HUMAN COMPUTER INTERACTION

Dr Kami Vaniea
18th September 2017
First, the news...

- First 5 minutes we talk about something interesting
- You will not be tested on the news part of lecture
- You may use news as an example on tests
- Why do this?
  1. Some students show up late
  2. Reward students who show up on time
  3. Important to see real world examples
What level of technical skill can we expect out of “average” users?
Distribution of Computer Skills Among People Aged 16–65

Level 3 (strong)
Level 2 (medium)
Level 1 (poor)
Below 1 (terrible)
Can’t use computers

https://www.nngroup.com/articles/computer-skill-levels/
Computer skills at each level can be characterized by the types of tasks a person at this level can accomplish.

"Can’t use computers" is obviously the bottom with people who cannot use computers at all.

(26% of adults)

Level 1
(29% of adults)
Widely available programs, minimal navigation, few steps, minimal number of operations

"Find all emails from John Smith"

Level 2
(26% of adults)
Generic and specific applications, navigation across pages, multiple steps, goal may have to be defined by the person, unexpected outcomes or impasses likely to occur.

“You want to know what percentage of the emails sent by John Smith last month were about sustainability.”

(20% of adults)

Level 3
(5% of adults)
Generic and specific applications, navigation across pages, multiple steps, goal may have to be defined by the person, high monitoring demand, unexpected outcomes or impasses likely to occur.

https://www.nngroup.com/articles/computer-skill-levels/
Today...

1. Course introduction
2. Design process
3. Two examples:
   - App permissions
   - Evaluating usability of email encryption plugin
Pronouncing my last name:

English: Van-yay
French: Vanier

Bit of American history:
Which course should I take?

- Human-Computer Interaction
  - Practical applied class
  - Emphasis: How do you build and test a user interface
  - Programming experience assumed
  - 30% coursework, 70% exam

- The Human Factor: Working with Users
  - More theoretical with some practical
  - Emphasis: strong knowledge of theory grounding
  - No programming knowledge
  - 100% coursework
Course Introduction
Modules

- Design requirements gathering
- Designing an interface
- Evaluating an interface
Books

- Quick guide to common methodologies
- Practical guide to building and testing usable interfaces
Coursework

CW1: Prototype a smart refrigerator app
- Groups of size 2
- Decide on tasks to support
- Create a functional prototype in Processing

CW2: Evaluate an app
- Groups of size 2
- Randomly given another group’s prototype from CW1
- Evaluate if it is usable
Readings

• Short readings
  • Should take less than 10 minutes to read
  • Typically only 2 pages per methodology
  • I expect you to know this, likely will show up on exam

• Long readings
  • Everything you need to know
  • Further clarification of slide material

• Supplemental readings
  • Extra information for those who are interested
Tutorials

- Starting in the third week
- Focus on hands-on doing of the methodologies
- Work through some sample exam questions
Any questions about the course setup?
Design Process
Design process

Many design processes, we use the Universal Methods of Design one

1. Planning, scoping, and definition
   • What do we want to do?

2. Exploration, synthesis, and design implications
   • Would it work? Would it solve the problem?

3. Concept Generation
   • Create a prototype and try it out

4. Evaluation, refinement, and production
   • Build it, test it, fix it

5. Launch and monitor
   • See if it works in the real world and perform ongoing review
**Design process**

1. Planning, scoping, and definition
   - What do we want to do?

2. Exploration, synthesis, and design implications
   - Would it work? Would it solve the problem?

3. Concept Generation
   - Create a prototype and try it out

4. Evaluation, refinement, and production
   - Build it, test it, fix it

5. Launch and monitor
   - See if it works in the real world and perform ongoing review

1. What is wanted/needed?

2. Analysis

3. Design

4. Prototyping

5. Implement and deploy
What is wanted / needed

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Dialog notations

Prototype

Implement and deploy

Architectures
Documentation
Help

Evaluation
Heuristics

What is there now vs.
what is wanted

Design process
The methods book always lists what design phases a method can be used in.
The following is part of a MSc project from 2016 on re-designing permission screens for Android.
Describing how an app uses permissions

Allowed to do

Actually does

- SMS_RECEIVED
- READ_PHONE_STATE
- MAIN
- SEND_SMS
- click
- SEND_SMS
Static analysis: Breaks an app up into a control flow diagram

requestLocationUpdates
Get access to your location probably to send it on using the later openConnection call
The brief:

Create a new permission screen using the output from a static analysis tool that helps people understand the context in which permissions will be used.
Problem 1:

What permissions do people worry about?

Sub-problem:

Most people don’t understand permissions enough to actually worry about them
Solution:

Affinity diagram using Computer Security MSc students
What is wanted / needed

What is there now vs. what is wanted

Analysis

Scenarios
Task analysis

Guidelines
Principles

Evaluation
Heuristics

Dialog notations

Design

Precise specification

Prototype

Implement and deploy

Architectures
Documentation
Help

Design process
Affinity diagram study protocol

1. Pre-print a list of Android permissions and contexts
2. Have students brainstorm answers to questions onto sticky notes
   A. Name three permissions
   B. App behaviors you are not comfortable with
   C. Situations that would cause a permission to be used
3. Put all notes on the wall and do an affinity diagram
4. Encourage hierarchy design
5. Discuss outcome with participants as a group
Pre-printed contexts

PERMISSION

ACCESS_CHECKIN_PROPERTIES

Allows read/write access to the "properties" table in the checkin database, to change values that get uploaded.

PERMISSION

ACCESS_COARSE_LOCATION

Allows an app to access approximate location.

CONTEXT

DESTROY_METHOD

Called when an activity* finishes its lifecycle. Called once in the lifecycle of an activity*

CONTEXT

SERVICE_METHOD

A Service is an application component that can perform long-running operations in the background and does not provide a user interface.

*activity: application window
Initial sorting

• The notes are then sorted by the students into groups
• New notes could be added
CAMERA.

- Push the camera button
- Taking a picture
- Allow to use camera
- Microphone/camera use without specifically having requested it
- Using camera
- Camera
- Open camera
- Use camera
- Take a picture
- Apps like Snapchat would need to use your camera
- Use allow camera
- Capture secure video output
- Capture video output
- Location/Map
- Access location
- Install location provider
- Access coarse location
- Location hardware
- Control location updates
- Gather my location without my permission (default setting)
- Retention
- Record location
We did that to the whole wall
What is wanted / needed

What is there now vs. what is wanted

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Dialog notations

Architecture
Documentation
Help

Design

Implement and deploy

Prototype

Evaluation
Heuristics
In the Background

- SMS
- Contacts
- Calendar
- Microphone
- Camera
- Location
- Accounts and device info
- Internet combined with read data

Settings
- Ads
- Sounds
- Notifications
- Screen space
Outcomes

- “with my permission”
  - Button presses
  - Opening an app
- Background vs. foreground
  - When the permission is accessed is important
- Purpose
  - Ads
  - Uploading private data like contacts and device ID
- Sensitive permissions focused on input/output
- Confusing permissions
What is wanted / needed

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Dialog notations

Evaluation
Heuristics

Prototype

Design

Implement and deploy

Architectures
Documentation
Help

Design process
We designed an interface that shows permissions in context of when they can be used.

- **Button push required**
  - Contacts
    - modify your contacts
    - read your contacts

- **Only when app is open**
  - Calendar
    - add or modify calendar events and send email to guests without owners' knowledge
    - read calendar events plus confidential information

- **Anytime in the background**
  - Identity (Ad software)
    - find accounts on the device
    - add or remove accounts
    - read your own contact card

- Location
  - approximate location (network-based)
What is wanted / needed

Analysis
- Scenarios
- Task analysis

Design
- Guidelines
- Principles
- Precise specification

Prototype
- Dialog notations

Implement and deploy
- Architectures
- Documentation
- Help

What is there now vs. what is wanted

Evaluation
- Heuristics
Created two interfaces to A/B test

(a) Control group screen  (b) Experiment group screen

Figure 5.1: Survey question screens
Which of the following can this app do?

<table>
<thead>
<tr>
<th></th>
<th>Absolutely Impossible</th>
<th>Impossible</th>
<th>Neutral</th>
<th>Possible</th>
<th>Absolutely Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge purchases to your credit card at any time.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get your location.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allow ads to know your location.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load ads.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write on the SD card</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is wanted / needed

What is there now vs. what is wanted

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Dialog notations

Evaluation
Heuristics

Prototype

Design

Implement and deploy

Architectures
Documentation
Help
The results:

PERMISSION STATEMENTS - CONTROL CONDITION

Neutral 13%
Incorrect 27%
Correct 60%

PERMISSION STATEMENTS - EXPERIMENT CONDITION

Neutral 13%
Incorrect 26%
Correct 61%

(a) Control Group  (b) Experiment Group

Figure 5.2: Permission Statements Results: Correct, Incorrect and Neutral
27% of people think they know what this screen says and are wrong.

13% are uncertain what this screen really means.
The following is part of a MSc project from last summer on evaluating an email encryption plugin.
The brief:

Google released a new plugin for email encryption called Mailevelop, is it usable?
Subject: Happy birthday!

To: frankchou1116@gmail.com

Encrypt attachments

Sign message with key: Qingyu Zhou <frankchou1116@yahoo.com> - 01C23B378BC3

Sign all messages with primary key

Options

Send

Sign Only

Cancel

Encrypt
Design process

What is wanted / needed

What is there now vs. what is wanted

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Design

Dialog notations

Implement and deploy

Architectures
Documentation
Help

Evaluation
Heuristics

Prototype
We already know a lot about what people want from email.

We already know why email encryption hasn’t worked.
Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0 by Whitten and Tygar

- Asked 12 Carnegie Mellon Computer Scientists to correctly send an encrypted email using PGP 5.0
- Only 4 managed to accomplish this within 90 minutes
- Dangerous errors
  - Accidentally emailing without encrypting
  - Confusions around key system
  - Giving up
What is wanted / needed

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Design

Dialog notations

Prototype

Implement and deploy

Architectures
Documentation
Help

Evaluation
Heuristics

What is there now vs. what is wanted

Design process
Cognitive Walkthrough

Scenario 1: User has already installed the Mailvelope plugin and wants to send an encrypted email to another person.

Step 1: Open the Mailvelope plugin by clicking on the icon.

Q1. Will users try to achieve the outcome of clicking on this button?

Q2. Will users see this button for the action?

Q3. Once users find this button, will users recognize that clicking on it will produce the effect they want?

Q4. After the action is performed, will users understand the feedback, so they can confidently continue on to the next action?
Cognitive Walkthrough

Step 2: Click on the “Options” button.

Q1. Will users try to achieve the outcome of clicking on this button?

Q2. Will users see this button for the action?

Q3. Once users find this button, will users recognize that clicking on it will produce the effect they want?

Q4. After the action is performed, will users understand the feedback, so they can confidently continue on to the next action?
Cognitive walkthrough identified expected areas of failure.

Next we setup a think aloud study to see if actual users would fail where expected.
<table>
<thead>
<tr>
<th></th>
<th>Webmail login</th>
<th>Composing email on</th>
<th>Opening Mailvelope popup</th>
<th>Sending encrypted email</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Success(hint)</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T2</td>
<td>Success(hint)</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T3</td>
<td>Success(hint)</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T4</td>
<td>Success(hint)</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T5</td>
<td>Success(hint)</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T6</td>
<td>Success</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T7</td>
<td>Success</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T8</td>
<td>Success</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
<tr>
<td>T9</td>
<td>Success(hint)</td>
<td>Mailvelope popup</td>
<td>Success</td>
<td>Success</td>
</tr>
<tr>
<td>T10</td>
<td>Success</td>
<td>Webmail editor</td>
<td>Failure</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Table 4.3: Completion details of Task 2 for each participant.
Design process

What is wanted / needed

What is there now vs. what is wanted

Analysis

Scenarios
Task analysis

Guidelines
Principles

Precise specification

Prototype

Dialog notations

Evaluation
Heuristics

Design

Implement and deploy

Architectures
Documentation
Help
Design process

What is wanted / needed

Scenarios
Task analysis

Analysis

Guidelines
Principles

Dialog notations

Prototype

Precise specification

Design

Implement and deploy

Architectures
Documentation
Help

What is there now vs. what is wanted

Evaluation
Heuristics
All participants selected the email provider from the dropdown list and clicked on the “OK” button. All of them noticed the auto-opening popup but they did not think it belongs to the Mailvelope. However, they considered it as the webmail editor. They all intended to write the email on this popup. D1 said “it is clear for me to compose the email on this editor.”
Questions