George Wise is an exceptional programmer. Testing his software modules reveals very

few errors, far fewer than the team’s average. He keeps his schedule promptly, and only

rarely is he late in completing his task. He always finds original ways to solve

programming difficulties, and uses an original, individual version of the coding style.

He dislikes preparing the required documentation, and rarely does he do it according to

the team’s templates.

A day after completing a challenging task, on time, he was called to the office of the

department’s chief software engineer. Instead of being praised for his accomplishments

(as he expected), he was warned by the company’s chief software engineer that he

would be fired unless he began to fully comply with team’s coding and documentation

instructions.

1. Do you agree with the position taken by the department’s chief software engineer?

2. If yes, could you suggest why his/her position was so decisive?

**Solution**

1. Yes.

2. The use of non-standard coding and documentation methods by George Wise cause:

– Extra difficulties to other programmers who have to develop software modules

that need to interface with George’s module that may result in errors.

– Extra difficulties to an inspection team and testing team that may result in higher

than regular rates of error detecting

– Extra difficulty to replacement programmer who might be recruited to continue

the work of George in the case of his leaving the company or being promoted to

a higher position in another project. Misunderstanding George’s coding and

documentation may result in software errors.

– Extra difficulty in performing maintenance task of failure repairs, adaptation of

the software to new customers and system improvement tasks

**It is claimed that the expanded definition of SQA supports those who are interested in**

**increasing client satisfaction.**

1. Do you agree with this claim?

2. If yes, provide arguments to substantiate your position.

Solution

1. I agree.

2. a. The expanded SQA definition adds the elements of managerial schedule and

budget control. As projects that do not keep scheduling and / or exceed their budget

tend to be under severe managerial pressures to save project resources and perform

procedures more quickly, the prospects of fully achieving all the quality goals and

an acceptable extent of software errors are reduced substantially. Thus the inclusion

of schedule and budget control in the SQA controls or reduces the prospect of failure –

increases the customer’s satisfaction. (b) Another element added to the SQA

expanded definition refers to software maintenance quality. Maintaining SQA

efforts throughout the software life cycle leads to the continuing satisfaction of the

client extended over long years of operation.

**When or at what defect level do we need to Stop Testing?**

* Stop the testing when deadlines like release deadlines or testing deadlines have reached.
* Stop the testing when minimum no of the test cases have been completed with some prescribed pass percentage.
* Stop the testing when the testing budget comes to its end.
* Stop the testing when the code coverage and functionality requirements come to the desired level.
* Stop the testing when the bug rate drops below a prescribed level.
* Stop the testing when the number of high severity Open Bugs is very low.
* Stop the testing when the period of beta testing/alpha testing is over.

The Software Development Plan (SDP) describes a developer's plans for conducting a software development effort

Quality plan is a document, or several documents, that together specify quality standards, practices, resources, specifications, and the sequence of activities relevant to a particular product, service, project, or contract.

It should be clear that development and quality plan procedures applicable to large projects cannot be automatically applied to small projects.

Situations where STANDARD development and quality plan may or may not be required.

1. Projects requiring 15 days – not required

2. Project requiring 50 days where no significant software risk item had been identified – at project leader’s discretion (the freedom to decide what should be done in a particular situation)

3. Small project but complicated that need to be complete within 30 days, in which there is heavy penalty on not being completed on time – obligatory

**Attribute Tradeoffs:**

● Usability vs Functionality

Usability increases no effect on functionality & Functionality increase so usability decrease

● Performance vs Security

Security increases performance decrease &Performance increase so no effect on security

● Usability vs Portability

Portability increases so usability decrease & Usability increase so portability has no effect

**Difference between Inspection and Walkthrough:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Inspection | | Walkthrough |
| 1. | | It is formal. | It is informal. |
| 2. | | Initiated by project team. | Initiated by author. |
| 3. | | A group of relevant persons from different departments participate in the inspection. | Usually team members of the same project take participation in the walkthrough. Author himself acts walkthrough leader. |
| 4. | | Checklist is used to find faults. | No checklist is used in the walkthrough. |
| 5. | | Inspection processes includes overview, preparation, inspection, and rework and follow up. | Walkthrough process includes overview, little or no preparation, little or no preparation examination (actual walkthrough meeting), and rework and follow up. |

**When to use the V-model:**

V-Model is used for small projects where project requirements are clear.

Simple and easy to understand and use.

This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product.

**When to not use the V-model:**

* For large projects where requirements are not clearly defined and fixed.
* When ample technical resources are not available with needed technical expertise.
* High confidence of customer is required for choosing the V-Shaped model approach.
* Where prototypes are required Since, no prototypes are produced in V-model, there is a very high risk involved in meeting customer expectations.

**All Web services are APIs but all APIs are not web services**

* Web services are API that are online, every web service is API. If API is not deployed yet, then it is not a web service.
* Web services might not perform all the operations that an API would perform
* API supports traditional CRUD
* Not all web services support CRUD

**Development and Quality plan for internal projects**

**Benefits of preparing development and quality plans for internal projects to the software development department:**

1. Avoiding budget overruns.

2. Avoiding damage to other projects due to the delays in releasing professional resources involved in internal project.

3. Avoiding loss of market status caused by delayed completion of external projects triggered by late

completion of internal projects.

**Benefits of preparing development and quality plans for internal projects to the internal customer:**

1. Smaller deviations from planned completion dates and smaller budget overruns.

2. Better control of the development process including earlier detection of potential delays.

3. Fewer internal delay damages.

**Why do we need quality models when we have various software process models? What**

**is the significance of Software Quality Models?**

Software Quality Models are a standardized way of measuring a software product. With the increasing trend in software industry, new applications are planned and developed everyday. This eventually gives rise to the need for reassuring that the product so built meets at least the expected standards.

Whereas software process model is to provide guidance for controlling and coordinating the tasks to achieve the end product and objectives as effectively as possible.

**Manual Testing Vs Automated Testing: Usage**

* Manual testing is suitable when the test cases are run once or twice. Therefore there is no frequent repetition of test cases.
* The manual testing comes to rescue while working with poorly written specification documentation and there is a shortage of time.
* Manual testing helps to find out the user-friendliness of an application. It involves human observation to find out any glitches.
* Manual testing helps in improving the customer experience.
* This type of testing is performed without any planning and documentation.

Automated testing is suitable when the test cases need to run repeatedly for a long duration of time.

Automated testing doesn’t ensure user-friendliness of the application

Automated testing is apt when there are frequent code changes which require frequent testing. This testing is conducted to ensure that the new code changes that are implemented don’t affect the existing functionalities.

Automated testing is preferred for load testing which helps to determine a system’s performance under real-life load conditions. This testing finds out how the system behaves when multiple users access it simultaneously.

Is it important for QA to know coding?

QA verifies the software only after the development phase. QA role is not as technical as a developer and **may not require coding**

Software tester into **manual testing does not require coding skills** but needs common sense and clear understanding of how the application works.