CSC585: Assignment-1

Submitted by: **Malika Hafiza Pasha**

Question 1: What are the four main components of a software product? How does the quality of each component contribute to the quality of the developed software?

Answer:

The four main components of a software product are Computer Programs, Procedures, Documentation, Data.

i. **Computer Programs**
ii. **Procedures**
iii. **Documentation**
iv. **Data**

(i) Computer Programs:

The computer programs, which are essentially a set of instructions that guide the computer to accomplish the tasks that the user specifies, turn on the computer systems to run the necessary tasks. Computer programs contain code in a variety of forms, including executable, test, and source code.

(ii) Procedures:

Procedures outline the time, sequencing, handling of typical software product concerns, and other aspects of how software or project programs should be carried out.

(iii) Documentation:

The primary objective of the documentation is to offer direction or support to end users, maintenance staff, and developers of new software product versions. Numerous design reports, test reports, user manuals, software manuals, and other documentations are available.

(iv) Data:

Code lists, parameter lists, and standard test results are among the required data. Standard test data is used to ensure that bug patching and other software maintenance operations haven't resulted in any illegal changes to the code or program data, and to help discover the underlying reasons of any malfunctions.

Question 2: Distinguish between software error, software fault, and software failure, and suggest a case where in a software package serving 300 clients, a new software failure ("bug") appears for the first time 6 years after the software package was first sold to the public.

Answer:

Software error:

The most fundamental faults that arise during software testing are those that the programmer or software developer made when creating the code or design of the program.

Software fault:

A flaw is another irregularity that the tester finds during the software testing process. Unlike mistakes, which stem from a discrepancy between predicted and actual results, faults are caused by software defects.

Software failure:

A software failure occurs when it does not function as intended and generates unexpected effects. These errors are brought about by improper external behavior, which leads to non-specified software behavior.

Software Error	**Software Fault**	**Software Failure**
Errors committed during the program's code development.	Incorrect use of the program or processing processes	Using the software improperly, inputting information inaccurately, or providing incomplete information are all examples of human error
Errors can occur because of incorrect coding logic, looping, and syntax errors.	A mismatch that causes the software to behave wrongly and inconsistently with the requirements	Failures may also occur because of faults in how the program is used.

Changes in the way the software package is utilized may be related to the fault. As the product is used by more and more clients, new and unexpected use cases may arise, which could result in the discovery of problems that were previously unknown. In this case, before delivering a patch or new version of the product to correct the bug, the software development team might need to collect information about the new issue.

The clients who regularly use the software could be severely impacted by this new bug, thus the software development team needs to address it as quickly as feasible. The group may also want to consider implementing stringent testing protocols, carrying out regular software updates and maintenance, and keeping lines of communication open with their clients to better understand their needs and usage habits in order to lower the possibility of future mistakes.

Question 3: South cottage Inc. is a manufacturer of washing machines and dishwashers. The requirements document for the new control unit includes the following specification:

The firmware should be suitable for all six variations of the following year's washing machine models.

To which of McCall's factors do the above requirement belong? Explain your answer.

Answer:

One of McCall's quality factors, along with portability and maintainability, is that the firmware needs to work with all **six** variations of the washing machine models over the next year.

Software that is easily transportable between environments is said to as portable. Given that the firmware in this case needs to be compatible with all six washing machine model versions, it should be able to work effectively with a range of hardware configurations and not rely on a specific set of hardware parts.

Maintainability is the ability of software to be modified with ease in order to correct errors or add new features. Under these conditions, the software must be designed with future upgrades and maintenance in mind, and it must be simple to update in order to work with new hardware configurations. This is because the firmware needs to work with all six washing machine model variations.
By achieving both quality aspects, the software development team can ensure that the firmware will work well and be beneficial for a range of washing machine models and that it will be simple to maintain and upgrade over time to meet the evolving needs of its clients.

Question 4: Some people claim that testability and verifiability are different terms for the same factor. Do you agree? Justify your answer.
Answer:

The terms "testability" and "verifiability" refer to the same concepts. There is a substantial overlap between testability and verifiability.

Verifiability:
Elements of programming and design that facilitate efficient verification of programming and design are specified by verifiability requirements. Verifiability criteria that are most utilized are modularity, simplicity, and compliance to documentation and programming standards.

Testability:
Testability requirements cover how a software system operates as well as the testing process. Testability criteria are related to the specific program features that make testing easier, including providing log files and predefined intermediate outcomes. Before it can be used, the software system has to do automatic diagnostics to make sure all of its parts are in excellent functioning order and to get a report on any issues that were discovered.

Question 5: There are three major differences between software products and other industrial products. Identify and describe the differences. Explains how these differences affect SQA.
Answer:

Complexity:
Software's number of operating modes can be used to gauge its level of complexity. This group comprises hundreds of thousands, millions, and sometimes even more individuals. It is acknowledged in the software limited warranty agreement that it is not possible to guarantee correct operation for every one of the many modes of operation.
The operational possibilities of hardware items are far fewer, allowing for the accuracy of each operational mode to be tested and verified.

Visibility:
Software flaws are invisible and intangible, much like any missing parts of a software package or file.

Software listings must be shown on screen or in print using a predetermined method in order to facilitate the discovery of software defects or missing software components. However, it is more likely that the workers on the production line will notice a broken or absent piece when it comes to industrial products. Imagine a situation where an automobile on a production line had the incorrect headlight or a missing door.

Few opportunities exist to find defects: Software faults are typically only discovered during the development phase of a program. During this phase, software products are designed and programmed, and team members and SQA function personnel examine and test the software to find bugs.

Because the product must be fully functioning for all scenarios, regardless of usage or complexity, quality assurance is impacted by the product's visibility and will be challenging. The developer is uncertain that their product will be free of defects because the other changes affect how hard it is to identify software flaws.
The focus of quality assurance typically lies in process control and removing variation since manufacturing is typically a much more standardized and predictable activity.

Question 6: Eight issues characterize the professional software development and maintenance environment. Identify and describe these characteristics. Which of these environmental characteristics require managerial efforts for executing software development and maintenance projects? Explain why such efforts are needed.
Answer:

The Eight characteristics of the software development, maintenance, and environment are,

- Conditions contractual for SQA, maintenance, and software development
- Affected by the supplier-customer relationship.
- The necessity of cooperation
- Coordination and collaboration with other internal development teams are essential.
- The requirement for external partners in the software development project to cooperate and coordinate.
- The necessary product communicates with different software platforms.
- The requirement to carry out a project even when team members depart.
- The requirement for long-term software system maintenance.

Conditions contractual for SQA, maintenance, and software development

The contract outlines the terms and conditions that apply to both the customer and the software developer- maintainer. The contract has an impact on the delivery, installation, and running-in activities as well as the three different types of maintenance activities. It also has an impact on the software development activities, which include software development as well as delivery, installation, and running-in activities**.**
Each of these must deal with: Project functional requirements, Project budget, Schedule, Qualification requirements of project team members.

Affected by the supplier-customer relationship.

All phases of the software development and SQA processes are under the customer's watchful eye, so it is crucial that the project team consistently upholds a constructive working relationship with the

client. To complete the following tasks, teamwork is essential. Getting the data needed for development process from the project and talking with the client about their requests for adjustments, talking with the customer about their concerns with the project, and getting their approval for the work done and for changes the development team has proposed.

The necessity of cooperation

Since the customer is monitoring every stage of the software development and SQA processes, the project team must constantly maintain a positive working relationship with the client. Collaboration is necessary to finish the following tasks. The customer's data must be obtained for the development process, and the work completed and changes the development team has suggested must be approved by the client. Discussions regarding the client's concerns about the project and requests for adjustments should also be held.

Coordination and collaboration with other internal development teams are essential.

In the software industry, it is highly usual for multiple teams to work together to complete larger projects. Coordination and collaboration may be needed with the following**:**

- Other software development team within the same company.
- Hardware development teams within the same company. This is true when a hardware manufacturer development project includes the software development assignment.

The requirement for external partners in the software development project to cooperate and coordinate.

Project execution, especially large-scale project execution, frequently involves participation from one or more other entities, with which the development team must coordinate and collaborate:

- Providers of hardware and software
- Subcontractors and development teams that are outsourced.
- Partner development teams working on the project.
- Development teams belonging to the customers.

One relationship with the hardware development team serves to reflect the relationships between our software development team and the hardware development teams as well as the other product development teams.

The necessary product communicates with different software platforms.

Interfaces with other software systems are a common feature of software systems. These interfaces enable the exchange of electronic data between the software systems. The main types of interfaces are

• Input interfaces, which allow data from other software systems to be sent to your program.
• Output interfaces: these are the points on software systems where yours sends processed data to other systems.
• Interfaces for input and output in software, such as those found in laboratory and medical control systems.

The requirement to carry out a project even when team members depart.

During the duration of a software project, team members frequently depart for various reasons such as receiving a promotion, switching jobs, relocating to a different city, etc. The team manager has to recruit a new hiring or an existing corporate employee to take the place of the leaving team member.

Regardless of the amount of time and energy required to teach the new team member, the show must go on.
The same rules apply if a team leader leaves; however, in this case, management has the duty of designating a replacement team leader as soon as possible.

The requirement for long-term software system maintenance

Customers usually plan to use software systems for five to ten years after creating or purchasing them. Maintenance will eventually be needed during the service period. These services are typically required to be supplied directly by the developer. When software is developed internally, internal consumers have similar expectations regarding software maintenance of their software system.

The need for intensive, continuous managerial efforts is typically driven by environmental conditions. These efforts are undertaken in addition to the required professional efforts to guarantee project quality, or, to put it another way, to guarantee the project's success.

Question 7: The top management contributes to software quality by employing three main managerial tools. Identify the tools applied by top management to achieve the software quality objectives. Describe each tool in your own words and explain how it affects software quality.
Answer:

The three main managerial tools are:
- Create and maintain the company's software quality policy.
- Assign one of the executives to handle complaints about the quality of the program.
- Regularly evaluate management's performance in relation to software quality issues.

Software quality policy:
The organization's software quality policy as defined by higher management is somewhat wide in nature, even if it is stated in that manner. complies with the goals and intentions of the company. makes sure that the effectiveness of software quality assurance procedures, as well as the organization's quality and efficiency, are continuously enhanced. identifies and expresses the necessity of reviewing the software quality assurance goals. demonstrates a dedication to meeting project specifications and guaranteeing client satisfaction.
It is sometimes possible to transfer one organization's software quality policy statement "as is" or with just minor alterations to another.

The executive in charge of software quality:
- Top management is responsible for formulating the quality policy and conducting policy reviews.
- Identify the SQA goals for the system for the upcoming year.
- Creating an annual SQA activities schedule and budget is your responsibility.
- Identify the extent of software purchases and subcontractor services anticipated for the upcoming year.

The primary objective of Quality Software is to deliver software products and software maintenance services that completely meet customer requirements and expectations, on schedule, and within the predetermined budget.

Prioritizing and promptly attending to the client's needs, expectations, requests, and concerns. engaging employees in the process of establishing and committing to high-quality goals. accomplishing development and maintenance tasks correctly the first time around will reduce the need for rework and correction.

The phrase "management review" refers to the regular meeting that is focused on quality-related matters. The meeting's goal is to give executives a general understanding of the issues with software quality that their business is facing. Usually, meetings for management reviews are planned for several times a year, or at least twice. monitoring of authorized activities and dangers that were discussed at the previous management review meeting but have not been addressed. Performance logs with quality metrics updated on a regular basis. obtaining input on client satisfaction.

Management reviews' primary goals are to evaluate how well the SQA system complies with the company's quality policy. Assess if the organization's software quality management system has met the predetermined standards for quality. Check to see whether the organization's quality policy is followed by the SQA system. Increase the amount of resources you give the software quality management system as needed. In order to better serve customers and increase satisfaction, software goods and operating services should start receiving changes.

Question 8: The executive in charge of software quality issues is responsible for overall control of the performance of SQA activities. Identify the types of SQA activities under the executive's responsibility. Describe in your own words the activities that the executive has to perform to control such SQA activities.

Answer:

Types of SQA activities under the executive's responsibility are:

- Creating the department's annual program for SQA activities.
- Creating plans for the SQA system development inside the department.
- General supervision of the planned SQA development initiatives and the annual SQA regular activities program's execution.
- Making SQA issues known to executive management and advocating for them.
- Assure the establishment and achievement of quality goals.

Creating the department's annual program for SQA activities.

- Evaluate recommendations made by the SQA unit for the annual activities program to see if they can accomplish the goals of the SQA system.
- Review the activities program to ensure that it aligns with the type and scope of software and services provided by subcontractors that are scheduled for the following year. Examine the appropriateness of the workforce and additional resources planned for the SQA program's execution.
- Give your approval to the final version of the annual SQA activities program and budget.

Creating plans for the SQA system development inside the department.

- An examination of the patterns that are expected to have an instant effect on the software quality of the organization.

• Look into ideas for changing SQA, like creating new protocols that fit the new tools and SQA requirements.
• Developing programs for both newly recruited team members and seasoned software development teams.
• Final approval of the budgets and timelines for the SQA development initiatives that are planned.

General supervision of the planned SQA development initiatives and the annual SQA regular activities program's execution.

• Overall authority over the annual cycle of the activities program.
• An assessment of the development status of the SQA adaption initiatives
• Overseeing generally the actions necessary to fulfill the high expectations established by the teams' objectives.
• A review of adherence to SQA guidelines and standards is carried out based on internal quality audits.
• Oversight of software development projects' general adherence to budgets and schedules
• Overseeing the provision of first-rate maintenance services to clients both inside and outside the business.

Making SQA issues known to executive management and advocating for them.

• Bringing the proposed yearly activity plan and budget to final approval.
• Putting forward suggested SQA adaptation projects and related expenses for final approval
• The creation and management of regular management reviews for the caliber of the company's software.
• Start of management-level discussions on exceptional software quality occurrences, like major quality failures, project completion threats due to severe professional staff shortages, operational crises in the SQA unit, etc.

Assure the establishment and achievement of quality goals.

• Developing the department's annual budget and schedule for SQA operations based on the SQA unit's recommended plan.
• The suggested strategy made by the SQA unit served as the basis for the department's SQA systems plans.
• Management of the annual program for SQA activities and the department's development goals.