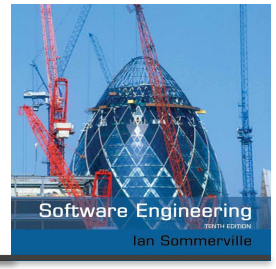
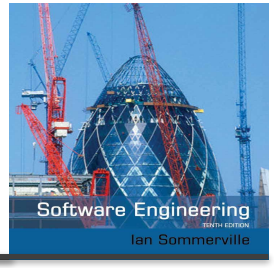
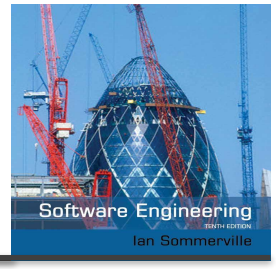


Chapter 8: Software testing



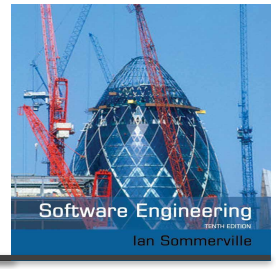
Release testing

Release testing



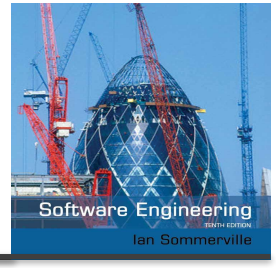
- Release testing is the process of testing a particular release of a system that is intended for use outside of the development team.
- System release -> for customers & users
- Complex projects -> release is for other teams developing the project
- Software product -> release for project management to prepare for sale
- Aim of release testing : to convince the system supplier that it is good enough for use.
  - Release testing 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. System is a black box whose behaviors is observed by giving inputs & observing outputs.

Release testing and system testing



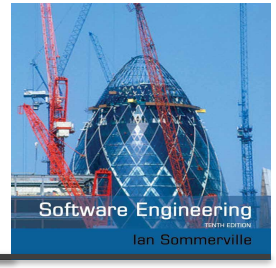
- Release testing is a form of system testing.
- Important differences:
  - Release testing :
    - done by a separate team that has not been involved in the system development.
    - The objective of release testing is to check that the system meets its requirements and is good enough for external use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

Requirements based testing



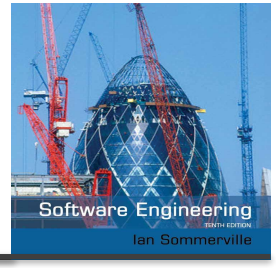
- Best requirement engineering practice is that the requirements should be testable. i.e., every requirement must be mentioned clearly so that a test can be defined for that.
- **Requirements-based testing:** Examine each requirement and then develop test(s) for it.
- Requirement based testing is validation testing (every req. is implemented correctly)

Requirements based testing : case study 1



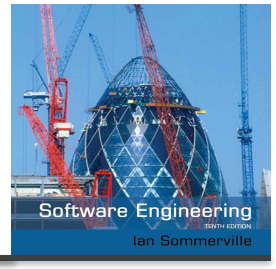
- Medicine recommendation system requirements:
  - **Task: prescribe the medicine to patient**
  - If a patient is known to be allergic to any specific medicine, then prescription of that medication shall result in a warning message being issued to the system user.
  - If a prescriber chooses to ignore an allergy warning, they shall provide a reason why this has been ignored.

Requirements tests



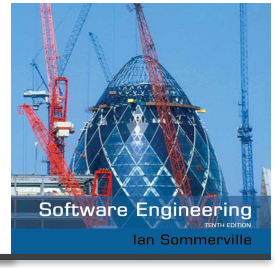
- Set up a patient record with no known allergies. Prescribe medication that exists in stock. Check that a warning message is not issued by the system.
- Set up a patient record with a known allergy. Prescribe the medication to that the patient is allergic to, and check that the warning is issued by 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.

Requirements based testing : case study 2



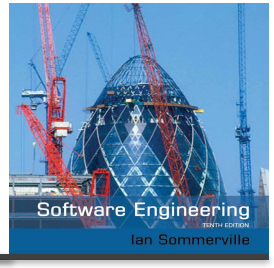
- Dvago is an online medicine store that supplies medicines to the patients. Define testcases for the requirements given below:
  - Customer search for the medicine availability. If the medicine is available, then the patient needs to upload the prescription image before adding it to cart. If medicine is in the prescription list, then the stock is reduced, and medicine is shipped. Otherwise, an error is prompted item sold out.

Scenario testing



- Requirement engineering process done in the form of user stories.
- User stories must be realistic and easy to understand by any system stakeholders.

A usage scenario for the Mentcare system



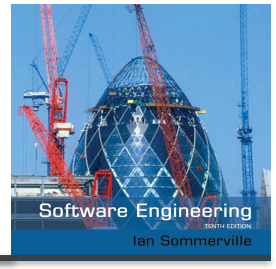
George is a nurse who specializes in mental healthcare. One of his responsibilities is to visit patients at home to check that their treatment is effective and that they are not suffering from medication side effects.

On a day for home visits, George logs into the Mentcare system and uses it to print his schedule of home visits for that day, along with summary information about the patients to be visited. He requests that the records for these patients be downloaded to his laptop. He is prompted for his key phrase to encrypt the records on the laptop.

One of the patients that he 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. George looks up Jim's record and is prompted for his key phrase to decrypt the record. He checks the drug prescribed and queries its side effects. Sleeplessness is a known side effect so he notes the problem in Jim's record and suggests that he visits the clinic to have his medication changed. Jim agrees so George enters a prompt to call him when he gets back to the clinic to make an appointment with a physician. George ends the consultation and the system re-encrypts Jim's record.

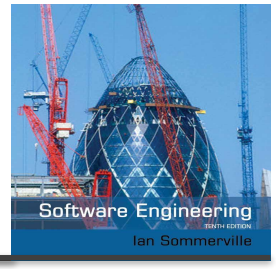
After, finishing his consultations, George returns to the clinic and uploads the records of patients visited to the database. The system generates a call list for George of those patients who He has to contact for follow-up information and make clinic appointments.

Features tested by scenario



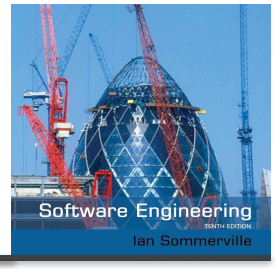
- Authentication by logging on to the system.
- Downloading and uploading of specified patient records to a laptop.
- Home visit scheduling.
- Encryption and decryption of patient records on a mobile device.
- Record retrieval and modification.
- Links with the drugs database that maintains side-effect information.
- The system for call prompting.
- The release tester will check for every possible state like false inputs, performance etc.
- Test for each requirement in scenario and test for their combination as well.

Performance testing

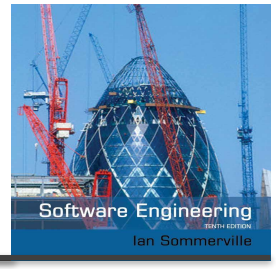


- It's a part of release testing.
- After system integration, its needed to test system for properties like reliability, performance etc.
- Check whether system can process intended load.
- Tests should reflect the profile of use of the system.
- Performance tests usually involve planning a series of 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 behavior.

Performance testing

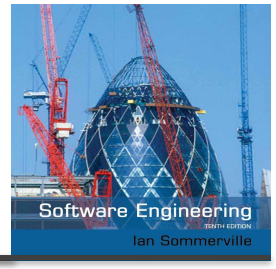


- Let's say system can process the upto 300 transactions at a time, start with lesser than 300 and deliberately increase the load until it reaches beyond the designed threshold and system fails.
- Stress testing is helpful for:
  - System failure must depict a “fail-soft” attitude i.e., it doesn't involve data corruption or loss of user service.
  - Show all the defects that occur when system is operating at maximum load.
- Mostly, Stress testing is done for distributed environments. **System degradation starts as soon as the threshold limit exceeds**. So, tests must be conducted in order to identify the point that will highlight that point and reject any further transactions.



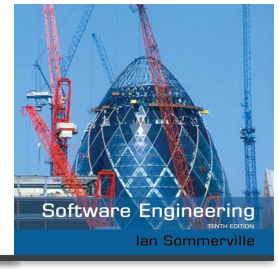
User testing

User testing



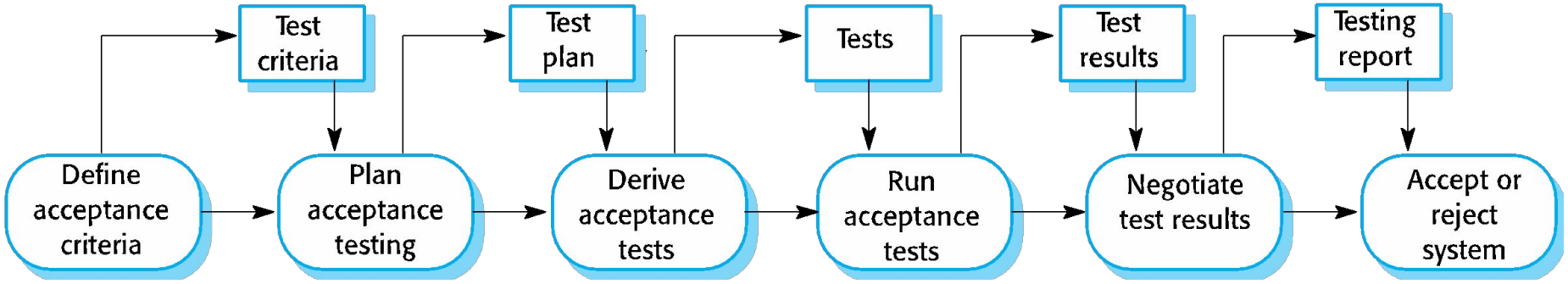
- User or customer testing is a stage in the testing process in which users or customers provide input and test the system for outputs.
- Process of formally testing a system provided by the supplier.
- Users experience the new software product and identify whether the system is doing the intended work, check for interface and defects.
- **User testing is essential**, even when comprehensive system and release testing have been carried out.
  - As 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

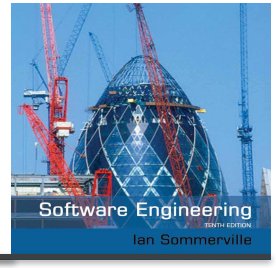


- Alpha testing
  - Users of the software work closely with the development team to test the software at the developer's site.
  - Experienced users help in requirements understanding and hence learn latest features beforehand
  - Provide info to developers that will help[them build an effective interface design.
- 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.
  - Done by early adopters of the system.
- Acceptance testing
  - Customers test a system to decide whether it is ready to be accepted from the system developers and deployed in the customer environment.

The acceptance testing process

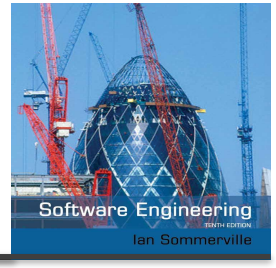


Stages in the acceptance testing process



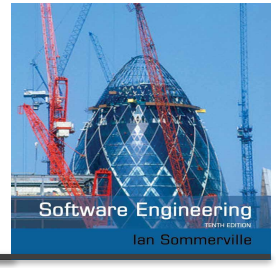
- Define acceptance criteria :
  - uncertain requirements
  - Part of system contract by user
- Plan acceptance testing
  - Decide resource, time, budget
  - Define risks associated with system & develop mitigation techniques
- Derive acceptance tests
  - Test for functional / non-functional requirements
  - Complete coverage of system requirements.

Stages in the acceptance testing process



- Run acceptance tests
  - Agreed acceptance tests executed on system
  - Done in actual working environment
- Negotiate test results
  - Hardly impossible that all test are passed & system is error free
  - If problems, occur the developer and customer negotiate for system acceptance or revision.
- Reject/accept system
  - If system is not up to mark, the revision is required
  - If not, then its accepted

Agile methods and acceptance testing



- Alpha testers ~ 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 issue is to find such user that have sound knowledge of system.