SE L2T05

1. Briefly describe possible standards that might be used for these real-world

products:

Soap

Aeroplane

Curry

Possible standards that might be used for the following real-world products are:

1. Soap:

- Quality standards: These standards would ensure that the soap is free from any impurities, has a specific pH range, and meets the required cleansing effectiveness.
- Safety standards: These standards would ensure that the soap is safe for human use and does not cause any adverse effects on skin, eyes, or any other parts of the body.
- Environmental standards: These standards would ensure that the soap is manufactured in an eco-friendly manner and does not cause any harm to the environment during production, use, or disposal.

2. Aeroplane:

- Safety standards: These standards would ensure that the aeroplane is safe to fly and complies with various aviation regulations and requirements, such as those set by the CAA - Civil Aviation Authority in the South Africa.
- Performance standards: These standards would ensure that the aeroplane meets specific performance metrics, such as fuel efficiency, speed, altitude, and range.
- Manufacturing standards: These standards would ensure that the aeroplane is manufactured with high-quality materials, meets certain design specifications, and is assembled according to strict guidelines.

3. Curry:

- Food safety standards: These standards would ensure that the curry is safe to eat and free from any harmful bacteria, viruses, or other contaminants that could cause illness.
- Quality standards: These standards would ensure that the curry has a consistent taste, flavor, and texture, and meets certain nutritional requirements.
- Packaging and labeling standards: These standards would ensure that the curry is packaged and labeled correctly, with all necessary information such as ingredients, allergens, and expiration date, clearly stated.

2. Explain why a high-quality software process should lead to high-quality software products. Discuss possible problems with this system of quality management.

   A high-quality software process should lead to high-quality software products because it ensures that the software is developed in a systematic and controlled manner. A good software process should include several key steps, including requirements gathering, design, implementation, testing, and maintenance. Each step should be carefully planned, executed, and reviewed to ensure that the software meets the necessary quality standards.

   One of the main benefits of a high-quality software process is that it reduces the number of defects in the software. By following a structured process and using proven techniques, developers can catch errors early in the development cycle and fix them before they become more serious and costly to fix. This can result in software that is more reliable, easier to maintain, and has fewer bugs.

   Another benefit of a high-quality software process is that it can help ensure that the software meets the needs of its users. By involving customers in the development process, software developers can get feedback on the software's functionality, usability, and performance. This feedback can then be used to improve the software and ensure that it meets the user's requirements.

   However, there are possible problems with this system of quality management. One problem is that it can be difficult to define and measure quality. Quality is a subjective term, and what one person considers high quality may not be the same as what another person considers high quality. This can make it challenging to set clear quality goals and metrics that everyone agrees on.

   Another problem is that implementing a high-quality software process can be time-consuming and expensive. It requires a significant investment in training, tools, and processes, which may not be feasible for all companies. Additionally, some companies may be resistant to change and may struggle to adopt new processes and practices.

Finally, there is the potential for human error and bias in the inspection process. Even with a well-defined process and trained inspectors, there is always the possibility that defects may be missed or that inspectors may be influenced by their own biases or preferences.

Despite these challenges, a high-quality software process can ultimately lead to better software and improved customer satisfaction. It requires a commitment to continuous improvement and a willingness to invest the necessary resources to achieve quality goals.

3. Explain why program inspections are an effective technique for discovering errors in a program. What types of errors are unlikely to be discovered through inspections?

   Program inspections or peer code reviews are an effective technique for discovering errors in a program because they allow multiple people to review the code, identify errors, and suggest improvements. This technique can help to identify errors early in the development process and reduce the likelihood of errors appearing in later stages or after the program is released.

   There are several reasons why program inspections are effective:

   1. Inspections are conducted by other developers or peers, who have a different perspective on the code than the original author. This helps to identify errors that may have been overlooked by the original author.

   2. Inspections can help to identify errors in the design and logic of the program. This can help to prevent errors that may not be apparent until later in the development process.

   3. Inspections can identify errors that are difficult to detect through testing, such as errors in error handling, boundary conditions, and performance issues.

   However, there are some types of errors that are unlikely to be discovered through inspections.

1. Runtime errors that occur only under specific conditions may not be detected during inspections.
2. Security vulnerabilities may not be detected during inspections unless the reviewers have specific expertise in security.
3. Code that is difficult to understand or requires extensive knowledge of the system may be more difficult to review effectively.

Overall, program inspections are an effective technique for discovering errors in a program and improving its quality. However, inspections should be used in combination with other testing techniques to ensure that all types of errors are detected and corrected.

4. Why is it difficult to validate the relationships between internal product attributes and external attributes?

It can be difficult to validate the relationships between internal product attributes and external attributes because internal attributes are often complex and have indirect effects on external attributes.

Internal attributes, such as the design of a product or the quality of the code in a software system, can affect external attributes, such as usability or reliability. However, there can be multiple factors that contribute to the external attribute, making it challenging to attribute it to a single internal attribute. Additionally, external attributes may be influenced by factors beyond the control of the product, such as user behavior or environmental conditions.

Furthermore, it may be difficult to measure or quantify the internal attributes accurately. For example, it can be challenging to measure the quality of software code objectively, even though it is an essential internal attribute that affects the external attribute of reliability.

Therefore, while it is essential to consider the relationships between internal product attributes and external attributes, it can be challenging to validate these relationships definitively. It requires careful analysis, testing, and validation methods to ensure that the internal attributes of a product are designed and implemented in a way that optimizes the desired external attributes.