Chapter 8 – Software Testing

Topics covered

✧ Development testing

✧ Test-driven development

✧ Release testing

✧ User testing

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 you test software, you execute a program using artificial data.

✧ You 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.

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 behavior of the software is incorrect, undesirable or does not conform to its specification.

- Defect testing is concerned with rooting out undesirable system behavior such as system crashes, unwanted interactions with other systems, incorrect computations and data corruption.

Validation and defect testing

◇ The first goal leads to validation testing

- You expect the system to perform correctly using a given set of test cases that reflect the system's expected use.

◇ The second goal leads to defect testing

- The test cases are designed to expose defects. The test cases in defect testing can be deliberately obscure and need not reflect how the system is normally used

  → test case dgn data yang sengaja dirancang tdk valid utk mengecek reaksi sistem
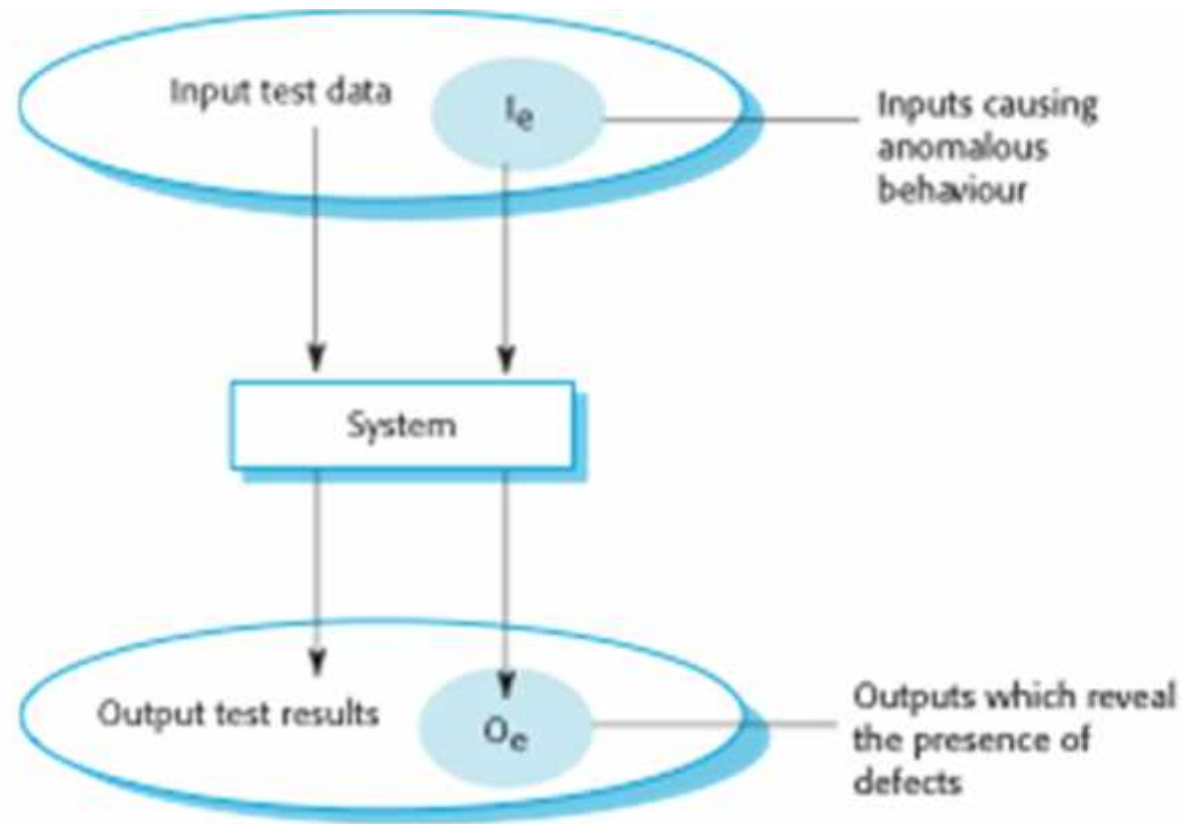
Testing process goals

◇ **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 behavior 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

Input test data — I_e — Inputs causing anomalous behaviour

System

Output test results — O_e — Outputs which reveal the presence of defects

Verification vs validation

◆ Verification:

"Are we building the product right".

- Apakah sudah melakukan prosesnya dgn benar?

◆ The software should conform to its specification.

◆ Validation:

"Are we building the right product".

◆ The software should do what the user really requires.

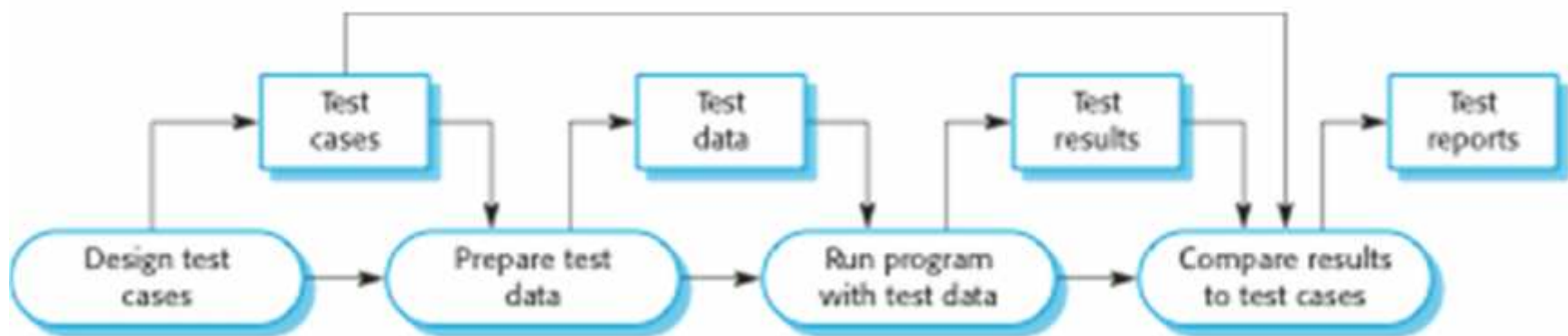- Apakah proses yang benar (yang dibutuhkan) sudah dipenuhi?

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.

A model of the software testing process

A flow diagram. Top row boxes: Test cases → Test data → Test results → Test reports. Bottom row process steps: Design test cases → Prepare test data → Run program with test data → Compare results to test cases.

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.

1. 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.

1a.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 guidelines (sequences)

♦ Test software with sequences which have only a single value.

♦ Use sequences of different sizes in different tests.

♦ Derive tests so that the first, middle and last elements of the sequence are accessed.

♦ Test with sequences of zero length.

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.

Key points

✧ 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.

1b.Component testing

✧ Software components are often composite components that are made up of several interacting objects.

- For example, in the weather station system, the reconfiguration component includes objects that deal with each aspect of the reconfiguration.

✧ You 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.

- You can assume that unit tests on the individual objects within the component have been completed.

1c.System testing

♦ System testing during development involves integrating components to create a version of the system and then testing the integrated system.

♦ The focus in 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 which define the required system test coverage may be developed → pengujian sistem secara lengkap = imposible

◇ 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.

2.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).

Requirements based testing

♢ Requirements-based testing involves examining each requirement and developing a test or tests for it.

♢ MHC-PMS requirements:

- If a patient is known to be allergic to any particular medication, then prescription of that medication shall result in a warning message being issued to the system user.

- If a prescriber chooses to ignore an allergy warning, they shall provide a reason why this has been ignored.

Requirements tests

✧ Set up a patient record with no known allergies. Prescribe medication for allergies that are known to exist. Check that a warning message is not issued by the system.

✧ Set up a patient record with a known allergy. Prescribe the medication to that the patient is allergic to, and check that the warning is issued by the system.

✧ Set up a patient record in which allergies to two or more drugs are recorded. Prescribe both of these drugs separately and check that the correct warning for each drug is issued.

✧ Prescribe two drugs that the patient is allergic to. Check that two warnings are correctly issued.

✧ Prescribe a drug that issues a warning and overrule that warning. Check that the system requires the user to provide information explaining why the warning was overruled.

Features tested by scenario

✧ Authentication by logging on to the system.

✧ Downloading and uploading of specified patient records to a laptop.

✧ Home visit scheduling.

✧ Encryption and decryption of patient records on a mobile device.

✧ Record retrieval and modification.

✧ Links with the drugs database that maintains side-effect information.

✧ The system for call prompting.

A usage scenario for the MHC-PMS

Kate is a nurse who specializes in mental health care. One of her responsibilities is to visit patients at home to check that their treatment is effective and that they are not suffering from medication side -effects.

On a day for home visits, Kate logs into the MHC-PMS and uses it to print her schedule of home visits for that day, along with summary information about the patients to be visited. She requests that the records for these patients be downloaded to her laptop. She is prompted for her key phrase to encrypt the records on the laptop.

One of the patients that she visits is Jim, who is being treated with medication for depression. Jim feels that the medication is helping him but believes that it has the side -effect of keeping him awake at night. Kate looks up Jim's record and is prompted for her key phrase to decrypt the record. She checks the drug prescribed and queries its side effects. Sleeplessness is a known side effect so she notes the problem in Jim's record and suggests that he visits the clinic to have his medication changed. He agrees so Kate enters a prompt to call him when she gets back to the clinic to make an appointment with a physician. She ends the consultation and the system re-encrypts Jim's record.

After, finishing her consultations, Kate returns to the clinic and uploads the records of patients visited to the database. The system generates a call list for Kate of those patients who she has to contact for follow-up information and make clinic appointments.

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

✧ 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.

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.

The acceptance testing process
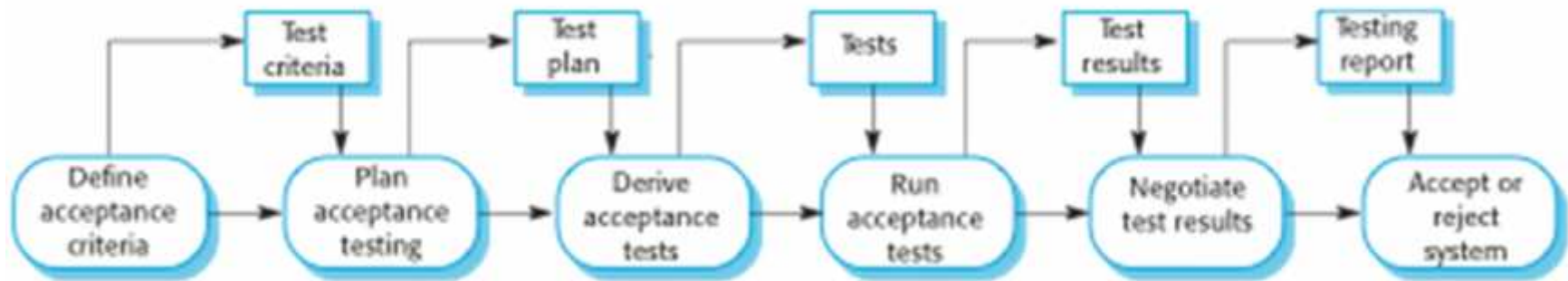
Stages in the acceptance testing process

✧ Define acceptance criteria

✧ Plan acceptance testing

✧ Derive acceptance tests

✧ Run acceptance tests

✧ Negotiate test results

✧ Reject/accept system

Key points

✧ When testing software, you should try to 'break' the software by using experience and guidelines to choose types of test case that have been effective in discovering defects in other systems.

✧ Wherever possible, you should write automated tests. The tests are embedded in a program that can be run every time a change is made to a system.

✧ Test-first development is an approach to development where tests are written before the code to be tested.

✧ Scenario testing involves inventing a typical usage scenario and using this to derive test cases.

✧ 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.