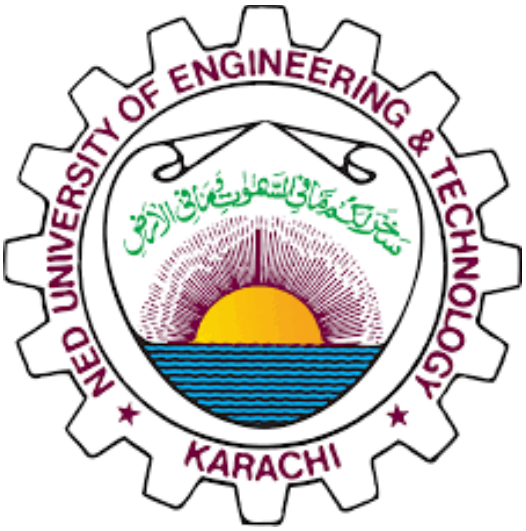
Software Quality Engineering (SE-309)
Assignment



Group Members:

- Muhammad Hamza Shahab (SE-004)
- Rana Muhammad Ibrahim (SE-034)
- Muhammad Abdullah Hayat (SE-041)
- Muhammad Khurram Meraj (SE-045)

Department: Software Engineering

Batch: 2018

Year: TESE - (A)

Submitted To: Miss Sana Fatima

Group ID	**Member's Name & Roll no.**	**Project Name**	**Tasks Distribution**
G-001	Muhammad Hamza Shahab **(SE-004)**	Appointment Scheduling System	**Module 1:** Roll # 34 (Q1, Q2, Q3 + Module 3: Appointment Creation)
	Rana Muhammad Ibrahim **(SE-034)**		**Module 1:** Roll # 41 (Q4, Q5, Q6 + Module 3: Admin Panel Management)
	Muhammad Abdullah Hayat **(SE-041)**		**Module 2** Roll # 04 (Q7, Q8, Q9 + Module 3: Email Confirmation)
	Muhammad Khurram Meraj **(SE-045)**		**Module 2:** Roll # 45 (Q10,Q11,Q12 + Module 3: Organizations Management)

1. **Compare McCall's and ISO 9126 quality factor models and discuss their pros and cons, which model do you prefer when you are asked to develop an off-the-shelf application and why?**

Comparison between McCall's And ISO 9126 Quality Factor Models:

McCall's Quality Model	ISO 9126 Quality Model
McCall's model focuses on internal as well as external characteristics which are related to user as well as developer point of view.	ISO 9126 mainly focuses on external characteristics which are mostly related to user point of view.
It contains total of 11 quality attributes, which are categorized into 3 groups i.e. product revision, product transition, and product operation.	It contains total of 6 quality attributes i.e. functionality, usability, efficiency, reliability, maintainability, and portability.
The quality attributes are further divided into internal attributes called quality criteria.	The quality attributes are further divided into 21 quality factors.
In McCall's model, one quality attribute affects multiple attributes.	In ISO 9126 model, one quality attribute affects only one attribute.
Interoperability and testability are high level characteristics in McCall's model.	Interoperability and testability are sub characteristics of functionality and maintainability in ISO 9126.

Pros:

McCall's Quality Model:

- This model played an important role and brought a revolution in software field by serving as basis for many upcoming quality models.
- It provides metrics for measuring of attributes which has multiple methods for measuring them.

ISO 9126 Quality Model:

- This model serves as hierarchical framework and divides attributes into groups and subgroups.
- It is easy to deal because it's one sub-characteristic does not affect another one.

Cons:

McCall's Quality Model:

- The model doesn't mainly focus on functionality of the software system.
- In this model, affecting one higher level attribute can change multiple other attributes.

ISO 9126 Quality Model:

- This model is much less flexible than other quality models.
- Customization is required for this model before applying it to any software.
- It mainly focuses on external characteristics which is only beneficial for user point of view.

Preferred Model:

As a software developer, I would prefer **McCall's Quality Model**, because it focuses on both internal and external characteristics of the project which is not only important for developer point of view but also for the end-user point of view. Also, maintainability is easy in McCall's model as compared to ISO 9126, and maintainability is one of the most important attribute nowadays for the software projects.

2. Choose your desktop/web-based project and map the software requirements to quality attribute using McCall's quality model.

Software Requirements:

1. User should be able to book appointment through their respective accounts.
   Integrity

2. The system should be available 24 × 7 for users to register, login, and use the system to schedule appointments.
   Reliability

3. The software is developed in such a way that its maintenance would be easier in future.
   Maintainability

4. The response from server must not take more than 5 seconds for each request.
   Efficiency

5. The web application should be operable in different operating systems.
   Portability

6. User will be able to update their personal information on their profile.
   Usability

7. User will log in first to use all the functionalities of the web application
   Integrity

8. The application should be compatible for different browsers.
   Portability

9. UI of the web application should be simple and easy to use.
   Usability

10. The application should be responsive for different devices.
    Flexibility

11. Continuous deployment will be set up for the main branches of the project so that the developer doesn't have to manually deploy the code each time.
    Integrity

12. Only correct data with specific data type will be stored in the database.
    Correctness

3. Identify the development plan & quality plan objectives, suggest ways in which each objective contributes to successful and timely completion of project.

Development Plan Objectives:

1. It identifies the team members involved in the project development as well as the tasks for each and every member.
2. It describes the project requirements and specifications.
3. It helps in identifying the progress of the software application.

Quality Plan Objectives:

1. It helps in identifying the quality factors that needs to be fulfilled.
2. It helps in identifying the best and most efficient approach for testing.
3. It describes responsibilities and resources during the development of the project.
4. It also helps in identifying risks and defects of the system.

Objectives Contribution for Timely Completion of Project:

1. Multiple techniques are used for the estimation of resources like WBS or Planning Poker. The project is broken down into small activities of few days and tasks are assigned to each individual of each team.
2. Many techniques are used to identify the progress of the project. Gantt charts and other techniques like calculating critical path are used to identify overall project progress from not only the development point of view but also from the other aspects.
3. Decisions related to testing techniques and approaches are made in the early stages. It is identified early that what testing techniques will be most suitable and efficient for the respective project.
4. Proper documentation is made to prevent the defects and risks in the project.
5. Reports of projects are made and updated consistently so that the flow and management of the project can be made easy.

4. a) It is said that with respect to subjects where qualitative and quantitative requirements can be defined, the quantitative alternative should be preferred. Do you agree? Justify your answer. Explain why the customer preferred the quantitative option? Also explain why developers preferred the quantitative option?

Yes, I agree. In terms of requirements, quantitative alternative should be preferred. Requirements plays important roles in the development of any product. These requirements are mostly provided by the stakeholders. All stakeholders belong to different technical backgrounds or different domains. Since qualitative requirements do not have any standard interpretation, there is a possibility that all these requirements can be interpreted in different ways as their own subjective opinion.

For example, a requirement, which states, "*The display of screen should be clear"*. This is not a proper requirement as the term clear is open to wide range of evaluations, which lead to dissatisfaction from the customer even developer is satisfied. It should be as: "*The display of screen must have clear resolution of 4K (4096 x 2160 pixels)*".

Above example clearly shows that irrespective of the subjective opinion of any stake holder, the requirement will interpret the clear resolution as only 4K resolution. Hence, using quantitative approach in requirements helps stakeholders reach on common ground for interpretation of the requirement.

The customer prefer this approach because of following reasons:

- It's easier for them to determine the productivity of the product (i.e. evaluation) and comparison of the developed product from the product defined in requirements. It's easier to deal in quantitative approach as the parameters will be numeric obtained by standard testing software and results will not be biased on any subjective opinion.

- There will be no post development conflict over any requirement based on the self-interpretation and estimation. For example, if the requirement only state that response time should be less. It means that it can vary over a whole number line and developer can misinterpret it and customer and developer thinking over this requirement will vary. So if the customer state the requirement as "*The response time should be quick vs the response time should be < 10ms",* it clear all the ambiguity.

- Based on the project, if the requirements are already collected and customer is just looking for development team, it helps customer in better resource usage and allocation and planning. On other scenario, if the requirements to be gather by developers, it also help developers to allocate proper resources based on reliability figures and productivity etc.

The developer prefers this approach because of following reasons:

- The quantitative requirements can be test easily to evaluate the quality of product. For example: *The response time should be quick vs the response time should be < 10ms.* It is easier to test as developer has some objective standard to compare results with. In first

requirement, developer will be using subjective opinion which has high possibility to vary from the customer opinion.

- The quantitative requirements are easy to determine the compliance of that requirement, and it is easy to cope up with the non-compliance situations by early corrections, in early stages.

- Since, requirement prioritization is one of important pre-development phase, quantitative parameters in the requirements helps in better prioritizing the requirement as it is independent of random subjective guessing.

b) Identify & list the qualitative requirements and convert them to quantitative requirements)

Case Study: A software system to serve the help desk operations of an electrical appliance manufacturer is to be developed. The help desk system (HDS) is intended to operate for 100 hours per week. The software quality assurance team was requested to prepare a list of Quantitative Quality Goals appropriate to certain qualitative requirements.

Qualitative Requirements	**Quantitative Quality Goals**
System should be able to run continuously.	The system should be able to recover in less than 50 minutes in more than 85% failures.
System should be consistent.	The failure time of the system should not exceed more than 2% of total time (2 hours) per week
System should be fault tolerant.	The continuous runtime of the system without shutting down per week should be minimum 98% (98 hours)
System should be efficient to handle maximum clients	The system must be stable enough to handle 50 clients in every 2 hours.
System should provide quick response to the clients	The waiting time for each client should not exceed 20 seconds in more than 85% calls
System should be user friendly	The user of the system should be able to know all the operations in less than 3 hours by having a training session and he should excel the system in maximum of 3 days.

5. **With respect to verification, validation & qualification, explain the difference between these three SQA activities**

Verification: Are we building the product right? It is the activity in which after every phase, it is determined that whether the output of the current phase or the product of current phase accomplish the requirements which were decided in the last phase.

In the more technical language, verification includes focusing on any given model in an enterprise modeling approach to determine the mistakes. It includes detecting the mistakes caused by the misunderstanding regarding the understanding of the meta-model (rules, constraints etc. determined during modeling phase).

It also includes determine the coherence between these models from:

- Different views (Organization, resources, Information, function etc.)
- Different Steps (Identification, concept, requirements, design, etc.)
- Different genericity degree (Generic, particular, partial)

After this activity, the output is either a proper model which is free from any error, or the model with few errors which arose from the incoherence of the models, incompatibility with existing models or bad behavior.

Validation: Are we building the right product? It is the activity in which after the development, it is determined whether the output or the product is free from failure and it is abide by with the software requirements.

In the more technical language, validation shows that the developed model / product shows the exact same representation as specified in the requirements. This activity helps in detecting any semantic errors, interpretation errors and inconsistencies.

So it is simply the activity to determine the deviation / misinterpretation of any product from its requirements defined for the product.

Qualification: It is the activity in which it is determined that in the development life cycle whether the standards set by the upper bodies like organization or the working environment are fulfilled or not.

In the more technical language, qualification involves determining the deviation of the approaches held in all phases of development from the standard approaches and from the standards set by the development team or organization. It involves all the standard practices of design, analysis and coding.

Can a project which successfully passed verification and validation reviews but failed the qualification review adequately supply users with the information needed? Explain your answer.

Yes, a project can successfully pass the verification and validation reviews but still fail the qualification review yet having all the features and requirements correct due to following reasons.

- The deadline is tough to meet so the developing team let go some of the formalities which should be taken care on of in order to pass qualification review.
- The team is new to the organization or professional work and is not aware of the working/organizational standards.

6. Explain the cleanroom approach as a quality process in your own words.

What is Cleanroom approach?

Cleanroom approach is a theoretical approach, which is based on collaborated team work, allowing organizations to have more predictable development and certification of software products under statistical quality control. The main philosophy behind this approach is to develop such systems which show 0 defects / failures in use.

The name cleanroom is inherited by the approach called hardware cleanroom which holds the key principle of preventing defects than removal of defects.

This modern approach is preferred by large enterprises as this approach is all about writing the code increments correct in the very first phase and verifying their accuracy before test phase, rather than depending on the expensive defect-removal processes.

What are the steps of Cleanroom approach?

There have been several steps and practices in this approach which have been updating over course of time, but these have been the fundamental activities or steps in the approach. They work in two teams: certification and development team.

1) Specification

In the approach, on the very first stage, two specifications are produced.

(a) Functional Specification

It defines the behavior of the external system in all scenarios in which system can be used. These specifications provide the foundations for incremental software development.

(b) Usage Specification

It defines the scenarios in which system can be used and the probabilities of correct or incorrect system usage. These specifications provides the foundation for test cases generation for the statistical testing on incremental basis and certification of quality.

2) Increment Planning

Both teams collectively based on the specifications design a plan to develop increments based on the size of the code of systems measured in KLOC. These increments adds and shape the final system.

3) Design and Verification

For each increment, development team repeats the cycle of design and correctness verification. Whereas, in parallel, based on the gathered usage specifications, test cases are generated by the certification team to test the expected use of increments.

4) Quality Certification

After the completion of the increment, the developed increment is integrated with the previous increments and are executed for the statistical test cases. For that increment and the integrated increments, test cases are executed against the functional specification to determine the correctness. Mean failure time is calculated and statistical measures are calculated via quality certification model. This certification process is done regularly throughout the development life cycle. Higher level increments are dealt first than the low level increments.

5) Feedback

Errors found in testing is then forwarded back to development team for the eradication of errors. This step gives overall feedback for every increment and the integrated increments so that if the quality is quite low, process improvement can be done.

7. **To perform unit testing on(Module1, part ii). What software testing technique/methodology do you choose? And why. To Perform system testing, what technique do you prefer and why? Write down the detailed test cases for load & install/uninstall testing (refer to project choose in module1 part ii.).**

Unit Testing

Unit testing tests the smaller components and units of software individually. It allows us to validate if each element of the application is functioning as expected or not. We perform unit testing during the implementation phase (coding) of a software development lifecycle.

I prefer White Box Testing Technique to perform **unit tests** on the **"Appointment Scheduling System"** because it enables the testers to check the **internal working** and **code defects** in smaller units of the application during the early stages of development.

Since the Appointment Scheduling Application involves direct interaction with the end-users, the application must be robust and user-friendly.

It can happen if the internal workings and the software's overall structure are well built and are free from defects. Therefore, white-box testing aids in the individual testing of each system component and source code.

System Testing

System testing validates a fully integrated software product. It evaluates the end-to-end specifications of a system.

I prefer Black Box Testing Technique to perform **system tests** on the **"Appointment Scheduling System"** because it focuses on the **application's functionality**. System testing tests the **external workings** of the system from the **user's perspective**.

Black box testing is applied to perform **system tests** to determine the system's compliance with the requirements. Using the black box testing would ensure that every application module would work well after integrating into the central system.

Test Cases

Project Name: Appointment Management System

Module: Appointment Dashboard Module

Created By: Hamza Shahab

Date: 17-4-2021

Load Testing

Test Scenario ID	Test Scenario Description	Test Case ID	Description	Steps	Test Data	Expected Outcome	Actual Outcome	Status
TS-AM-01	Testing Load on Application (Dashboard / Admin Panel)	TC-AM-01	Verifying response time of each appointment creation	-Login dashboard as a user -Create an appointment -Note the time it takes for the scheduling of an appointment	Name: Hamza Shahab Email: abc@gmail.com Appointment Date: 18-4-21 Appointment Time: 9 AM	The appointment should be created and appear on the user dashboard within 5 seconds.	The appointment does appear on the user dashboard in less than 5 seconds	Pass
		TC-AM-02	Fetching all appointments of a patient as an admin	-Login to the admin panel as an admin -Enter the patient ID -Enter the filters, including the period and duration of appointments	Patient ID: 123 Start Date: 17-3-21 End Date: 17-4-21 (Will include all the appointments created by patient id 123 during the last month)	The system should list all the appointments within 3 seconds	It takes 4 seconds to fetch all the appointments from the database for a particular patient	Fail
TS-AM-02	Testing Load on Database	TC-AM-03	Testing database components under heavy loads	-Add a significant record of patients simultaneously using simulation or manually -Similarly, create a large	Large data set of patients Large data set of appointments for each patient	The database should continue to operate and deliver data normally after adding all the data.	The database successfully adds all the data input into the application	Pass

				number of appointments for each patient				
TS-AM-03	Testing Network Load	TC-AM-04	Testing the network capability under heavy loads	-Send multiple POST and GET requests to the backend server from different workstations -Check the performance and response of the server under heavy load	POST and GET requests to the backend server with different payloads for data delivery to the application	The application should continue to perform well without experiencing any downtime or failure.	The server crashes due to sudden heavy traffic from the application	Fail

Installation Testing

Test Scenario ID	Test Scenario Description	Test Case ID	Description	Steps	Test Data	Expected Outcome	Actual Outcome	Status
TS-AM-01	Pre Deployment Scenario	TC-AM-01	Building the web app through different commands	-Run “npm run build” to create a build folder of the web app	Commands: -npm run build	It should create the build folder correctly.	It creates the build folder without any errors.	Pass
		TC-AM-02	The web app is uploaded to the selected platform (Azure in our case) during the dev stage to verify if the build folder is deployed correctly on Azure.	-Set credentials for “Azure App Services” to correctly deploy the app on Azure -Run “npm deploy dev” to start the deployment process in the dev stage	Commands: -npm deploy dev	It should upload the build folder to the Azure App services correctly	The build folder is uploaded correctly to Azure	Pass

TS-AM-02	Production Environment Check	TC-AM-03	Production environment's **continuous deployment and integration** feature	-The Azure pipelines are checked if they are functioning correctly by auto-deploying the build folder in the production environment	Commands: -npm deploy prod	The continuous deployment feature **(CD/CI)** should work correctly	The web app is ideally auto deployed on Azure	Pass
		TC-AM-04	To check the production environment database for possible discrepancies.	-Run the metrics test for database and cloud environment to ensure everything is ideal for deployment of web app	N/A	The database should robust and should be available at all times	The tests reveal that the cloud environment's database is available 24/7 and is robust	Pass
TS-AM-03	Post Deployment Scenario (Release)	TC-AM-05	To test the website on multiple devices and workstations and noting the site's performance on all devices.	-Test the website on different desktops and mobiles, and its performance is noted on all devices using the **Lighthouse score**	N/A	The website should work correctly on all devices without any downtime or errors	The site does not perform well on mobile devices and is less performant, having a lower **Lighthouse score** on mobiles	Fail

Uninstallation Testing

Test Scenario ID	**Test Scenario Description**	**Test Case ID**	**Description**	**Steps**	**Test Data**	**Expected Outcome**	**Actual Outcome**	**Status**
TS-AM-01	Verifying the uninstallation of web app	TC-AM-01	To disconnect the master branch of the web app from the Azure **CD/CI** pipelines.	To disconnect the master branch, go in the deployment section on the Azure portal and disconnect the Azure app service from the GitHub repository where the build folder of the web app resides.	N/A	It removes the site successfully from the URL provided by the Azure cloud service.	The site is removed successfully from the Azure cloud, and users can no longer access it through the domain provided by the Azure App service.	Pass

8. Compare white box & black box testing techniques with pros/cons. Which testing technique is best suited @ what testing level? Discuss.

Comparison between White Box and Black Box Testing

White Box Testing	Black Box Testing
White box testing tests the software's infrastructure and source code	Black box testing is the method that involves the reviewing of structure, design document, and implementation of the application.
It requires the tester to know the implementation details and should be able to interpret the code along with its internal workings.	It does not require the tester to know the implementation details or interpret the code's internal workings.
It allows the tester to verify the internal structure of the system's components.	It allows the tester to verify the input and output methods of the system.
It focuses on the internal structure of the software. Hence called structural testing or logic-driven testing.	It focuses on the functionality of the application hence called functional or specification-based testing.
The test cases in white box testing are based on detailed design.	The test cases in black box testing are based on requirements.
Testers do this testing from the development perspective.	Testers do this testing from the user's perspective.
It is applicable mainly at the unit testing level.	It is applicable at every software testing level, including unit, integration, system, and acceptance testing.
The internal structure of the system is known.	The internal structure of the system is not known.

Pros and Cons of White Box Testing

Pros	Cons
It applies to unit testing, which allows the testers to locate the errors quickly in small chunks of code and earlier stages of development.	It is very time-consuming and complex.
It helps the testers to detect the security issues or flaws by performing attacks which makes	It requires testers to have high-quality programming skills and an internal

dealing with the side effects easier.	understanding of the software structure they are testing.
It allows experienced testers to identify problematic code quickly due to unit testing, which saves time.	It requires highly proficient testers who can carry out the testing efficiently. Therefore the resources are limited.
It helps in the optimization of code due to the programming implementation.	Due to the limited amount of resources and testers, this method is relatively expensive.
It ensures stability by finding bugs efficiently due to the thorough examination of the application at more minor levels.	Regular maintenance is required because the application's internal structure and code are bound to change frequently, which can cause the previously implemented tests to break and defies all the previous assumptions.

Pros and Cons of Black Box Testing

Pros	Cons
It helps identify the issues in the functional specifications of the application.	There is limited coverage since the testers can only perform a small number of possible test scenarios.
It helps in quick test case development since the testers are only concerned with the GUI of the software and how a typical user would interact with it.	If the testers do not clearly understand what they need to test, then the testing process would become inefficient and time-consuming.
It simplifies the testing process since the tester only focuses on the inputs and outputs of the application.	The testers have limited knowledge about the application's internal workings, due to which testers do not review some features of the application.
It can be easily performed without any programming knowledge or having a technical background.	Since the testers and development team are separate, some test cases can be duplicated or missed during the testing process.
It is unbiased because a different team performs testing with no knowledge of the application's code, thus removing bias.	It is difficult to carry out the maintenance if the UI of the application constantly changes.

Testing Techniques at Different Testing Levels

Now let's discuss which testing technique is best suited at what testing level.

White Box Testing

White box testing is applicable at the **unit testing level** of software testing. It helps in identifying defects during the early stages of the software development life cycle. It helps prevent the minor problems from becoming a cluster of more significant issues after the development team integrates each unit's code into the central system.

Therefore, it is performed before integration with the already tested code, which prevents errors from occurring during the later stages of development.

Black Box Testing

Black box testing ensures that the module works well with the central system after getting integrated. It is **applicable at every level of software testing**, which is as follows:

- **Unit testing:** It tests the interface against the specifications provided by clients.
- **Integration testing:** identifies and removes errors while integrating different components of the system.
- **System testing:** analyses the system against the requirements.
- **Acceptance testing:** in which the users validate and accept the product by testing in unexpected situations.

9. Question 9

Martin Adams, an experienced project leader at David's Software Ltd., a medium-size software house, has been appointed project leader for development of an advanced help desk software system for a leading home appliance maintenance service. This is the 12th help desk system developed by his department in the last three years.

The current project is somewhat special with respect to its timetable. The contract with the customer was signed 6 days after submission of the proposal, and the development team is scheduled to begin working at full capacity, with 8 team members, 10 days later. The contract offers a significant early completion bonus for each week below 26 weeks, but determines high late completion penalties for each week after 30 weeks.

In a meeting with his superior, Adams claims that the comprehensive proposal documentation "as is", which has been thoroughly checked by the contract review team, should serve as the project's development and quality plans. His superior does not agree with him and demands that he immediately prepare comprehensive project and quality plans, according to company procedures.

1. Do you agree with Adams? If yes – list the arguments that support his claim.
2. Do you agree with his superior? If yes – list the arguments that support the superior's claim.
3. Considering the circumstances of the project, what, in your opinion, should be done in this case.
4. Comparing the circumstances described here to those of the opening anecdote, are there any justifications for different recommendations?

1. No, I **disagree** with him. Although the proposal material can become a part of the development and quality plans, **the proposal itself cannot become the project and quality plans** as a whole.

The project manager would have to **elicit some extra details and elaborations** to complete the development and quality plans.

A project manager is **responsible for preparing documentation** of the projects. Therefore, he should **provide comprehensive quality plan** and **project development** according to organization's procedures.

2. Yes, I **agree** with the superior, keeping in view that the contract includes a **short completion time**. It also involves **bonuses** and **late completion penalties** upon the early or late completion of the agreement, respectively.

Therefore the project leader should make a comprehensive project development and quality plan. All the available resources should be examined, including external participants. The team members should also do a careful inspection to **resolve risks** during development.

3. Considering the project's current situation, in my opinion, **Adam should agree with his superior** and make a comprehensive project development and quality plan as per the company procedures.

The project manager's responsibility is to provide **complete project documentation** so that his **team may have a clear idea about the project scope** and to **prevent risks** during the later stages of development.

4. According to the circumstances mentioned here, seven months (26 weeks) have passed since submitting the proposal, a relatively long period in a project lifecycle.

Therefore this difference between the current circumstances and the opening anecdote **justifies the preparation of new development and quality plans**, and the proposal material should be updated.

In this case, partially updating the proposal material would be sufficient.

10. Explain the 8 dimensions of s/w quality (a/c to ISO 25010) also suggest how we can measure them in a software project.

The 8 dimensions of software quality according to ISO 25010 are:

1. Functional Suitability
2. Performance Efficiency
3. Compatibility
4. Usability
5. Reliability
6. Security
7. Maintainability
8. Portability

Explanation	How to Measure
Functional Suitability	
Capability of the software to have set of functions that are required by the user. **Functional Completeness:** The delivered system should have a complete set of functions according to the specified objectives. **Functional Correctness:** The ability of system functions to provide accurate results as expected. **Functional Appropriateness:** The ability of functions to be easy to use in the way that the required task can be performed appropriately.	Integration testing Automation Automation of UI
Performance Efficiency	
Response of the software system under specific conditions. **Time Behavior:** The measure of the time taken by the system in performing functions. **Resource Utilization:** The ability of the system to efficiently utilize the resources while performing the functions. **Capacity:** Measure of the capacity of the system to meet the specified requirements.	Cycle time Throughput Page load time Load testing
Compatibility	

Ability of the system to correctly communicate with the other software. **Co-existence:** Ability of the system to perform its functions in a sharing environment with other software without affecting the other software. **Interoperability:** Ability of the software to correctly connect with the other software that are being used and exchange the information.	Different platform testing Various environment testing Automation
Usability	
Measure of the system to achieve its specified tasks and objectives with effectiveness. **Appropriateness recognition:** Users should be able to identify if the product is according to their needs. **Learnability:** Measure of how easy is the product to for the specified users to learn it to use effectively. **Operability:** Measure of how easy is the product to operate and perform tasks. **User error protection:** Measure of how much system is efficient to guide the user from doing errors. **User interface aesthetic:** Measure of how the interface is easy to understand for the user. **Accessibility:** Measure of how the system's functions are usable and accessible to the user for completing specified tasks.	Time for specified tasks User satisfaction Error rate Software engagement time
Reliability	

Capacity of the software to maintain its performance over a specific time period or over unexpected time **Maturity:** Capacity of the system to work under normal conditions and complete specified goals. **Availability:** System should be accessible when required to use. **Fault Tolerance:** Capacity of the system to perform accurately in presence of hardware or software faults. **Recoverability:** Capacity of the system to recover from a failure and carry on its goals accordingly.	Number of unit tests Number of integrations tests Installation under observation
Security	
Capacity of the software system to defend the cyber-attacks and have appropriate data access to every user. **Confidentiality:** The system should make sure to allow only authorized personals to access the sensitive data. **Integrity:** Capacity of the system to defend the unauthorized access to data, or changes in the system. **Non-repudiation:** Ability of the system to stop the repudiation of the actions. **Accountability:** Ability of the system to uniquely identify each entity. **Authenticity:** Ability of the system to prove the identity of the resources.	Known vulnerabilities Number of breaches Cyber-attacks and threats over time
Maintainability	

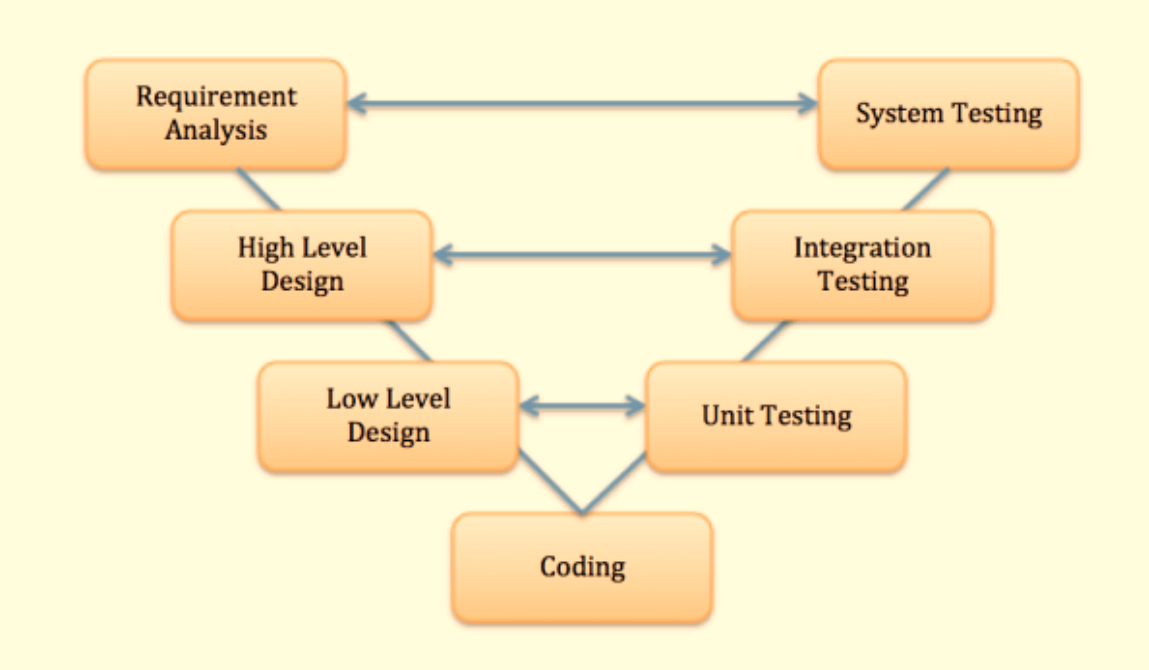
Capacity of the system to accept the changes so that it can be modified and improved over time. **Modularity:** Ability of the system to independently accept the changes in one component without affecting the other. **Reusability:** Ability in the software system that its components can be reused in other systems. **Analyzability:** Ability in the system to make it possible to analyze the impact of the modifications on one part to other. **Modifiability:** Capacity of the system with which its components can be easily modified or changed without affecting the existing quality. **Testability:** Capacity of the system to allow the tests to be done on specific components after changes have been done.	Lines of code Static code analysis Bug count Cyclomatic complexity
Portability	
Capacity of the system with which it can be easily transferred from one hardware to another. **Adaptability:** Ability of the system to adapt to the changes in hardware, software or environment. **Installability:** Ability of the system to be easily installed or uninstalled on specified environment **Replaceability:** Ability of the software to replace any other software in the same environment with same goals.	Installation and uninstallation on different platforms Real testing environment.

11. Explain how v model software process helps to improve software quality in detail. also explain when or at what defect level we can stop testing?

V-Model Software Process:

V-Model is the extension of waterfall model where the process is completed in sequential manner in "V" shape. Here the testing is done alongside of the development phase so it is impossible to miss out any details in the requirement which may be the case in waterfall model. The waterfall model lacks behind because in the complex systems the requirement details may be not met completely and project would have to be redone, also the mistakes in architecture may not be addressed once the project is complete. Whereas, V-model carry on the testing in parallel to development phase, so when a development stage is complete its testing is done before proceeding to next stage. This model is also known as Validation and Verification

In the V-model the SDLC and STLC are parallel to each other as shown in the figure.



When to stop testing:

There are several testing criteria based on which testing can be excited:

I. **Requirement:**
The testing can be stopped once all the requirements are met and code coverage is reached at the desired level.

II. **Defects:**
When the defect rate falls and all the major defects have been reported and documented.

III. **Real World environment:**

When the software gives the expected results in the environment that the user will be using that means that major requirements have been completed with least bugs and testing can stopped.

IV. **Testing deadlines reached:**

Testing can be stopped when the testing deadlines are arriving and software is able to pass all the test cases documented.

There are several other conditions which affect the stopping of the software testing. The decision to stop testing is also based on the model chosen for development according to which testing was started.

12. List and briefly describe various causes of s/w errors

There are several causes of software errors from which some are listed below:

1. **Faulty Definition of Requirement:**
   The requirement that are not defined correctly are one of the main problem in software errors. The requirements prepared by the client may have absence of important requirements, partial or wrong definition of requirements. The requirement containing unnecessary information is also a cause of failure of requirement.
2. **Client-developer communication failures:**
   Miscommunication between client and developer is also one of the biggest cause of software failure. If the client is unable to explain the instructions to the developer in written or in oral form then the software will not meet its requirements. Developer may also be unable to describe the design problems faced by the system in development and may result in software errors.
3. **Intentional deviations from software requirements:**
   If the developer is himself deviating from the mentioned requirements and not giving the required software, this can cause software errors. The main reason for this cause is the reuse of the different components from previous software that may have changes that cannot be adopted by the current software and cause errors in integrating them together.
4. **Logical design errors:**
   If the algorithms and system design documents are not generated accurately according to the need of the software requirements document then this may cause errors and software may deviate from its main goals.
5. **Coding errors:**
   One of the most common type of the error are coding errors by the developers done in the development phase. The cause of these are the bad coding examples followed by the developers or misunderstanding the requirements. These errors can also be caused in wrong application of CASE.
6. **Noncompliance with document and coding instructions:**
   These errors are caused when the development or maintenance team is unable to understand the requirements and may make changes that are not acceptable in the requirements.
7. **Shortcomings of testing process:**
   The shortcoming are the result of incomplete test plans, failures in reporting or documenting errors. This increase the chances of more errors being neglected and going to users without being detected.
8. **Procedure errors:**
   The software may not be able to guide the user correctly to the path he should and may result in wrong task performance by the user and cause errors.
9. **Documentation error:**
   The errors in the design document and requirements document that are incorporated in the system can further increase the error in development and maintenance phases.

Also classify the causes of error a/c to the groups responsible for the error: the client, system analyst, programmer, testing staff- or it is a shared responsibility belonging to more than one group.

Type of error	Client's staff	System analyst	Programmers	Testing staff
Faulty Definition of Requirement	Responsible	-	-	-
Client-developer communication failures	Responsible	Responsible	-	-
Intentional deviations from software requirements	-	Responsible	-	-
Logical design errors	-	Responsible	-	-
Coding errors	-	-	Responsible	-
Noncompliance with document and coding instructions	-	Responsible (for design)	Responsible (for coding)	-
Shortcomings of testing process	-	-	-	Responsible
Procedure errors	-	Responsible	-	Responsible
Documentation error	-	Responsible (for design)	Responsible (for coding)	-

13. MODULE 3

Feature 1 (Email notifications - Roll # 004)

Upon submitting a user's details and their appointment schedule, the system **sends a confirmation email** to the **user's email address** notifying him about his appointment's schedule. Furthermore, the system also sends an **email to the respective organization** where the patient has made the appointment to keep track of all of their **customer's appointments**. This **two-way confirmation** adds to the reliability of the whole **appointment creation procedure** and removes all the **ambiguity**.

Testing Table

Project Name: Appointment Management System

Module: Appointment Confirmation Email Module

Created By: Hamza Shahab

Date: 17-4-2021

Test Scenario ID: TS-AM-01

Test Scenario Description: Verifying the email notifications feature in Appointment Management

Test Case ID	Description	Steps	Test Data	Expected Outcome	Actual Outcome	Status
TC-AM-01	Submitting the form to test the email notification feature	-Fill user details form -Fill appointment details form -Click on submit	Name: Hamza Email: abc@gmail.com Organization: XYZ Appointment Date: 18-4-21 Appointment Time: 9 AM	An appointment confirmation email is sent to both the user and the related organization	The user and the organization both receive appointment confirmation emails	Pass
TC-AM-02	Leave the organization field name empty	-Fill in user details -Leave organization name while filling in appointment details -Click on submit	Name: Hamza Email: abc@gmail.com Organization: N/A Appointment Date: 18-4-21 Appointment Time: 9 AM	The confirmation email system marks organization as an empty field and does not let the user submit	The confirmation email system does not allow the user to submit the forms while the organization field is empty	Pass
TC-AM-03	Entering invalid user email	-Fill in user and appointment details -Enter invalid user	Name: Hamza Email: adafaf Organization: XYZ	The confirmation email system	The user cannot enter an invalid	Pass

		emails -Click on submit	Appointment Date: 18-4-21 Appointment Time: 9 AM	identifies the user email as invalid and prompts the user to resubmit	email and has to enter a valid email address	
TC-AM-04	Leave user email field empty	-Fill in user and appointment details -Leave user email field empty -Click on submit	Name: Hamza Email: N/A Organization: XYZ Appointment Date: 18-4-21 Appointment Time: 9 AM	The confirmation email system prompts the user to enter the required user email	The user has to fill in the user email, which is a required field	Pass

Feature 2 (Admin Panel Management - Roll # 41)

The admin panel is an exclusive feature in the Appointment Management System that lets the admin update and delete all the entities registered into the system. Let it be users, registered organizations, or the appointments created by the individual users.

Testing Table

Project Name: Appointment Management System

Module: Admin Panel Management

Created By: M. Abdullah Hayat

Date: 17-4-2021

Test Scenario ID: TS-AM-02

Test Scenario Description: Verifying the update and delete appointment functionality in the admin panel

Test Case ID	Description	Steps	Test Data	Expected Outcome	Actual Outcome	Status
TC-AM-01	Updating the appointment details of a user	-Go to the appointments section -Select the appointment to be updated -Update the appointment time and day -Click on save changes	Appointment Date: 29-4-21 Appointment Time: 5 PM	The selected appointment is updated with the new details	The admin successfully updates the appointment with recent details	Pass
TC-AM-02	Leave the appointment time field empty	-Go to the appointments section -Select the appointment to be updated -Update the appointment date but leave appointment time empty -Click on save changes	Appointment Date: 29-4-21 Appointment Time: N/A	The system prompts that the appointment time is empty, and it is a required field	The admin gets the error alert the system asks to enter the appointment time	Pass
TC-AM-03	Deleting an appointment	-Go to the appointments section -Select the appointment to be deleted -Click delete	N/A	The selected appointment is deleted successfully	The admin successfully deletes the appointment	Pass
TC-AM-04	Updating the user details	-Go to the Users section -Select the user to be updated -Update the user details, e.g., name, email, etc -Click save changes	Name: Abdullah Email: abc@gmail.com	The selected user's name and email gets updated	Admin successfully updates the user details	Pass

TC-AM-05	Leave the user email field empty	-Go to the Users section -Select the user to be updated -Update the username but leave the user email empty -Click on save changes	Name: Abdullah Email: N/A	The system prompts that the user email is empty and it is a required field	The admin gets the error alert the system asks to enter the user email	Pass
TC-AM-06	Deleting the user	-Go to the Users section -Select the user to be deleted -Click delete	N/A	The selected user gets deleted	Admin successfully deletes the selected user	Pass
TC-AM-07	Updating the organization details	-Go to the organization section -Select the organization to be updated -Update the organization details, e.g., name, email, etc -Click save changes	Name: XYZ Email: xyz@gmail.com	The selected organization's name and email gets updated	Admin successfully updates the organization details	Pass
TC-AM-08	Deleting the organization	-Go to the organization section -Select the organization to be deleted -Click delete	N/A	The selected organization gets deleted.	Admin successfully deletes the selected organization.	Pass

Feature 3 (Appointment Creation- Roll # 034)

Appointment creation is the most important feature of the whole application. It lets the user to create an appointment with the selected organization at the specific selected time slot.

Testing Table

Project Name: Appointment Management System

Module: Appointment Creation

Created By: Rana M. Ibrahim

Date: 17-4-2021

Test Scenario ID: TS-AM-03

Test Scenario Description: Verify the creation of appointment in Appointment Scheduling Website.

Test Case ID	Description	Steps	Test Data	Expected Output	Actual Output	Status
TC-AM-01	Verify appointment page UI	1. Load appointment page	N/A	Appointment page will load correctly	Appointment page loaded correctly	Success
TC-AM-02	Leave all fields empty	1. Click "Submit" button	N/A	Error message will be shown	Error message is shown	Success
TC-AM-03	Enter all fields	1. Enter Name 2. Enter Schedule Time 3. Select Organization 4. Enter Phone Number 5. Click "Submit" button	1. Name: Abdullah Hayat 2. Schedule Time: 4 PM, 25/04/2021 3. Organization: Ziauddin Hospital 4. Phone Number: 03331234567 5. N/A	Appointment will be created	Appointment is created	Success
TC-AM-04	Enter all fields with one field incorrectly	1. Enter Name 2. Enter Schedule Time 3. Select Organization 4. Enter incorrect Phone Number 5. Click "Submit" button	1. Name: Abdullah Hayat 2. Schedule Time: 4 PM, 25/04/2021 3. Organization: Ziauddin Hospital 4. Phone Number: 03331234567456 5. N/A	Error message will be shown	Error message is shown	Success
TC-AM-05	Enter all fields except one field	1. Enter Name 2. Enter Schedule Time 3. Select Organization 4. Leave Phone Number 5. Click "Submit" button	1. Name: Abdullah Hayat 2. Schedule Time: 4 PM, 25/04/2021 3. Organization: Ziauddin Hospital 4. Phone Number: N/A 5. N/A	Error message will be shown	Error message is shown	Success

Feature 4 (Organizations Management- Roll # 045)

Any organization or hospital that want to provide its services to patient can register on the system using this registration portal that is why organization registration is the integral part of this system. This section asks the user about the details of their organization and then creates an organization dashboard where patient can book their appointments and in the respective time slot of hospital.

Testing Table

Project Name: Appointment Management System

Module: Organizations Management

Created By: M. Khurram Meraj

Date: 17-4-2021

Test Scenario ID: TS-AM-03

Test Scenario Description: Verify the registration of an organization in Appointment Scheduling Website.

Test Case ID	Description	Steps	Test Data	Expected Output	Actual Output	Status
TC-AM-01	User Interface verification	1. Load registration page	N/A	Registration page will load correctly	Registration page loaded correctly	Success
TC-AM-02	Leave all fields empty	1. Click “Register” button	N/A	Error message “Please fill required fields”	Error message “Please fill required fields”	Success
TC-AM-03	Enter all fields	1. Enter Organization name 2. Enter available time 3. Select number of patients in a day. 4. Enter Phone Number 5. Click “Register” button	1. Name: SIUT 2. Available Time: 9 AM-12 AM 3. 20 4. Phone Number: 03331234567 5. N/A	Organization will be registered	Organization is registered	Success
TC-AM-04	Enter all fields with one field incorrectly	1. Enter Organization name 2. Enter available time	1. Name: SIUT 2. Schedule Time: 9AM-12AM 3. 20	Error message will be shown	Error message is shown	Success

		3. Select number of patients in a day. 4. Enter Phone Number 5. Click “Register” button	4. Phone Number: 03331234567456 5. N/A			
TC-AM-05	Enter all fields except one field	1. Enter Organization name 2. Leave available time 3. Select number of patients in a day. 4. Enter Phone Number 5. Click “Register” button	1. Name: SIUT 2. N/A 3. Organization: Ziauddin Hospital 4. Phone Number: 03331234567 5. N/A	Error message will be shown	Error message is shown	Success