Advanced Higher
Computing Science

Testing & Evaluation: All Units

Pupil Notes

Name:_____

TESTING

TYPES OF TESTING

Each version (release) of the software must be fully tested before being released to the customer.

For all units testing includes:

- Integrative Testing
- Usability Testing based on prototypes
- Final Testing
- End User Testing

For SDD only:

- Component testing during the development

For Databases only:

- SQL operations testing(tables & queries)

Integrative Testing

Integrative testing tests:

- integration or interfaces between components
- interactions to different parts of the system such as an operating system, file system and hardware
- Interfaces between systems.

Integration is now a key concept of the Advanced Higher course so testing is expected to be demonstrated within your project.

Usability Testing

In usability testing the testers tests the ease with which the user interfaces can be used.

It tests whether the application is user-friendly or not. Prototypes may be used during this type of testing to ensure the final solution is effective in terms of usability.

Final Testing

A final test of the entire system proves if the program is fit for purpose.

It involves looking back at the specifications and checking the requirements carefully.

Ask such questions as:

- How well does the design function?
- Does the design look good?
- Is the product safe to use?
- How could I have improved on my design?

End User Testing

Takes place at customer's side.

It sends the system/software to users who install it and use it under real-world working conditions.

The goal of this type of testing is to discover any flaws or issues from the user's perspective that you would not want to have in the final version of the application.

Advantages:

- You have the opportunity to get your application into the hands of users prior to releasing it to the general public.
- Users can test your application, and send feedback to you during the testing period.
- Your beta testers can discover issues with your application that you may have not noticed.
- Using the feedback you get from these users, you can fix problems before it is released to the general public.
- Testers will generate excitement about the application.

Component Testing (SDD only)

Component testing (module testing) is a method where testing of each component in an application is done separately.

If an application has 5 components. Testing of each 5 components separately and efficiently is called as component testing.

SQL Testing (DB only)

SQL tables should be checked to ensure that tables have been created successfully and match the design documentation.

Any Queries that have been implemented should undergo test tables with expected results to ensure they are producing the correct results.

EVALUATION

TERMINOLOGY

Once the solution is complete and testing has been finalised it is important to reflect and consider a variety of factors to ensure the clients' meets have been met.

You should consider the following factors during evaluation for all units:

- Fitness for purpose
- Maintainability
    - Perfective
    - Corrective
    - Adaptive
- Robustness

For SDD only:

- Efficiency
- Usability

For DB only:

- Accuracy of Output

For Web only:

- Usability

Fitness for Purpose

You should use the analysis documentation including the end user and functional requirements to ensure your solution has meet the client's needs.

Maintainability

Perfective

At some point after completion the client may wish to add new features to software

This occurs on a system that has been in place and running fine for a while.

Over time, the end user will often find tweaks or minor improvements which could be made to improve the way the system works.

These tweaks are not sufficient to prompt the development of a new system.

Corrective

This involves fixing errors not found during testing. This may happen for a variety of reasons and is normal during the development of all software.

- paid for by the software company
- Obscure errors may only be encountered as users interact with the system day after day.
- Part of the system should include a way for users to report these problems.
- Error logs should be included in the system so maintenance staff can spot problems even if they are not reported.

Adaptive

Making the software work on a different OS or hardware than the one it was designed for.

This type of maintenance often occurs as a result of external influences or strategic changes within a company.

The system is being adapted to remain up to date.

Robustness

Solutions are deemed to be robust if it can cope with unexpected or incorrect input.

Solutions are likely to encounter incorrect or unexpected data:

- because of user error
- when they receive incorrect data from external files

Input validation should include clear error messages so that users are able to understand what they need to do to enter acceptable data.

Efficiency (SDD only)

Code efficiency is a broad term used to depict the reliability, speed and programming methodology used in developing codes for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. It is the key element in ensuring high performance. The goal of code efficiency is to reduce resource consumption and completion time as much as possible with minimum risk to the business or operating environment.

When designing a solution iteration, data structures and all other programming constructs relate to efficiency of code.

Usability (SDD only)

Usability is the ease of use and learnability of a solution. In software development, usability is the degree to which a software can be used by specified consumers to achieve specific objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

Accuracy of Output (SQL only)

SQL tables should be checked to ensure that tables have been created successfully and match the design documentation.

Ensuring that all testing has been completed to a satisfactory standard and the results are as expected.

Usability (Web only)

Usability is the ease of use and learnability of a solution. In web development, usability is the degree to which a website can be used by specified consumers to achieve specific objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.