Usability Testing

- Attempts to find human-factor, or usability, problems.
- List of questions you might ask to derive testing considerations:
  1. Has each user interface been tailored to the intelligence, educational background, and environmental pressures of the end user?
  2. Are the outputs of the program meaningful, non insulting to the user, and devoid of computer gibberish?

3. Are the error diagnostics, such as error messages, straightforward, or does the user need a PhD in computer science to comprehend them? For instance, does the program produce such messages as IEK022A OPEN ERROR ON FILE 'SYSIN' ABEND CODE¼102?

→Unhelpful messages such as, “An unknown error has occurred,” or “This program has encountered an error and must be restarted.”

4. Does the total set of user interfaces exhibit considerable conceptual integrity, an underlying consistency, and uniformity of syntax, conventions, semantics, format, style, and abbreviations?

5. Where accuracy is vital, such as in an online banking system, is sufficient redundancy present in the input? For example, such a system should ask for an account number, a customer name, and a personal identification number (PIN) to verify that the proper person is accessing account information.

6. Does the system contain an excessive number of options, or options that are unlikely to be used?

- One trend in modern software is to present to users only those menu choices they are most likely to use, based on software testing and design considerations.

7.Does the system return some type of immediate acknowledgment to all inputs? Where a mouse click is the input,example, the chosen item can change color, or a button object can depress or be presented in a raised format.

- If the user is expected to choose from a list, the selected number should be presented on the screen when the choice is made.

- If the selected action requires some processing time—which is frequently the case when the software is accessing a remote system—then a message should be displayed informing the user of what is going on.
- This level of testing sometimes is referred to as component testing.

8.Is the program easy to use? For example, is the input case-sensitive without making this fact clear to the user? Also, if a program requires navigation through a series of menus or options, is it clear how to return to the main menu? Can the user easily move up or down one level?

9. Is the design conducive to user accuracy? One test would be an analysis of how many errors each user makes during data entry or when choosing program options. Were these errors merely an inconvenience—errors the user was able to correct—or did an incorrect choice or action cause some kind of application failure?

10. Are the user actions easily repeated in later sessions? In other words, is the software design conducive to the user learning how to be more efficient in using the system?

11. Did the user feel confident while navigating the various paths or menu choices? A subjective evaluation might be the user response to using the application. At the end of the session did the user feel stressed by or satisfied with the outcome?

- Would the user be likely to choose this system for his or her own use, or recommend it to someone else?

What kind of testing is a usability test?
(whitebox? Blackbox?)

Usability testing process

Assume usability test for a customer tracking application. Some of the processes are:

- Locate an individual customer record and modify it.
- Locate a company record and modify it.
- Create a new company record.
- Delete a company record.
- Generate a list of all companies of a certain type.
- Print this list.

- Export a selected list of contacts to a text file or spreadsheet format.
- Import a text file or spreadsheet file of contacts from another application.
- Add a photograph to one or more records.
- Create and save a custom report.
- Customize the menu structure.

- During each phase of the test, have observers document the user experience as they perform each task.
- When the test is complete, conduct an interview with the user or provide a written questionnaire to document other aspects of the user’s experience, such as his or her perception of usage versus specification.
- In addition, write down detailed instructions for user tests, to ensure that each user starts with the same information, presented in the same way.

- Test User Selection
  A complete usability testing protocol usually involves multiple tests from the same users, as well as tests from multiple users. Why multiple tests from the same users? One area we want to test is user recall, that is, how much of what a user learns about software operation is retained from session to session.
- Any new system presented to users for the first time will require some time to learn, but if the design for a particular application is consistent with the industry or technology with which the target user community is familiar, the learning process should be fairly quick.

→Follow certain conventions of terminology, menu design, and perhaps even color, shading, and font usage.

→ Certainly, a developer may stray from these conventions purposefully to achieve perceived operational improvements, but if the design goes too far afield from industry standards and expectations, the software will take longer for new users to learn; in fact, user acceptance may be so slow as to cause the application to be a commercial failure.

- software targeted for a specific end-user type or industry should be tested by what could be described as expert users, people already familiar with this class of application in a real-world environment.
- software with a more general target market—mobile device software, for example, or general-purpose Web pages—might better be tested by users selected randomly. (Such test user selection sometimes is referred to as hallway testing or hallway intercept testing.

- How many users?

Data gathering methods

- Test administrators or observers can gather test results in several ways.

1. Videotaping a user test and using a think-aloud protocol can provide excellent data on software usability and user perceptions about the application.

A think-aloud protocol involves users speaking aloud their thoughts and observations while they are performing the assigned software testing tasks. Using this process, the test participants describe out loud their task, what they are thinking about the task, and/or whatever else comes to their mind as they move through the testing scenario.

- Even when using think-aloud protocol testing, developers may want to follow up with participants after the test to get posttest comments, feelings, and observations.
- Taken together, these two levels of user thoughts and comments can provide valuable feedback to developers for software corrections or improvements.
- A disadvantage to the think-aloud process, where videotaping or observers are involved, is the possibility that the user experience will be clouded or modified by the unnatural user environment.

2. Developers also may wish to conduct remote user testing, whereby the application is installed at the testing user's business where the software may ultimately be applied. Remote testing has the advantage of placing the user in a familiar environment, one in which the final application likely would be used, thus removing the potential for external influences modifying test results. Disadvantage : developers may not receive feedback as detailed as would be possible with a think-aloud protocol.

Nevertheless, in a remote testing environment, accurate user data still can be gathered. Additional software can be installed with the application to be tested to gather user keystrokes and to capture time required for the user to complete each assigned task. This requires additional development time (and more software), but the results of such tests can be enlightening and very detailed.

3. A sophisticated but potentially useful data-gathering protocol is eye tracking. When we read a printed page, view a graphical presentation, or interact with a computer screen, our eyes move over the scanned material in particular patterns.

- Research data gathered on eye movement over more than 100 years shows that eye movement—particularly how long an observer pauses on certain visual elements—reflects at least to some degree the thought processes of the observer.

- Tracking this eye movement, which can be done with video systems and other technologies, shows researchers which visual elements attract the observers attention, in what order, and for how long.

- Such data is potentially useful in determining the efficiency of software screens presented to users.

Usability Questionnaire

→A usability questionnaire should be carefully planned to return the information required from the associated test procedure. Although you may want to include some questions that elicit free-form comments from the user, in general you want to develop questionnaires that generate responses that can be counted and analyzed across the spectrum of testers. Three general types:

Yes/no answers

True/false answers

Agree/disagree on a scale

For example, instead of asking ‘‘What is your opinion of the main menu system,’’ you might ask a series of questions that require an answer from 1 to 5, where 5 is totally agree and 1 is totally disagree:

1. The main menu was easy to navigate.
2. It was easy to find the proper software operation from the main menu.
3. The screen design led me quickly to the correct software operational choices.

4. Once I had operated the system, it was easy to remember how to repeat my actions.
5. The menu operations did not provide enough feedback to verify my choices.
6. The main menu was more difficult to navigate than other similar programs I use.
7. I had difficulty repeating previously accomplished operations.