OBJECT ORIENTED ANALYSIS AND DESIGN

UNIT-V: Software Quality Assurance-Quality Assurance Tests-Testing strategies-Test Plan-Usability Testing-User Satisfaction Test.

TEXT BOOK: Ali Bahrami, “Object Oriented Systems Development”, TATA McGraw-Hill Edition, 2008.

PREPARED BY
Dr. D. Devakumari

Content

- Software Quality Assurance
- Quality Assurance Tests
- Testing strategies
- Test Plan-Usability Testing
- User Satisfaction Test.

Quality Assurance Tests

- Debugging is the process of finding out where something went wrong and correcting the code to eliminate the errors or bugs that cause unexpected results.
- By selecting appropriate testing strategies and a sound test plan, you can locate the errors in yours system and fix them using reality available debugging tools.
- Kinds of errors you might encounter when you run your program:
  - Language (syntax) errors result from incorrectly constructed code, Such as an incorrectly typed keyword or some necessary punctuation omitted.
  - Run-time errors occur and are detected as the program is running, when a statement attempts an operation that is impossible to carry out. For example, if the program tries to access a non existent object (say, a file), a run-time error will occur.
  - Logic errors occur when code does not perform the way you intended.
- Quality assurance testing can be divided into two major categories: error-based testing and scenario _ based testing.
- Error-based testing techniques search a given class method for particular clues of interests, then describe how these clues should be tested.
- Scenario-based testing, also called usage-based testing, concentrates on what the user does, not what the product does.

Testing strategies

- Black Box Testing
- White Box Testing
- Top - Down Testing
- Bottom - Up Testing

Black Box Testing

- In a black box, the test item is treated as "black", since its logic is unknown; all that is known is what goes in and what comes out, or the input and output.
- In black box testing, you try various inputs and examine the resulting output.

White Box Testing

- White box testing assumes that the Specific logic is important and must be tested to guarantee the system's proper functioning. The main use of the white box is in error-based testing.
- one form of white box testing, called path testing, Makes certain that each path in a object's method is executed at least once during testing.
- ***Statement testing coverage***:- The main idea of Statement testing coverage is to test every statement in the object's method by executing it at least once.
- ***Branch testing coverage***: - The main idea behind branch testing coverage is to perform enough tests to ensure that every branch alternative has been executed at least once under some test.

Top- Down Testing

- Top-Down testing assumes that the main logic or object interactions and systems messages of the application need more testing than an individual object's methods or supporting logic.
- A top-down strategy can detect the serious design flaws early in the implementation.

Bottom - Up Testing

- Bottom -up testing starts with the details of the system and proceeds to higher levels by a progressive aggregation of details until they collectively fit the requirements for the system.

FIGURE 13–1
The black box is an imaginary box that hides its internal workings.

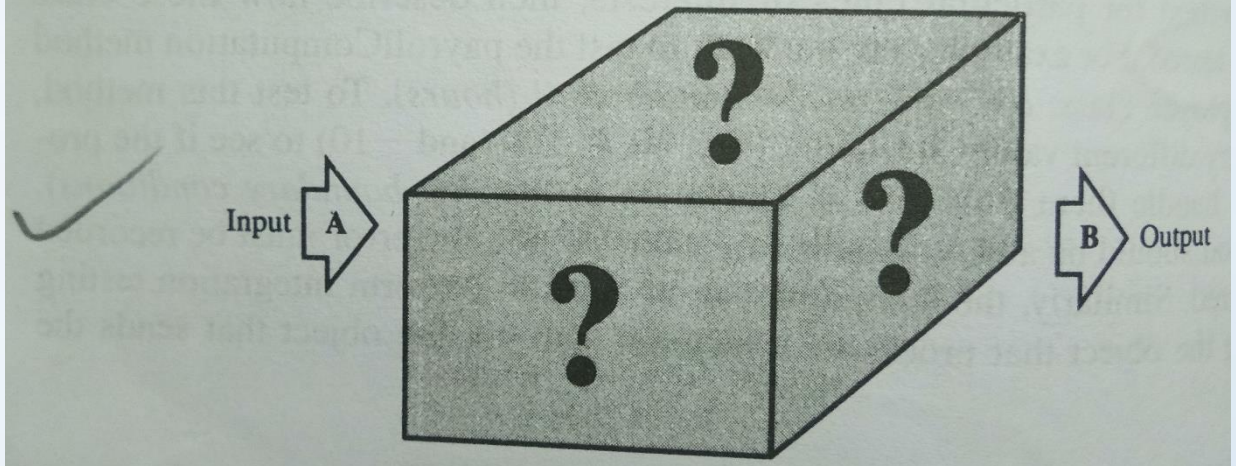
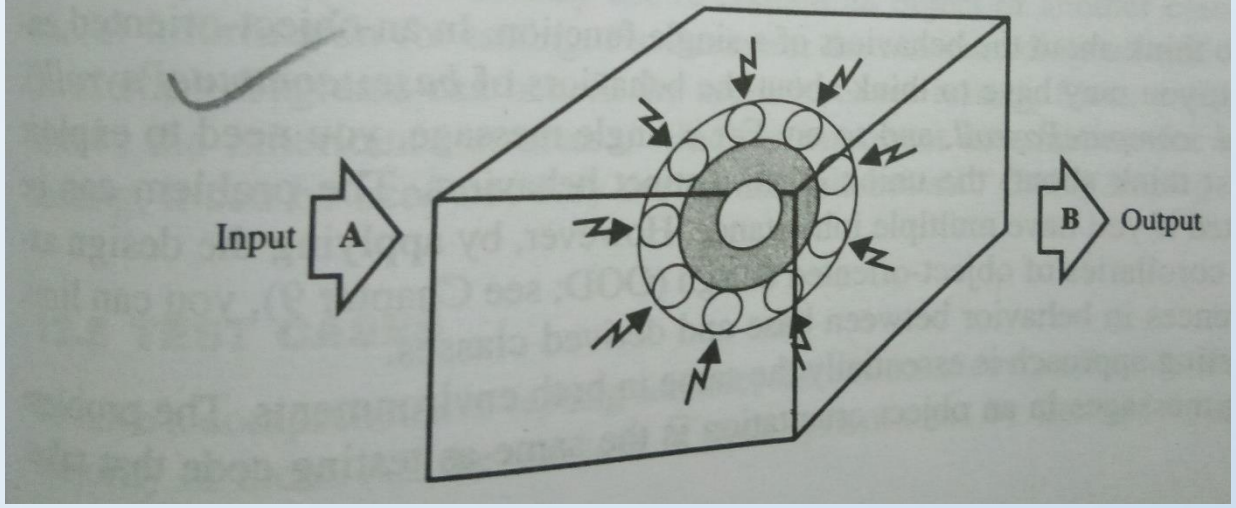


FIGURE 13–2
In a white-box testing strategy, the internal workings are known.



Test Plan

- A test plan is developed to detect and identify potential programs before delivering the software to its users.
- The following steps are needed to create a test plan: -

I. Objectives of the test.

2. Development of a test case.

3. Test analysis.

Guidelines for Developing Test Plans.

- The following guidelines
- Try to include as much detail as possible about the tests.
- The test plan should contain a Schedule and a list of required resources.
- Document every type of test you plan to complete.
- A configuration control system provides a way of tracking the changes to the code.
- A well- thought-out design tends to produce better code and result in more complete testing.
- At the end of each month or as you reach each milestone, take time to complete routine updates.

Usability Testing

- Defines usability as the effectiveness, efficiency, and satisfaction with which a specified set of users can achieve a specified set of tasks in particular environments. The is definition requires: -
  - Defining tasks. What are the tasks?
  - Defining Users. Who are the Users?
  - A means for measuring effectiveness, efficiency, and satisfaction. How do we measure usability?
- Usability testing measures the ease of use as well as the degree of comfort and Satisfaction users have with the software.
- products with poor usability can be difficult to learn, complicated to operate, and Misused or not used at all.
- Usability is one of the most crucial factors in the design and development of a product, especially the user interface.
- Therefore, usability testing must be a key part of the UI design process.
- Usability test cases begin with the identification of use cases that can specify the target audience, tasks, and test goals. [Fig 14.2]

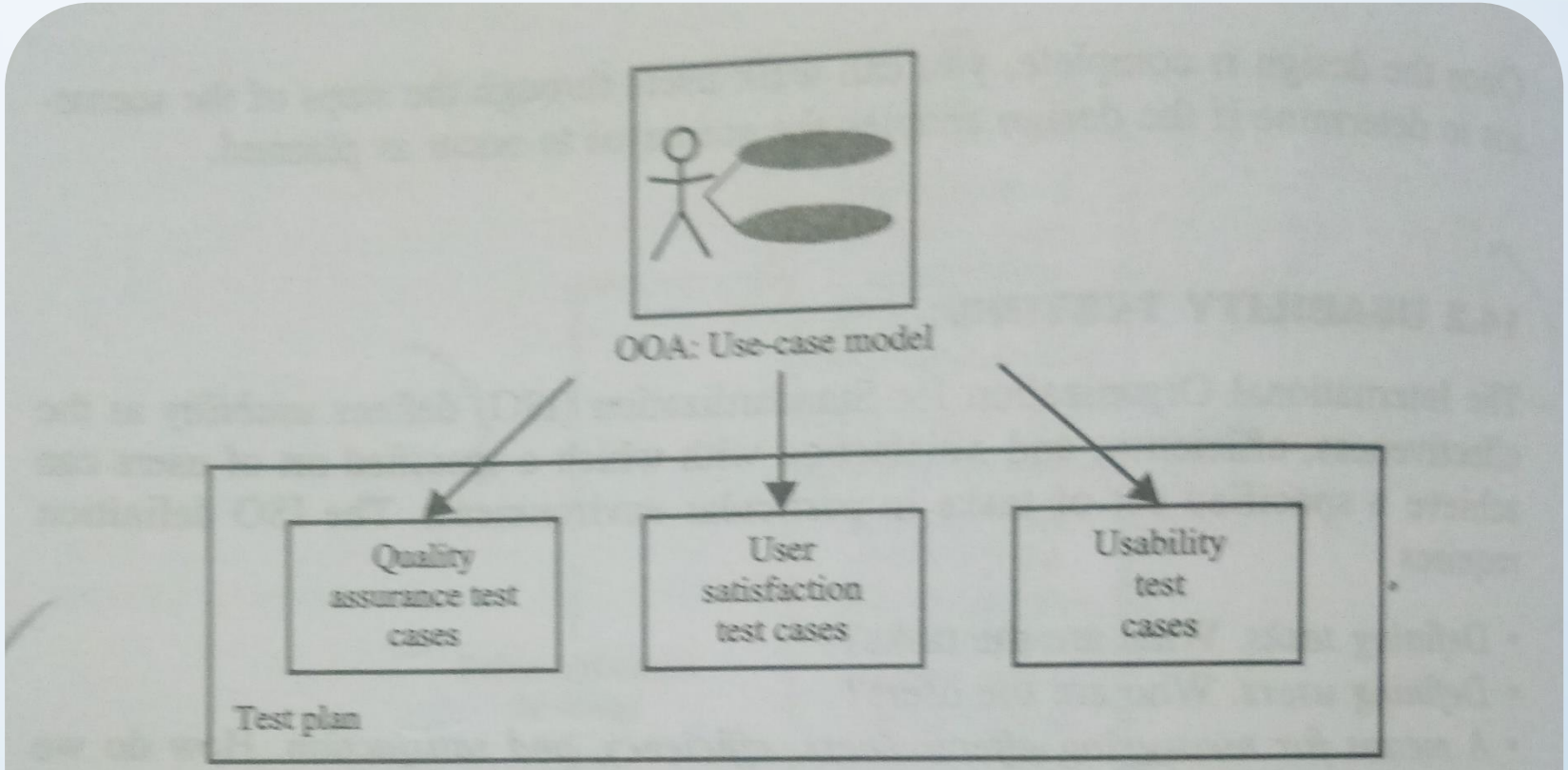


FIGURE 14–2
The use cases identified during analysis can be used in testing the design. Once the design is complete, walk users through the steps of the scenarios to determine if the design enables the scenarios to occur as planned.

Guidelines for Developing Usability Testing:-

- The usability testing should include all of a software's components.
- Usability testing need not be very expensive or elaborate such as including trained specialists working in a
- Soundproof lab with one-way mirrors and sophisticated recording equipment. Even the small investment of tape recorder, Stopwatch, and notepad in an office or conference room can produce excellent results.
- Similarly, all tests need not involve many subjects.
- More typically, quick, iterative tests with a small, well-targeted sample of b to 10 participants can identify 80-90 percent of most design problems.
- Consider the user's experience as part of your software usability. you can study 80-90 percent of most design problem with as few as three or four users if you target only a single skill level of users, such as novices or intermediate level Users.
- Apply usability testing early and often.

Recording the Usability Test: -

- When conducting the usability test, provide an environment comparable to the target setting;
- The users must have reasonable time to try to work through any difficult situation they encounter.
- As the participants work, record the time they take to perform a task as well as any problems they encounter.
- You may want to follow up the session with the user satisfaction test (more on this in the next section) and a questionnaire that asks the participants to evaluate the product or tasks they performed.
- Record the test results
- Recorded data also allows more direct comparison among multiple participants.

User Satisfaction Test

- User satisfaction testing is the process of quantifying the usability test with some measurable attributes of the test, such as functionality, cost, or ease of use.
- 95 percent of users should be able to find how to withdraw money from the ATM machine without error and with no formal training,
- To percent of all users should experience the new function as "a clear improvement over the previous One".
- 90 percent of consumers should be able to operate the VCR within 30 minutes.
- Principal objectives of the user satisfaction test:
  - As a communication vehicle between designers, as well as between users and designers.
  - To detect and evaluate changes during the design process.
  - To provide a periodic indication of divergence of opinion about the current design.
  - To enable pinpointing specific area of dissatisfaction for remedy.
  - To provide a clear understanding of just how the completed design is to be evaluated.

Guidelines for Developing a user satisfaction Test:-

- You must work with the users or clients to find out
- What attributes should be included in the test.
- Ask the users to select a limited number (5 to 10) of attributes by which the final product can be evaluated.
- Ease of use, functionality, cost, intuitiveness of user interface, and reliability.

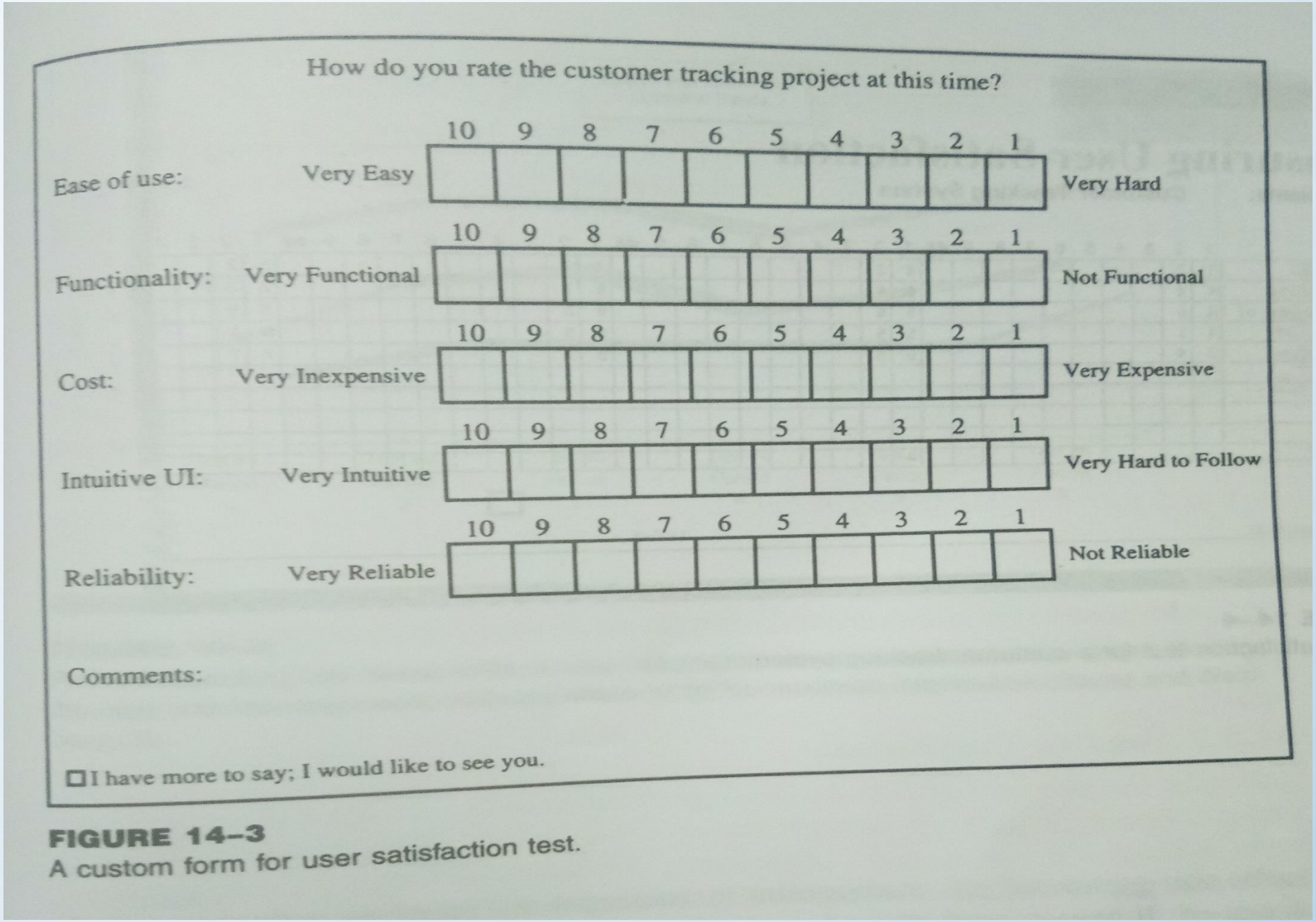


FIGURE 14–3
A custom form for user satisfaction test.