or "cold" decision so we covered both out of 6 decision.

Decision coverage is 4/6 or 67% with the test

We have not taken the No exit from either the "milk" or "Sugar" decisions, so there are two decision outcomes that we have not tested yet.



Q What additional tests would be needed to achieve 100% statement coverage? What additional tests would be needed to achieve 100% decision coverage?

Ans No additional test is needed to achieve Statement Coverage, as we already have 100% coverage of the Statements.

- One additional test is needed to achieve 100% decision coverage — That is

Test 3

Hot drinks

1

No Milk, No Sugar

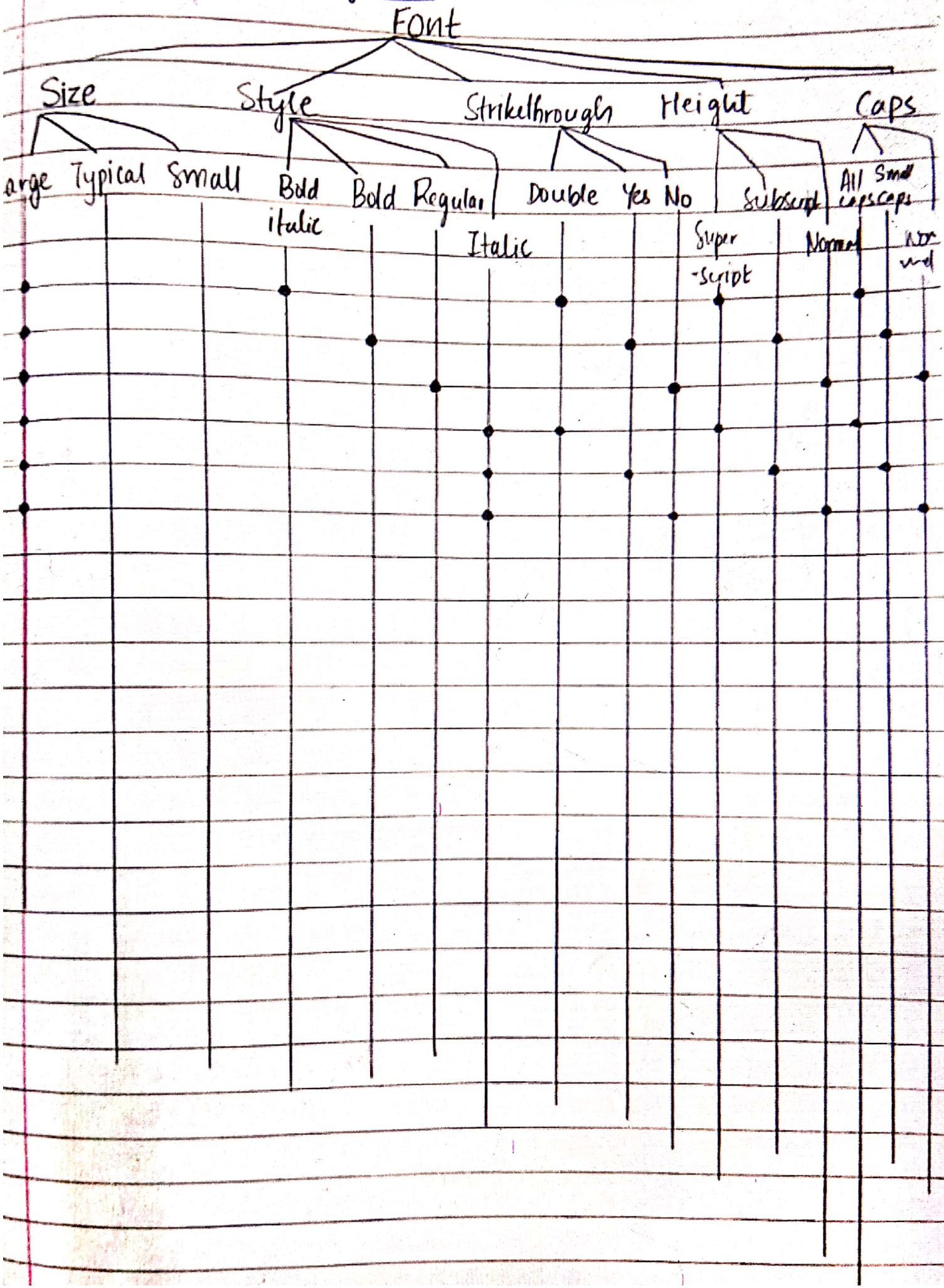
Q8. Explain the classification tree. Draw Configuration data table, Configuration diagram for configuration data table from given configuration tree.

Ans Classification tree method is a black box testing technique to test combinations of features. It is a graphical representation of the combinations of conditions to be tested. The items to be tested are created as classes and classifications within the classes.

- Configuration data table

Size	Style	Strike through	Height	Caps
Large	Bold Italic	Double	Superscript	All caps
Typical	Bold	yes	Sub script	Small caps
Small	Regular Italic	No	Normal	Normal

Configuration diagram



VALENTINE
Page No.:
Date: / /

Q9e If you take the train before 9:30 am or in the afternoon after 4:00 PM until 7:30 PM, you must pay full fare. A Saver ticket is available for trains between 9:30 am and 4:00 PM, and after 7:30 PM. What are the partitions and boundary values to test the train times for ticket types? Which are valid partitions and which is an invalid partition what are the boundary values? Derive test cases for the partition and boundaries.

and Here Boundaries values are 9:29 am, 9:30 AM, 4:00 PM, 4:01 PM, 7:30 PM, 7:31 PM. Valid partitions are when train is running and Invalid are when train is not running.

Scheduled Departure time	<= 9:29 am	9:30 AM to 4:00 PM	4:01 PM to 7:30 PM	>= 7:31 PM
Ticket types	full	Saver	full	Saver

- Test Cases :-

TC ID	Input	Expected Outcome
1	Depart 4:30 am	Full fare
2	Depart 9:29 am	Full fare
3	Depart 9:30 am	Saver
4	Depart 1:00 pm	Saver
5	Depart 4:00 pm	Saver
6	Depart 4:01 pm	Full fare
7	Depart 6:00 pm	Full
8	Depart 7:30 pm	Full
9	Depart 7:31 pm	Saver
10	Depart 11:00 pm	Saver

Test cases 1, 4, 7 and 10 are based on equivalence partition values, test cases 2, 3, 5, 6, 8 and 9 are based on boundary values.

Unit 3

Q1e Explain exploratory testing with example.

Ans • Exploratory testing examine the accuracy and it is not ad hoc testing.

- It occurs when the tester plans, designs and executes tests concurrently and learns about the product.
- It planned and guided by a charter the provides a general description of the goal of test.
- Documentation for exploratory test is very light weight.
- Coverage is determined by the use of charter helps to define the task and objectives of the testing.
- The charter use to specify - what is to be tested, what is the goal, and what is considered to be in and out of scope, and sometime it also indicates what resources will be committed.

Q2. Describe the experience-based testing technique? State its limitation and weakness

Ans There are four major types of experience-based testing discussed in here:-

1. Error Guessing
2. Checklist based
3. Exploratory
4. Attacks

- Error Guessing :- is commonly used in risk analysis to 'guess' where errors are likely to occur and to assign a higher risk to the error-prone areas.

It is to determine the potential errors that might have been introduced during the software development and to devise methods to detect these errors as they manifest into defects and failures.

- Checklist based :- it is used by experienced testers who are using checklists to guide their testing. The checklist is basically a high-level list. This include items to be checked, lists of rules, or particular criteria or data conditions to be verified.

- Exploratory :- Exploratory testing occurs when the tester plans, designs and executes tests concurrently and learns about the product while executing the tests.
Exploratory tests are planned and usually guided by a test charter that provides a general description of the goal of the test.
The process is interactive and creative.

- Attacks :- Software attacks are focused on trying to induce a specific type of failure.
When performing attack testing, you should consider all areas of the software and its interaction with its environment as opportunities for failures.

Strengths and weakness

- Strengths

⇒ These techniques can be very high yield when done well and methodical

⇒ They are fast and focused and will tend to find the more obvious bugs first.

⇒ These techniques work well in time restricted situations and on projects where the documentation may be less than attractive.

- weakness

⇒ Inexperienced testers will not get the same bug yield as an experienced tester.

⇒ They may not know the likely vulnerable areas to target for an attack.

⇒ They may not be able to "guess" the types of errors that will occur.

VALENTINE
Page No.:
Date: / /

Q3. Describe checklist - based testing

⇒ Checklist - based testing is used by experienced testers who are using check lists to guide their testing. The checklist is basically a high level list, or a reminder list, of whatneeds to be tested. This may include items to be checked, lists of rules, or particular criteria or data conditions to be verified.

Checklists are usually developed over time and draw on the experience of the tester as well as on standards, previous trouble-ares, and known usage scenarios. Coverage is determined by the completion of the checklist.

Q4. Explain Software Quality Attributes for test analysts and technical test analysts.

Ans Software quality attributes for Test analyst

→ 1. Accuracy
2. Suitability
3. Interoperability
4. functional security
5. Usability
6. Accessibility

- Software quality attributes for Technical test analyst

→ 1. Efficiency
2. Technical security
3. Reliability
4. Maintainability
5. portability

→ These attributes address "how" the system is delivering the functionality

VALENTINE
Page No.:
Date: | |

Unit-4

Q1. Describe the test process for usability and accessibility testing

- Test process for usability and accessibility testing

1. **planning issue :-** The first thing here is to ensure that the right people with right skills are available when needed.
The test analyst must have this knowledge as well as the skills required to conduct the other areas of testing.
Usability testing environment must be same as user environment.

2. **Test Design :-** Test cases must be specifically designed to test for understandability, learnability, operability and attractiveness.

Need to approach the product from several different angles. And, any documentation that are prepared for the project should be reviewed to determine both usability issues and usability test scenarios

3. Test execution :- Here it comes inspecting and reviewing

- Usability inspections and reviews are sometimes conducted by the test, but they are generally more effective if the user is present as well.

4. Reporting :- After executing the usability test cases we have designed and documenting the result as we would any testing.

- It surveys for questioning and answering
- There are standard publicity available usability surveys such as (SUMI) Software Usability Measurement Inventory and (Website) website Analysis and Measurement Inventory
- SUMI provide a set of Measurement against which we can evaluate the usability of the software.
- WAMMI is used to supply ongoing feedback from users regarding a website

This is done by presenting a question series to the user when they leave the website.

Q2. Explain typical efficiency risks and types of defects that occur.

(write it in tabular format)

- Typical efficiency risk —

1. Design risk : insufficient processing capacity, poor architectural design, or an inappropriate software design.

2. Risk that system components are incorrectly configured or are unable to physically handle high volumes of data or transactions.

3. Insufficient network bandwidth

4. Risk that the software implementation does not use resources efficiently

VALENTINE
Page No.:
Date: / /

- Type of defect that may occure.

1. User response times too slow. The application cannot be used efficiently
2. Stress situations cannot be handled. System crashes when large number of users logged on. System growth limits are reached too quickly after introduction into production.
3. Database queries with large - volume response impair the system for other users.
4. Main memory exhausted under stress loads. Database or files capacity exceeded after a period of continuous usage.

VALENTINE
Page No.:
Date: | |

Q3/ Q4e Explain approaches to efficiency tests.

Ans Test should be based on operational profiles that represent typical system usage patterns

- performing static analysis or a code-level technical review may be a useful way to evaluate programming practices and their impact on efficiency
- where time-critical components have been identified as test objects performance profiling may be performed with specific tools or benchmarking carried out against predefined criteria. CPU usage may be monitored when performing these test.
- Volumes tests are performed on the business processes that rely on large data transfers.

  Eg: information searching.

Q5. What are points to consider for using tools for efficiency testing?

Ans. It is generally not a good idea to perform meaningful and repeatable efficiency tests manually, our test planning needs to consider a number of test manually & auto environment points that include required testing tools.

For tooling, the following points should be considered.

- Simulation needs
A tool must be able to generate the loads required for our planned tests.
These may include the high loads often needed for stress testing and perhaps also scalability testing. Such loads are typically defined in terms of virtual users that represent the real system users to be simulated by the tool.

VALENTINE
Page No.:
Date: / /

- Financial consideration.
For large-scale simulation that may be complex and required many virtual users, the cost of tooling can take up a major part of the available testing budget.