CHALMERS |  UNIVERSITY OF GOTHENBURG

DIT045 H17 Requirements and User Experience

Lecture 11: Usability & Intro to Usability Testing

Jennifer Horkoff
Email: jenho@chalmers.se

Usability

- Usability is one of many system qualities (NFRs)
  - Others include availability, performance, security, maintainability, etc., (See lecture 2)
- Has sub-qualities
  - Often called usability factors (next slide)
- We focus on the usability of user interfaces
  - The system could have other technical interfaces, e.g., to payment system, to physical sensors, etc.

(Laueson, Chapter 1)

Usability Factors

- Fit for use (functionality): The system can support the tasks the user has in real life
- Ease of learning: How easy is the system to learn for various groups of users
- Task efficiency: how efficient is it for the frequent user
- Ease of remembering: how easy is it to remember for the occasional user
- Subjective satisfaction: how satisfied is the user with the system
- Understandability: How easy is it to understand what the system does?
- Ease of use (User friendliness): combination of all factors but the first

(Laueson, Chapter 1)

Usability Factors

- Difficult to satisfy them all at once in one system
- Different factors will have different priorities depending on your system
  - Examples: web-based commercial system should emphasize ease of learning and satisfaction
  - Air-traffic control system should emphasize task efficiency and understandability

(Laueson, Chapter 1)

Usability Goals / Dimensions

- Another View on Usability factors:
  - **Effectiveness:** allows users achieve their goals.
  - **Efficiency:** allows quick performance of tasks
  - **Safety:** does not put user's safety (or the safety of any artefact, work product) at risk.
  - **Utility:** users actually use the offered features.
  - **Learnability:** easy to learn.
  - **Memorability:** easy to remember how to use

User Experience Goals

- satisfying
- enjoyable
- pleasurable
- exciting
- entertaining
- helpful
- motivating
- emotionally fulfilling
- engaging
- aesthetically pleasing
- supportive of creativity
- rewarding
- fun
- provocative
- surprising
- enhancing sociability
- challenging

- boring
- frustrating
- annoying

Another View: What Makes Something Usable?

- Usability: absence of frustration in using something
- When a product or service is usable, the user can do what they want to do the way they expect to, without hindrance, hesitation, or questions.
- To be usable a product should be:
  - Useful: can a user achieve their goals?
  - Efficient: speed
  - Effective: the system behaves the way the users expect, and the ease which users can do what the intend
  - Learnable: part of effectiveness, the ability to use the system after some amount of training (maybe none)
  - Satisfying: perceptions, feelings, and opinions on the product

(Rubin & Chisnell, Chapter 1)

- Accessibility and Usability are siblings
  - Need access to the system (particularly important for users with disabilities)

Why is a Product Hard to Use?

1. Development focuses on the machine or system
2. Target audiences change and adapt
3. Designing usable products is difficult
4. Team specialists don't always work in integrated ways
5. Design and implementation don't match

(Rubin & Chisnell, Chapter 1)

Program Defect Types

- 1. **Missing functionality or bug:** function the user wants is not there, or crashes, doesn't work
- 2. **Task Failure:** user cannot figure out how to do a task in a fixed amount of time
- 3. **Annoying:** user can do a task in a reasonable amount of time, but is annoyed
- 4. **Medium problem:** user succeeds in task after a long time
- 5. **Minor problem:** user succeeds in task after a short amount of time (but it still took longer than expected)

- **Critical problems**

(Laueson, Chapter 1)

Defect Examples

- P1: the user cannot figure out how to start the search. The screen says that he should use F10, but for some reason he doesn't see it until he has tried several other ways.
  - Medium problem
- P2: The user believes that he has completed the task and that the result is saved, but actually she should have pressed update before closing the window
  - Task failure
- P3: The user cannot figure out which discount code to give the customer, although he knows which field to use
  - Task failure
- P4: The user says it is completely crazy to have to go through six screens in order to fill in ten fields
  - Annoying
- P5: The user wans to print out the list of discount codes in order to put her own notes on it. After many attempts, she calls a hotline. They say the system cannot do that.
  - Missing functionality

(Laueson, Chapter 1)

Usability Testing

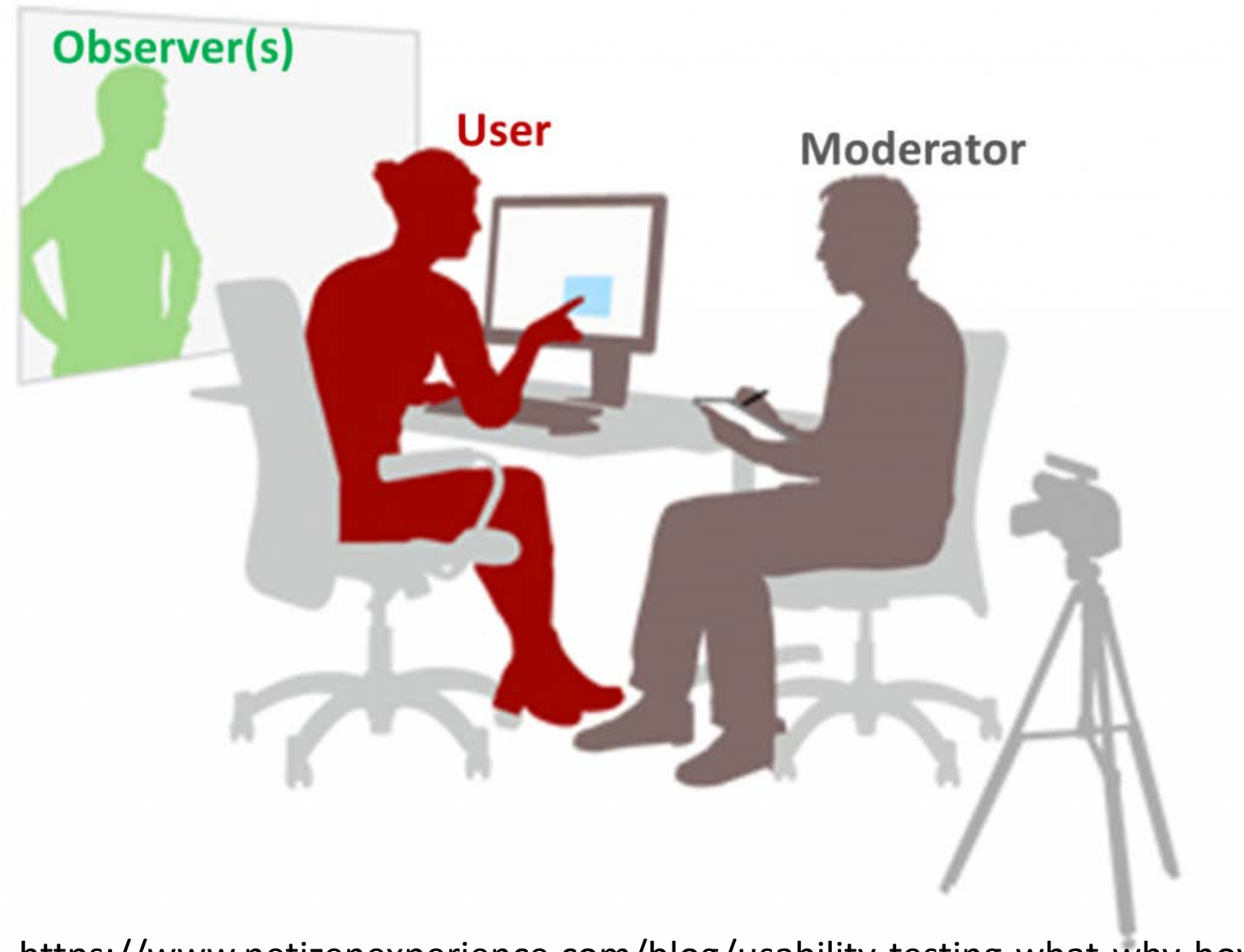
- During a usability test, we let a user (the test subject or test user) try to carry out realistic tasks using the real system or a mock-up
- One favored technique is the think aloud technique
- User thinks aloud, trying to explain what he/she is doing when doing it
- Helpers the testers to understand the rationale for the test subject's choices

Usability Testing Test Team

- Best to have 2-3 people on the test team
  - (Conveniently, the size of your groups)
- The facilitator tasks with the user
  - Moves the mockup if necessary (e.g., places down right piece of paper on click)
- The log keeper notes down what happens, in particular the problems the user encounters
- A 3rd person can observe and help

(Laueson, Chapter 1)

Usability Tests



https://www.netizenexperience.com/blog/usability-testing-what-why-how/

Usability Test Tasks

- Find tasks that the users should attempt in order to test usability
- Something they would do in a real situation
- Hint: One usability task per use case makes sense
- Usually involves several steps, clicks, screens
- Examples:
  - Wrong: Find a bus connection around 11 pm from route 6, stop 12 to route 8, stop 23.
  - Better: You are planning to go to a party tomorrow at 20 Brickwood Street, Brighton. You would like not driving home to 55 Westbank Terrance, Richmond. IS there any public transportation that can help you? How late? And what would it cost?
  - Make it realistic
- Some examples for the online marking tool uploaded to GUL

(Laueson, Chapter 1)

Carrying out the Test

- Users are nervous (usually), don't want to look dumb
- Explain the purpose of the test:
  - "We want to find out where the system is hard to understand or inconvenient. We know the system too well, so we cannot see it ourselves. We need your help. If you have problems with the system, it's the system's fault -- not yours."

Plan
Test-users
Test-tasks
Study system yourself

Carry out
Explain Purpose: find problems, system's fault
Give task – think aloud please
Observe, listen, note down
Ask cautiously: What are you looking for? Why?
If they are totally lost, help them out (task failure)

Reporting: list usability problems within 12 hours

(Laueson, Chapter 1)

Other ways to Test the System

- Heuristic evaluation: UI experts evaluate your design using heuristic guidelines
- User reviews: find an experienced user (if possible) and discuss what is missing and wrong
- Experiments (next lecture): structured studies meant to test alternatives

(Laueson, Chapter 1)

Usability Metrics

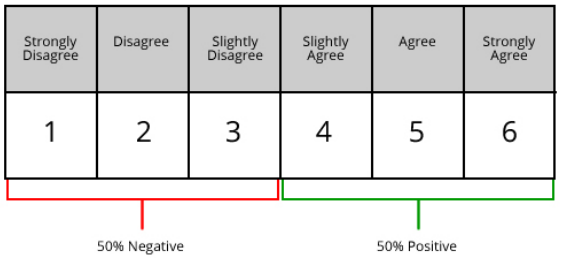
- Task time (performance measurement)
  - How long did it take the user to complete a task?
  - Specify targets: you did this already (hopefully) in your NFRs in A1 and A2
  - Examples: 70% of users should be able to rent a bike within 90 seconds
  - 80% of users should be able to return a bike within 60 seconds

- Problem counts
  - Count the number and type of errors per task
  - Set targets again
  - Examples: 95% of users should not report frustration when renting a bike
  - 90% of all users should be able to complete 90% of all tested tasks (list use cases)

(Laueson, Chapter 1)

Usability Metrics

- Keystroke counts
  - Count the number of keystrokes, mouse clicks and other operations for each task
  - Specify targets again, example: User must be able to rent a bike within less than 8 clicks, including entering payment card and PIN.
  - Hard to do in your assignments (usually have automated tools to capture this, observation too difficult and error-prone)
- Opinion polls (surveys)
  - Ask the user a number of questions
  - Questions should not be leading
  - Usually the users answers via a Likert scale
  - Will learn about this more in next lecture



(Laueson, Chapter 1)

Metrics vs. Usability Factors

Factor	Time	Problem Count	Keystroke	Polls
Fit for use	Hard to measure all tasks	Hard to measure all tasks	Hard to measure all tasks	Well-suited
Ease of learning	Well-suited	Well-suited	Not suited	Some indication
Task efficiency	Well-suited (experienced users)	Indirectly	Some indication	Some indication
Ease of remembering	Difficult, need to run twice	Difficult, need to run twice	Not suited	Some indication
Subjective satisfaction	Not suited	Some indication	Not suited	Some indication
Understandability	Not suited	Some indication	Not suited	Some indication

(Laueson, Chapter 1)

Questions?

Sources