 Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022 **APEC**

Topic covered

- Development testing
- Test-driven development
- Release testing
- User testing

SOFTWARE TESTING 5/12/2022 2

Program Testing

- Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use.
- When we test software, we execute a program using artificial data.
- Check the results of the test run for errors, anomalies or information about the program's non-functional attributes.
- Can reveal the presence of errors NOT their absence.
- Testing is part of a more general verification and validation process, which also includes static validation techniques.

Validation and defect testing

- The first goal leads to **validation testing**
 - Expect the system to perform correctly using a given set of test cases that reflect the system's expected use and behaviour.
- The second goal leads to **defect testing**
 - The test cases are designed to expose defects and reveal errors. The test cases in defect testing can be deliberately obscure and need not reflect how the system is normally used.

Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC


5/12/2022

6

SOFTWARE TESTING

Program testing goals

- To demonstrate to the developer and the customer that the software meets its requirements.
- For custom software, this means that there should be at least one test for every requirement in the requirements document. For generic software products, it means that there should be tests for all of the system features, plus combinations of these features, that will be incorporated in the product release.
- To discover situations in which the behaviour of the software is incorrect, undesirable or does not conform to its specification.
- Defect testing is concerned with rooting out undesirable system behaviour such as system crashes, unwanted interactions with other systems, incorrect computations and data corruption.



Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC


5/12/2022

6

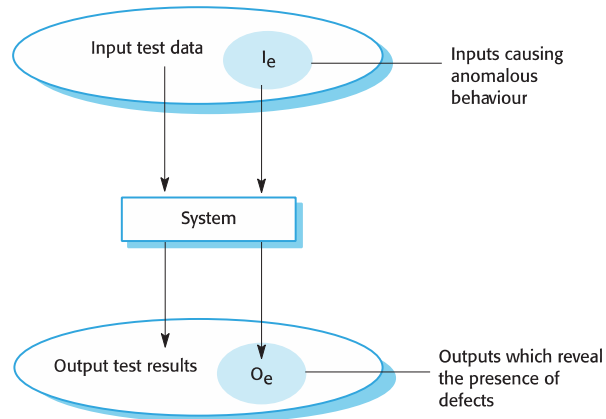
SOFTWARE TESTING

Testing process goals

- **Validation testing**
 - To demonstrate to the developer and the system customer that the software meets its requirements
 - A successful test shows that the system operates as intended.
- **Defect testing**
 - To discover faults or defects in the software where its behaviour is incorrect or not in conformance with its specification
 - A successful test is a test that makes the system perform incorrectly and so exposes a defect in the system.



An input-output model of program testing



Verification vs validation

Verification:

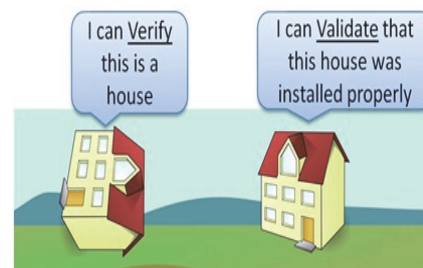
"Are we building the product right"?


The software should conform to its specification.

Validation:

"Are we building the right product"?

The software should do what the user really requires.



 APEC
Asia-Pacific
Economic Cooperation


Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022 **APEC**

V & V confidence

Aim of V & V is to establish confidence that the system is 'fit for purpose'.
Depends on system's purpose, user expectations and marketing environment

- **Software purpose**
 - The level of confidence depends on how critical the software is to an organisation.
- **User expectations**
 - Users may have low expectations of certain kinds of software.
- **Marketing environment**
 - Getting a product to market early may be more important than finding defects in the program.

SOFTWARE TESTING 5/12/2022 9

 APEC
Asia-Pacific
Economic Cooperation

Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022 **APEC**

Inspections and testing

- **Software inspections**
 - Concerned with analysis of the static system representations or artefacts to discover problems (static verification)
 - Maybe supplemented by tool-based document and code analysis.
- **Software testing**
 - Concerned with exercising and observing software product behaviour (dynamic verification)
 - The system is executed with test data and its operational behaviour is observed.

SOFTWARE TESTING 5/12/2022 10

Software inspections

- These involve people examining the source representation with the aim of discovering anomalies and defects.
- Inspections do not require execution of a system so may be used before implementation.
- They may be applied to any representation of the system (requirements, design, configuration data, test data, etc.).
- They have been shown to be an effective technique for discovering program errors.

Advantages of inspections

- During testing, errors can mask (or hide) other errors. Because inspection is a static process, we don't have to be concerned with interactions between errors.
- Incomplete versions of a system can be inspected without additional costs.
- As well as searching for program defects, an inspection can also consider broader quality attributes of a program, such as compliance with standards, portability and maintainability.



Inspections and testing

- Inspections and testing are complementary and not opposing verification techniques.
- Both should be used during the V & V process.
- Inspections can check conformance with a specification but not conformance with the customer's real requirements.
- Inspections cannot check non-functional characteristics such as performance, usability, etc.
- May refers to software metrology standards, guidelines and ISTQB (International Software Testing Qualifications Board).



Stages of testing

- **Development testing**, where the system is tested during development to discover bugs and defects.
- **Release testing**, where a separate testing team test a complete version of the system before it is released to users.
- **User testing**, where users or potential users of a system test the system in their own environment.



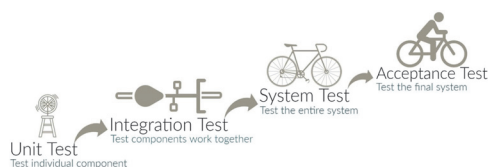
Development testing

Development testing includes all testing activities that are carried out by the team developing the system.

- **Unit testing**, where individual program units or object classes are tested. Unit testing should focus on testing the functionality of objects or methods.
- **Component testing**, where several individual units are integrated to create composite components. Component testing should focus on testing component interfaces.
- **System testing**, where some or all of the components in a system are integrated and the system is tested as a whole. System testing should focus on testing component interactions.



Unit testing



- Unit testing is the process of testing individual components in isolation.
- It is a defect testing process.
- Units may be:
 - ✓ Individual functions or methods within an object
 - ✓ Object classes with several attributes and methods
 - ✓ Composite components with defined interfaces used to access their functionality.

Testing strategies

- Partition testing, where you identify groups of inputs that have common characteristics and should be processed in the same way.
 - ❖ You should choose tests from within each of these groups.
- Guideline-based testing, where you use testing guidelines to choose test cases.
 - ❖ These guidelines reflect the previous experience of the kinds of errors that programmers often make when developing components.



Partition testing

Input data and output results often fall into different classes where all members of a class are related.

Each of these classes is an equivalence partition or domain where the program behaves in an equivalent way for each class member.

Test cases should be chosen from each partition.



Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC

APEC
Asia-Pacific
Economic Cooperation

Equivalence partitioning

Possible inputs Correct outputs Possible outputs

Input equivalence partitions Output partitions

System

SOFTWARE TESTING 5/12/2022 19

Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC

APEC
Asia-Pacific
Economic Cooperation

General testing guidelines

- Choose inputs that force the system to generate all error messages
- Design inputs that cause input buffers to overflow
- Repeat the same input or series of inputs numerous times
- Force invalid outputs to be generated
- Force computation results to be too large or too small.

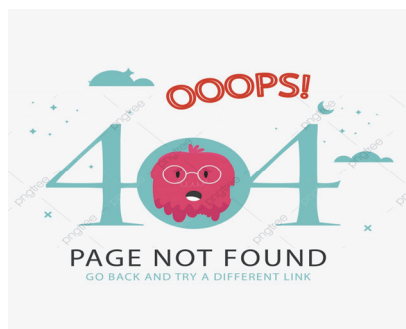
SOFTWARE TESTING 5/12/2022 20

Component testing


- Software components are often composite components that are made up of several interacting objects.
- We access the functionality of these objects through the defined component interface.
- Testing composite components should therefore focus on showing that the component interface behaves according to its specification.
- We can assume that unit tests on the individual objects within the component have been completed.



Interface testing




- Objectives are to detect faults due to interface errors or invalid assumptions about interfaces.
- Interface types
 - Parameter interfaces Data is passed from one method or procedure to another.
 - Shared memory interfaces Block of memory is shared between procedures or functions.
 - Procedural interfaces Sub-system encapsulates a set of procedures to be called by other sub-systems.
 - Message passing interfaces Sub-systems request services from other sub-systems



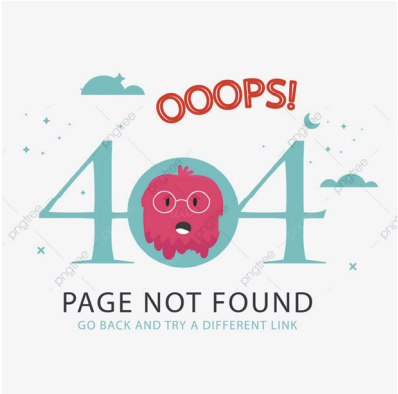
Asia-Pacific
Economic Cooperation

Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022



Interface errors


- Interface misuse
 - A calling component calls another component and makes an error in its use of its interface e.g. parameters in the wrong order.
- Interface misunderstanding
 - A calling component embeds assumptions about the behaviour of the called component which are incorrect.
- Timing errors
 - The called and the calling component operate at different speeds and out-of-date information is accessed.



SOFTWARE TESTING


5/12/2022

23



Asia-Pacific
Economic Cooperation

Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022



Interface testing guidelines

- Design tests so that parameters to a called procedure are at the extreme ends of their ranges.
- Always test pointer parameters with null pointers.
- Design tests which cause the component to fail.
- Use stress testing in message passing systems.
- In shared memory systems, vary the order in which components are activated.

SOFTWARE TESTING

5/12/2022

24

System testing

- System testing during development involves integrating components to create a version of the system and then testing the integrated system.
- The focus of system testing is testing the interactions between components.
- System testing checks that components are compatible, interact correctly and transfer the right data at the right time across their interfaces.
- System testing tests the emergent behaviour of a system.



System and component testing



- During system testing, reusable components that have been separately developed and off-the-shelf systems may be integrated with newly developed components. The complete system is then tested.
- Components developed by different team members or sub-teams may be integrated at this stage. System testing is a collective rather than an individual process.
 - In some companies, system testing may involve a separate testing team with no involvement from designers and programmers.

Testing policies

- Exhaustive system testing is impossible so testing policies that define the required system test coverage may be developed.
- Examples of testing policies:
 - All system functions that are accessed through menus should be tested.
 - Combinations of functions (e.g. text formatting) that are accessed through the same menu must be tested.
 - Where user input is provided, all functions must be tested with both correct and incorrect input.



Release testing

Release testing



- Release testing is the process of testing a particular release of a system that is intended for use outside of the development team.
- The primary goal of the release testing process is to convince the supplier of the system that it is good enough for use.
 - Release testing, therefore, has to show that the system delivers its specified functionality, performance and dependability and that it does not fail during normal use.
- Release testing is usually a black-box testing process where tests are only derived from the system specification.

Release testing and system testing



- Release testing is a form of system testing.
- Important differences:
 - A separate team that has not been involved in the system development, should be responsible for release testing.
 - System testing by the development team should focus on discovering bugs in the system (defect testing). The objective of release testing is to check that the system meets its requirements and is good enough for external use (validation testing).

Performance testing

- Part of release testing may involve testing the emergent properties of a system, such as performance and reliability.
- Tests should reflect the profile of use of the system.
- Performance tests usually involve planning a series of tests where the load is steadily increased until the system performance becomes unacceptable.
- Stress testing is a form of performance testing where the system is deliberately overloaded to test its failure behaviour.




User testing

Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC

APEC
Asia-Pacific
Economic Cooperation

User testing



- User or customer testing is a stage in the testing process in which users or customers provide input and advice on system testing.
- User testing is essential, even when comprehensive system and release testing have been carried out.
 - The reason for this is that influences from the user's working environment have a major effect on the reliability, performance, usability and robustness of a system. These cannot be replicated in a testing environment.

SOFTWARE TESTING

5/12/2022 33


Capacity Building Workshop On Understanding Conformity Requirements For Software Controlled Weight And Measuring Instruments For Sustainable Trade 2022

APEC

APEC
Asia-Pacific
Economic Cooperation

Types of user testing

- Alpha testing
 - Users of the software work with the development team to test the software at the developer's site.
- Beta testing
 - A release of the software is made available to users to allow them to experiment and to raise problems that they discover with the system developers.
- Acceptance testing
 - Customers test a system to decide whether or not it is ready to be accepted from the system developers and deployed in the customer environment. Primarily for custom systems.



SOFTWARE TESTING

5/12/2022 34

Agile methods and acceptance testing



In agile methods, the user/customer is part of the development team and is responsible for making decisions on the acceptability of the system.

Tests are defined by the user/customer and are integrated with other tests in that they are run automatically when changes are made.


There is no separate acceptance testing process.

Main problem here is whether or not the embedded user is 'typical' and can represent the interests of all system stakeholders.

Conclusion


- Testing can only show the presence of errors in a program. It cannot demonstrate that there are no remaining faults.
- Development testing is the responsibility of the software development team. A separate team should be responsible for testing a system before it is released to customers.
- Development testing includes unit testing, in which you test individual objects and methods, component testing in which you test related groups of objects and system testing, in which you test partial or complete systems.



 APEC
Asia-Pacific
Economic Cooperation


Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022 **APEC**

Conclusion



- When testing software, we will try to 'break' the software by using experience and guidelines to choose types of test cases that have been effective in discovering defects in other systems.
- Wherever possible, we should write automated tests. The tests are embedded in a program that can be run every time a change is made to a system.
- Acceptance testing is a user testing process where the aim is to decide if the software is good enough to be deployed and used in its operational environment.

SOFTWARE TESTING 5/12/2022 37

 APEC
Asia-Pacific
Economic Cooperation

Capacity Building Workshop On Understanding Conformity
Requirements For Software Controlled Weight And Measuring
Instruments For Sustainable Trade 2022 **APEC**

THANK YOU

Q&A

SOFTWARE TESTING 5/12/2022 38