TESTING METHODS

THINKING ALOUD

- Thinking aloud may be the single most valuable usability engineering technique – often used in other methods also - Mixing methods
- A thinking-aloud test involves having a test subject use the system while continuously thinking out loud.
- By verbalising their thoughts, the test users enable us to understand how they view the computer system, and this makes it easier to identify the users’ major misconceptions.

THINKING ALOUD

- We do not know what people think; thinking aloud gives a glimpse of what is going on.
- Thinking aloud in this manner involves tasks
- Users are asked to ”describe what they are doing”.
  - It’s usually necessary to explain the method to the user in terms of: ”think aloud; tell [me] what you are doing, what you are looking at or looking for, what do you expect. And always when you’d like to know something, please ask.”
  - **First task is usually quite simply** to help the user to concentrate on thinking aloud.

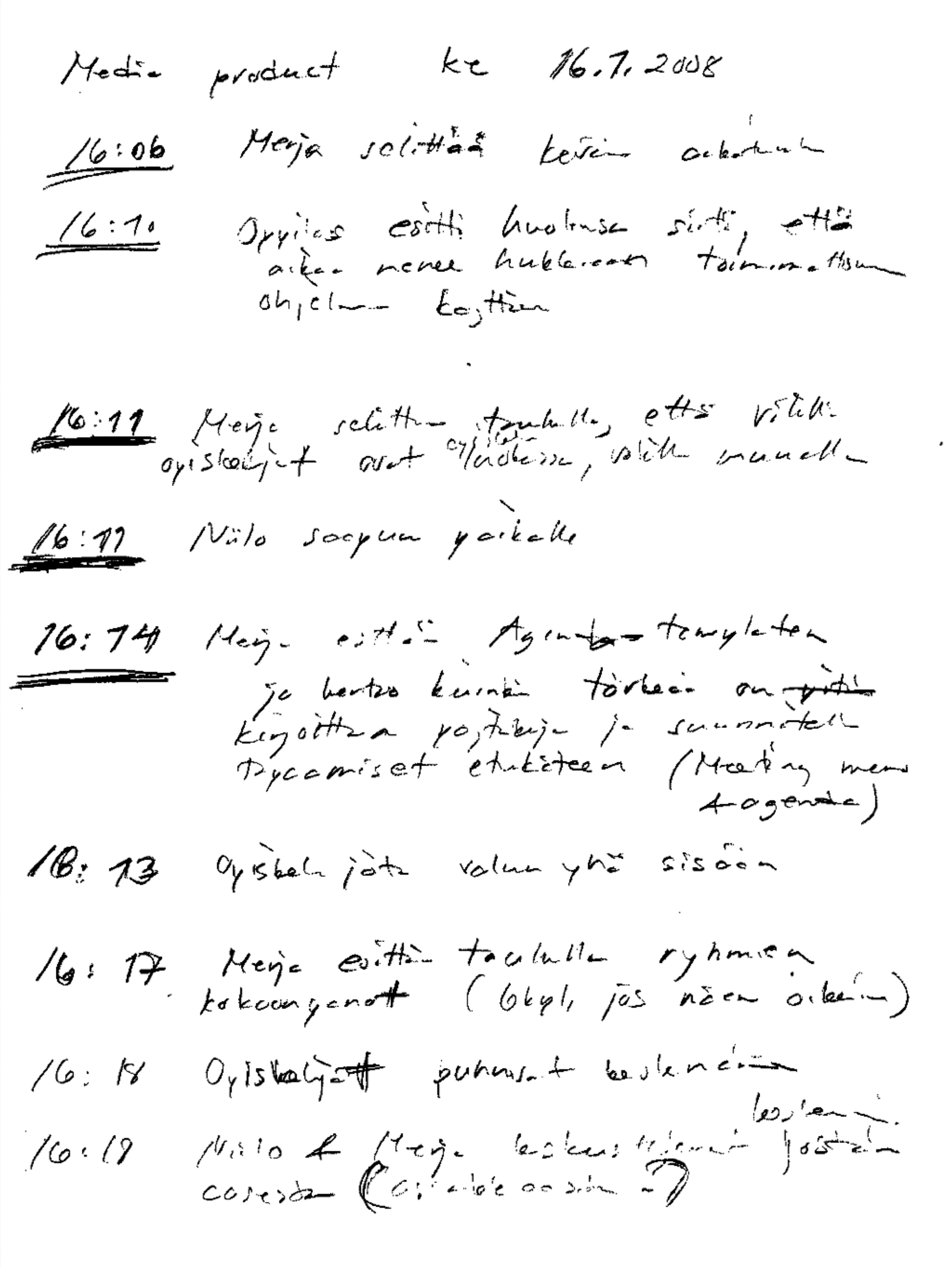
THINKING ALOUD

- Help the user to think aloud during testing to get better results (and more feedback).
  - Questions such as ”What are you looking for now”, ”what do you think about what you see” may help the user to think aloud.
- If the user ceases thinking aloud:
  - There may be something worth of reading, or
  - The user has to concentrate on the interface.

USABILITY TEST VARIATIONS

- Testing in pairs
    - Two users at a time; more natural discussion and problem solving
- Effectiveness testing
    - Main interest in time-per-task and percentage-of-errors (the US way)
- **Observation as testing**
    - Users are observed as they do their normal tasks

OBSERVATION



Media product ke 16.7.2008

16:06 Merja selittää kevään aikataulun

16:10 Oppilas esitti huolensa siitä, että aikaa menee hukkaan toimimattoman ohjelman käyttöön

16:11 Merja selittää taululle, että viikoilla opiskelijat ovat luokassa, välillä muualla

16:11 Niilo saapuu paikalle

16:14 Merja esittää Agenda-templaten ja kertoo kuinka tärkeää on ~~pitää~~ kirjoittaa pöytäkirjaa ja suunnitella tapaamiset etukäteen (Meeting memo + agenda)

16:13 Opiskelijoita valuu yhä sisään

16:17 Merja esittää taululle ryhmien kokoonpanot (6kpl, jos näen oikein)

16:18 Opiskelijat puhuvat keskenään

16:19 Niilo & Merja keskustelevat keskenään jostain casesta (oikeaa asiaa ?)

TWO METHODS IN OBSERVATION

- "Unobtrusive observation" means you observe what test users do and refrain from interacting with them
  - With unobtrusive observation you learn whether the system is easy to use.
- "Obtrusive observation" means you interact with test users, e.g. by asking questions.
  - With obtrusive observation you learn more about the usefulness and acceptance of the system.

ONE WAY TO START ....

- Start each usability test with unobtrusive observation: you observe how users execute the tasks you give them or what they naturally do without pre-given tasks **in use context.**
  - After that, you reserve some time to ask questions, explain design decisions, and answer the test user's questions.

WHAT DO YOU GET OUT OF IT

- With unobtrusive observation you learn whether:
  - People can **use your design** in an **easy and efficient way,** and where this is not the case.
- How people behave and how people explain their behaviour are two different things.
- If you want to learn about behaviour, you have to study behaviour.
- If you want to learn whether users can use your design, you have to observe how they use it, not ask them what they think of it.

TIPS FOR UNOBTRUSIVE OBSERVATION

- **1. Observe: be quiet, watch, understand.**
  - Don't explain
  - Don't ask the test user's opinion
  - Don't defend the design
  - Don't apologise
  - Don't suggest
  - Don't contradict the test user nor agree with him/her: stay neutral

TIPS FOR UNOBTRUSIVE OBSERVATION

- **2. Only help to overcome the limitations of the prototype.** Explain briefly and in a neutral way what would happen in the future system.

- 3. **First observe, then take notes.** Don't let your note taking get in the way of observing what the test user is doing. You don't have to write down everything you notice during observation. Instead, take 15' to clean up and complete your notes after the observation session. But don't postpone it because you will quickly forget important details.

- **4. Stimulate users to think aloud.** But use neutral prompts, e.g. “What do you see, what are you thinking, what do you want to do, what are you looking for?”

TIPS FOR UNOBTRUSIVE OBSERVATION

- **5. Limit the time test users have to execute a task.** Don't prolong the test user's suffering longer than necessary. If a test user is really stuck on a task and you have learned why this is the case, thank the user for trying and ask him/her to continue. Usability testing can be a frustrating experience for test users because people seem to be naturally inclined to do the best they can.

- **6. Elicit detailed information.**
  - Test User: “I see a lot of information.”
  - Observer: “Could you tell me what information you see?

TIPS FOR UNOBTRUSIVE OBSERVATION

- **7. Answer test user questions with questions**. Dealing with test user questions is probably the most difficult aspect of unobtrusive observation. It seems natural and polite to answer questions.
  - **Before each usability test, you should explain the test user how you will deal with questions.**
  - You should encourage the test user to ask questions because that allows you to learn what is not clear in the system.
  - Explain that you will not answer these questions right away but that you will write them down and answer them at the end of the test.

TIPS FOR UNOBTRUSIVE OBSERVATION

- A productive way of dealing with test user questions is to answer them with questions. If you simply leave the question unanswered, the test user might feel ignored.
  - His/her motivation to think aloud and to continue asking questions will diminish or even vanish.
- To minimise your influence on the test user, use the same words as the test user.
  - Test User: “What does this text mean?”
  - Observer: “What do you think it means?”
  - Test User: “Do I have to click here?”
  - Observer: “What do you think will happen if you click there?”

OBTRUSIVE OBSERVATION

- What **unobtrusive** methods **will not** tell you, is what test users think of your design. I.e.,:
  - Do they like to use it?
  - Does it answer their needs?
  - You can't observe opinions by just watching people.
- Even extreme emotions are difficult to observe accurately: does my test user get angry because s/he doesn't understand my design or because s/he thinks it is so simple it insults her/his intelligence.
- If you want to learn more about the usefulness and the acceptance of your design, **you will have to ask test users.**

TIPS FOR OBTRUSIVE OBSERVATION

- **1. Think about what you want to ask before the test.** When you have just finished a test session with unobtrusive observation, you will have a lot to talk about with the test user.
  - You can ask the user to clarify or explain actions you have observed, you can explain and discuss design decisions, etc.

- But you will also need a checklist of things you want to know from all test users. This checklist will contain items related to functionality (what is useful - useless - missing?) and items related to user acceptance (what do they like - don't like).

- **2. Ask open questions**. Avoid closed questions, i.e. questions that can be answered with yes or no. You will get more detailed and accurate information with open questions.
  - Bad: “Do you understand what this means?”
  - Good: “What do you think when you see this?”
  - Bad: "Did you know you can click here to achieve that?"
  - Good: "What would you do if you would want to achieve that?"

TIPS FOR OBTRUSIVE OBSERVATION

- **3. Don't blame the test user.**
- Remember: you're testing the usability of the design, not the computer literacy of the test user. If the test user does not understand something, this something will have to be improved in the design.
  - Bad: “Why don't you understand this?”
  - Good: “Could you tell me what this means for you?”
- **4. Don't ask the test user for design solutions.**
- Test users are not interaction designers. That is why test users will almost never provide you with good design solutions. Don't bother asking, but take note of suggestions they make spontaneously.
  - Bad: “Do you need a News button here?”
  - Good: “Which information do you need at this point?”

TIPS FOR OBTRUSIVE OBSERVATION

- **4. Don't ask the test user for design solutions.**
  - It is sometimes useful to ask the test user to compare the system with related products s/he has used before.
  - By using interactive systems everybody gains tacit knowledge of interaction design. But because test users are not interaction designers, they lack the vocabulary to make this tacit knowledge explicit.
  - It is much easier to compare the system with actual experience with a related product.
    - Bad: "How can we redesign this page to make it easier for you?"
    - Good: "Compared to the product you have used before, what do you think is better or worse in this system?"

TIPS FOR OBTRUSIVE OBSERVATION

- **5. You can do obtrusive observation with groups of test users**.
- People often find it easier to formulate their opinions when they are confronted with the opinions of others. That is why you learn a lot about the usefulness and acceptance of your design when you ask a group of test users to discuss your design.
- Your role is to facilitate and focus the discussion.
- Group discussions are harder to organise than individual sessions because you need several test users at the same place at the same time.
- They are also harder to manage because you have to control not only individual reactions but also group processes.
- You will want to stimulate some group processes (e.g. opinion disclosure) but at the same time avoid unwanted group effects (e.g. group intimidation).

LINKS:

- http://www.interactionarchitect.com/knowledge/article19991212shd.htm
- http://www.otal.umd.edu/hci-rm/ethno.html
- http://www.ul.ie/~infopolis/methods/observat.html

QUESTIONNAIRES

QUESTIONING TECHNIQUES

- Questionnaires are sets of predefined questions arranged on a form and typically answered in a fixed sequence.
- The questionnaire is completed by the individual representing user section.
- There are essentially two types of questions : closed questions (respondents are asked to select an answer from a choice of alternatives) or
- Open questions (the respondent is free to provide his own answer).

QUESTIONING TECHNIQUES

- Questionnaires can be used in a very broad series of applications.
  - They can be used for evaluating specific features or issues of a system, and
  - They can also be used in order to investigate different opinions, perceptions, knowledge or attitudes towards this system.
- Designing a questionnaire is a much more complex task than it might appear.
- It is important to make the questionnaire easy for the user to complete. There are two quite obvious reasons for this.
  - 1) in order for the analysis to be statistically significant it is necessary to generate a high return rate.
  - 2) it is important to **avoid ambiguity** in the questions which are posed. Ambiguous questions generate unreliable answers.

QUESTIONING: ADVANTAGES

- Are the flexibility of administration; questionnaires can be filled in at any time by any number of individuals. Because it is relatively easy to administer a large number of questionnaires, they are especially useful for comparing the opinions or understanding of a large number of users.
- The use of closed questions can prevent digressions (going aside of the question).
- The use of multiple choice items and rating scales can produce answers that can be weighted for their reliability and consistency.
- Careful decisions have to be made concerning the sample of user population and whether they are representative of the whole group of interest.
- Is to gather a large amount of information from many people, relatively rapidly and at low cost.

QUESTIONING: DISADVANTAGES

- Respondents do not necessarily give correct answers and may often be influenced by what they believe the analyst requires, or what they themselves wish to portray, especially for personal information. Anonymity and having the questionnaire completed remotely from the analyst can reduce distortions.
- Loss of information or subjective insights may be introduced by inappropriate perceptions of what the question is asking. The correct choice of target population minimises this risk.
- There is room for considerable bias in what questions are asked and how the answers are interpreted, mainly if open-ended questions are incorporated.
- Response rates for posted questionnaires can be very low (30 % or less is considered as normal).

TYPES OF QUESTIONING TECHNIQUES

- **Multiple Choice Items**
  - Multiple Choice items provide two or more specific responses from which respondents must choose an item which is most representative of their opinions, perceptions, knowledge or attitudes.
- **Rating Scales**
  - Rating scales can be used to obtain subjective information from respondents which gives an indication of both the nature and magnitude of their opinions about certain aspect of a task.
- **Bipolar alternatives**
  - Bipolar alternatives is a type of rating scale, where respondents must make a choice between two alternatives.
- **Ranking**
  - Ranking requires respondents to order several items according to precise criteria.
- **Open-ended questions**
  - The respondents are free to write their own comments, answer or opinions on the questions.

LINKS:

- http://www.ul.ie/~infopolis/methods/question.html
- http://oldwww.acm.org/perlman/question.html

INTERVIEWING

OVERVIEW

- The interview technique is a systematic collection of verbal information.
- It consists in asking about users opinions and attitudes to get basic information with prepared questions asked by the interviewer.
- The answers are either written or recorded.
- The interviews can be structured or unstructured.
- The terms **structured interviews** implies that the content of the interview, in terms of the questions and their sequence, is predefined.
  - Because of the structuring the interview offers the opportunity for more systematic collection data.
- The **unstructured interview** is more free, and the interviewee develops the themes proposed by the interviewer

APPLICATION

- Interviews can be used at any stage during a comprehensive task analysis activity.
  - They can usefully be applied early on in an investigation for collecting basic information about the situation.
  - An interview with the user will provide a deeper understanding of the requirements for the system and of the user tasks involved.
- It can also assist in finding out more about individual user perceptions
- In order to encourage frank answers, the participant should be assured that his contribution will remain confidential.
- If any recording is made, the interviewer should also assure that this will remain confidential. Newman and Lamming (1995) describe a number of elements of a successful interview: ...next slide....

APPLICATION

- Determine some **basic domain knowledge before the interview** so that time is efficiently utilised during the interview.
  - Clearly state to the interviewee the purpose of the interview at the outset.
  - Enumerate all user activities with general and follow-up specific questions.
  - Find out how user activities are performed.
  - Trace interconnections with other users.
  - Uncover issues that determine and affect the performance of the user tasks.
  - Follow up on exceptions, the unusual activities which are unlikely to occur during observation.

ADVANTAGES

- Interview techniques are useful for identifying possible areas for more detailed analysis.
- Interviews are easy to conduct and direct, the unstructured interview can generate interesting points
- The data collected provides information about general rules and principles and is faster than observational techniques.
- Interview techniques are useful for investigating events which occur infrequently.
- The interviews can be recorded for a future analysis.

DISADVANTAGES

- Respondents are not committed to give correct answers and may often be influenced by what they believe the interviewer requires, or what they themselves wish to portray.
- The interviewer may need to acquire domain knowledge in order to know what questions to ask.
- There is a range of considerable bias due to the understanding by the users of the questions, and the subjective collected information might be misleading or inaccurate.
- The critical aspects are the choice of the place for the interview and how to conduct it.

TYPES OF INTERVIEW TECHNIQUES

- **Unstructured interview or depth interview**
  - In an unstructured interview, the respondent must develop the theme proposed by the interviewer.
  - The investigator does not impose a questionnaire and must follow the thoughts and reactions of the respondent.
  - Unstructured interviews are good for investigating potential emotional and / or sensitive personal issues.

TYPES OF INTERVIEW TECHNIQUES

- **Semi-structured interview**
  - Semi-structured interviews should only be carried out in a situation where broad issues may be understood, but the range of respondents' reactions to these issues is not known or suspected to be incomplete.
  - This type of interview is mostly applicable in situations where both qualitative and quantitative feedback are required.
- **Structured interview**
  - Structured interviews are useful in situations where the respondents range of replies <u>may be estimated</u> and there is a need to <u>clarify details, opinions or ideas.</u>
  - Structured interviews work well **when the assessment goals are clear.**

THE ART OF INTERVIEWING

- Have a protocol (list of questions to cover)
- Don’t feel bound by it; keep rapport and adjust to participant
- Avoid leading questions
- Pause and let them talk
- Practice: record/transcribe it and see what you learn
- Quotes are data!

ALL KIND OF LINKS:

- http://jthom.best.vwh.net/usability/
- http://user.meduni-graz.at/andreas.holzinger/holzinger%20de/usability%20holzinger.html
- http://jthom.best.vwh.net/usability/

PAPER PROTOTYPING

SUMMARY

- Paper prototypes or other mockups are used clarify requirements and enable draft interaction designs and screen designs to be very rapidly simulated and tested.
- Benefits
  - Potential usability problems can be detected at a very early stage in the design process before any code has been written.
  - Communication between designers and users is promoted.
  - Paper prototypes are quick to build / refine, thus enabling rapid design iterations.
  - Only minimal resources and materials are required.

PLANNING

- Arrange a workshop attended by:
  - user(s) [including other stakeholders if possible]
  - developer(s)
- You will also need a facilitator and a person to record the issues raised during the meeting.
- Four stages of paper prototyping may be required:
  - concept design: to explore different metaphors and design strategies
  - interaction design: to organise the structure of screens or pages
  - screen design: for initial design of each individual screen
  - screen testing: to refine the screen layout

1. CONCEPT DESIGN

- Sit round a table and sketch out possible approaches in a brainstorming environment.
- Evaluate the extent to which each approach meets the usability requirements and objectives agreed in the stakeholder meeting

2. INTERACTION DESIGN

- Use affinity diagramming to structure the user interface:
  - Write the name of each suggested screen, page or activity on a post-it-note.
  - Put each post-it-note on the wall close to related notes.
  - Group the post-it-notes in clusters that are meaningful to users.
  - Consolidate duplicates.
  - Give a name to each cluster.
  - Document the sequence in which user tasks will make use of each set of post-it-notes.
  - Review how the screens/pages can be organised to simplify user tasks.

3. SCREEN DESIGN

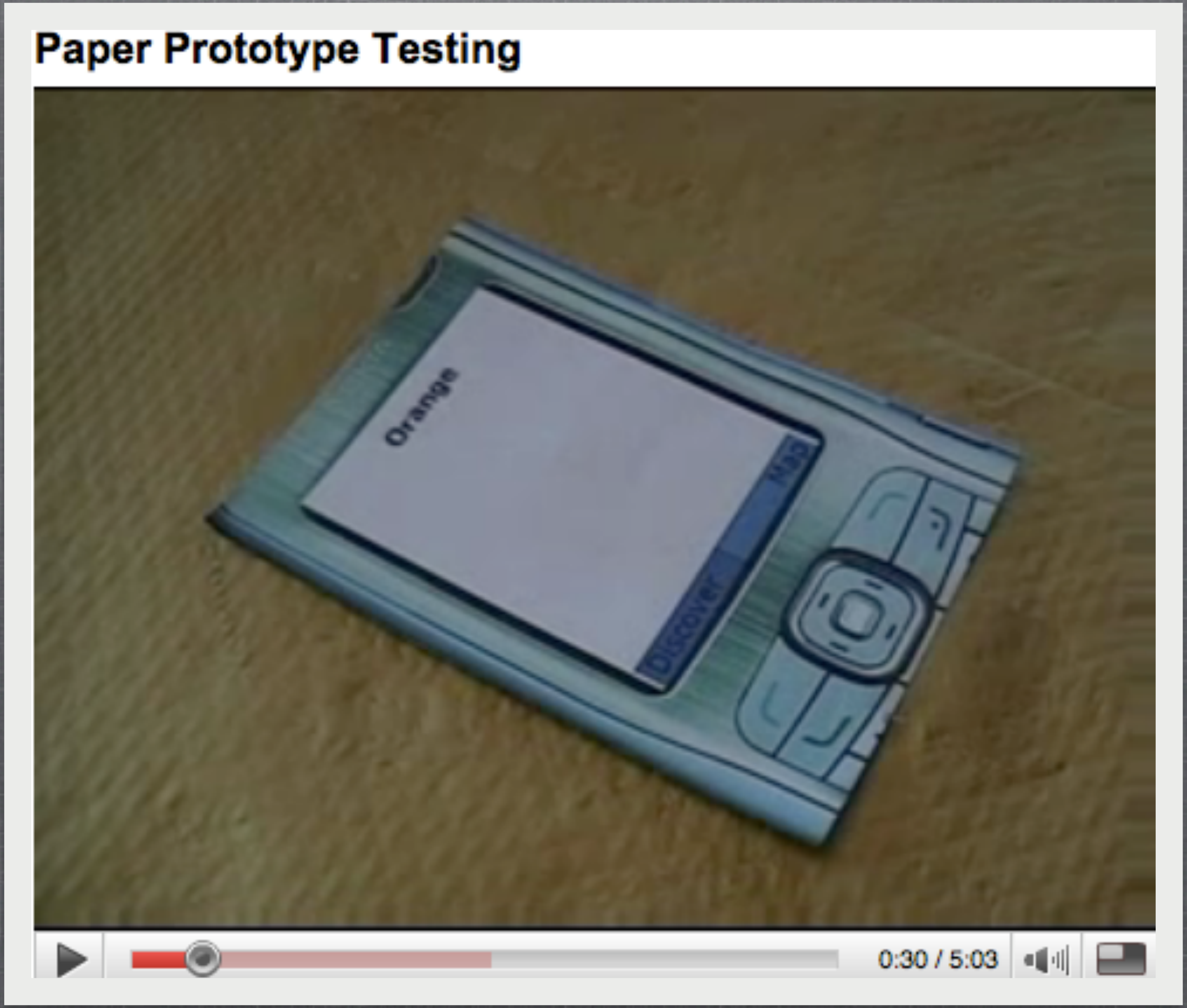
- Sit round a table and sketch out design ideas in a brainstorming environment.
- Use this as a basis for rough sketches of each screen.
- If the links between screens have not been finalised, pin each screen on the wall as for Interaction Design above.

4. SCREEN TESTING

- Use a drawing package or prototyping tool to **produce a rough design** for **each screen.**
- [If the links between screens have not been finalised, pin each screen on the wall as for Interaction Design above – not used often]
- Ask the user to **carry out a realistic task** (based on the context of use and scenarios).
- As the user selects options on each screen, the developer explains what happens, and either points to the next screen or presents the next screen to the user (without giving any hints).
- **To test more detailed interaction, prepare pieces of paper with menus, scroll boxes, dialogue boxes, etc., and present these to the user. The user simulates pointing and clicking using a pencil, and simulates typing by writing on paper**

4. SCREEN TESTING

- **Variations**
  - If the necessary skills are available, the design can be evaluated and improved by expert evaluation. This can complement prototyping, or use methods which replace users by usability experts if it is impossible to gain access to users.
- **Reading**
  - Hix, D and Hartso, H R. Developing User Interfaces
  - Dumas, JS, and Redish, Janice, A. (1999) Practical Guide to Usability Testing, Intellect Books.
  - Rubin, Jeffrey (1994) Handbook of Usability Testing. John Wiley and Sons, New York, NY
  - Snyder, Carolyn, offsiteUsing Paper Prototypes to Manage Risk, October 1996, Software Design and Publisher Magazine
  - Usability net: http://www.usabilitynet.org/tools/prototyping.htm



HTTP://WWW.YOUTUBE.COM/WATCH?V=BQ1RKVTZLTU

HEURISTICS

HEURISTICS – EXPERTS ASSESSMENTS

- Systematic inspection method for the GUI
- Goal is to find usability problems
- Check if basic design rules have been followed appropriately
- Can be made with paper prototype, prototype, functional system, or GUI itself
- Carried out often before the user test
- Usability problems found depends on the amount of evaluators
- Evaluators go trough the list of ten rules (principles)

HEURISTICS

10 heuristics

- Use of simple and natural dialogue
- Speak the users language
- Minimise users' memory load
- Make user interface consistent
- Give user feedback

10 heuristics

- Mark exits clearly
- Make shortcuts available
- Give clear error messages
- Prevent errors
- Provide enough help and documentation

HEURISTICS

Stages of evaluation

- Evaluators do it independently
- They inspect the GUI twice
  - First to get the overall look
  - Second time to look for the usability problems
- It lasts about 2 hours
- The problems found are compared with the others evaluators

HEURISTICS

- Ratings for the found problems in the scale of 0-4
  - 0 = Don’t agree that it is a usability problem at all
  - 1 = Cosmetic problem; no need to fix if not extra time and money
  - 2 = Minor usability problem; fixing is given low priority
  - 3 = major usability problem; fixing is give high priority
  - 4 = Usability catastrophe; has to be fixed immediately before any though of launching the service

- Severity can be defined also as:
  - The frequency in which the problem occurs
  - The impact the problem has on the user
  - The persistence of the problem (only ones or always)

HEURISTICS

		Proportion of users experiencing the problem	
		FEW	MANY
Impact of Problem on the Users who Experience it	Small	Low severity	Medium severity
	Large	Medium severity	High severity

Example of how to estimate the severity of usability problems based on the frequency with which the problem is encountered by users and the impact of the problems on those users who do encounter it. (Jakob Nielsen, 1993:104)

EXPERT/INSPECTION METHODS

Cognitive "Walk Trough"

- Aim is to "walk trough" all the tasks in the service that the users is supposed to perform.
- Early stages of the design (paper version "mock-ups")
- Evaluators need knownledge about psychological theories

Pluralistic Usability Walk Trough

- The service is gone trough with experts, the designers, and users and they discuss the elements
- The GUI is evaluated step by step
- It can be based on printed graphics (paper mock-ups)
- The use needs know-how and experience

Feature Inspection

- Experts list all the functions and inspect them phase by phase
- Aim is to find out complicated or long phases, illogical proceedings, parts that require previous knowledge (e.g. in Windows - start = shut down, Mac - trash gives the zip)

Consistency Inspection

- Expert inspect that the standards are used and that the GUI's work in similar ways (e.g. short cuts, commands, etc.)

COGNITIVE "WALK TROUGH"

- **What is it?**
- Cognitive walkthrough is a review technique where expert evaluators construct **task scenarios** from a specification or early prototype and then **role play the part of a user working with that interface**--"walking through" the interface.
- They act as if the interface was actually built and they (in the role of a typical user) work through the tasks.
- **Each step the user would take is scrutinized:** impasses where the interface blocks the "user" from completing the task indicate that the interface is missing something. Convoluted, circuitous paths through function sequences indicate that the interface needs a new function that simplifies the task and collapses the function sequence.

COGNITIVE "WALK TROUGH"

- **How do I do it?**
- Begin by evaluating a system specification in terms of the tasks users will perform with that system.
- It helps to identify the **user's goals and purpose for each task.** For example, the interface for operating a car begins with the goals of opening the door, sitting down in the driver's seat with the controls easily accessible, and starting the car. And we're not even driving yet!
- The example shows the **granularity that some walkthroughs attain**.
- The goal of "opening the door" could be broken down into sub-goals: find the key, orient the key, unlock the door, grasp the handle, pull to open the door.
- Each of these goals requires cognitive (thinking) and physical actions. To open the door, do I orient my hand with the palm up or with the palm down? What affordances are provided for opening the door?

COGNITIVE "WALK TROUGH"

- **How do I do it?**
- During the walkthrough, **identify problems in attaining the goals**.
- For example, some car doors accept keys only if they're oriented one way. Does this cause an unacceptable delay for the user? **Since the sub-goal of opening the door is a prerequisite to operating the car, this might be a large issue.**
- **When should I use this technique?**
- Cognitive walkthroughs are great for the **early stages** of development because they **can be performed using just system specifications as a basis**. Artists conceptions of what screens might look like can be used to give the walkthrough a more realistic bent.
- © http://jthom.best.vwh.net/usability/

COGNITIVE "WALK TROUGH"

- Rowley, David E., and Rhoades, David G. "The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure." CHI `92 Proceedings, (May 3-7, 1992): 389-395.
- Spencer, Rick. " The streamlined cognitive walkthrough method." CHI 2000 Proceedings, (April 1 - 6, 2000): Pages 353-359.
- Wharton, Cathleen, et. al., "The Cognitive Walkthrough Method: A Practictioner's Guide." in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)
- http://www.sigchi.org/chi95/proceedings/tutors/jr_bdy.htm
- http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-87/html/

FEATURE INSPECTION

- **What is it?**
- Feature inspections analyze only the feature set of a product, usually **given as end user scenarios** for the end result to be obtained from the use of the product.
- For example, a common user scenario for the use of a word processor is to produce a letter. The features that would be used include entering text, formatting text, spell-checking, saving the text to a file, and printing the letter. Each set of features used to produce the required output (a letter) is analyzed for its availability, understandability, and general usefulness.

FEATURE INSPECTION

- **How do I do it?**
- **List the features in the product in the sequences they would be used to perform various tasks.** Look at the accessibility of each feature in the context of the tasks.
  - Can the user get to each feature without much trouble?
  - Are the features well named and easily recognized?
- One time-tested way to perform feature inspection is to have the documentation staff attempt to document each user scenario as procedures. Features that are hard to describe in the documentation are probably hard to find for the user in the first place.

FEATURE INSPECTION

- **When should I use this technique?**
- This technique is best used in the **middle stages** of development. At this point, the functions of the product--what the product is to be used for--are known.
  - **The features users will use to produce their desired output are known.**
  - **The question that remains is how hard is it to use those features**?
- Bell, Univ. of Colorado; Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)

FEATURE INSPECTION

- © http://jthom.best.vwh.net/usability/
- http://jthom.best.vwh.net/usability/ftrinsp.htm
- http://issco-www.unige.ch/ewg95/node86.html

CONSISTENCY INSPECTION

- **What is it?**
- Consistency inspections ensure consistency across multiple products from the same development effort.
- For example, in a suite of office productivity applications, common functions should look and work the same whether the user is using the word processor, spreadsheet, presentation, or database program.
- Consistency inspections begin with a usability professional analyzing the interfaces to all of the products and noting the various ways that each product implements a particular user interaction or function.
- An evaluation team then meets, and using the usability analysis as a basis, negotiates and decides on the one implementation for the usability attributes of each product.

CONSISTENCY INSPECTION

- **How do I do it?**
- Form an inspection team, drawing members from each development team for all products covered in the inspection.
- The members should have the authority to negotiate for or against different design elements, and the power to change their product's design at the review meeting.
- **Have a usability professional analyze each product as to its user interface, taking care to highlight areas that are vastly different from other products.** This initial document will serve as the basis for the team's discussion during the meeting.

CONSISTENCY INSPECTION

- **How do I do it?**
- During the meeting, the team discusses the user interfaces to their products in terms of the usability professional's document, and for each element, **comes to an agreement on what that element should look and work like in all of the products.**
- Agreement must be unanimous by every member of the team, and buy-in from each product's development team (buy-in for their representative's decisions) should be procured prior to the convening of the meeting.
- Keep a running tally of the changes to be made and decisions agreed upon. **Any issues that cannot be resolved quickly should be "parked" and discussed at a later, more focused meeting.**

CONSISTENCY INSPECTION

- **When should I use this technique?**
- This technique is best used in the **early stages** of development, when initial development work has not progressed to the point where products that require extensive changes to ensure consistency will not require total overhauls.
- The ideal time for consistency inspections are when design documents for each of the individual products are almost done, and before any actual work on building the products has commenced.
- © http://jthom.best.vwh.net/usability/
- Wixon, Dennis, et. al., "Inspections and Design Reviews: Framework, History, and Reflection," in Nielsen, Jakob, and Mack, R. eds, Usability Inspection Methods, 1994, John Wiley and Sons, New York, NY. ISBN 0-471-01877-5 (hardcover)
- Nielsen, Jakob, Usability Inspection Tutorial, 1995, CHI '95 Proceedings
- http://jthom.best.vwh.net/usability/consist.htm

READINGS & WEBPAGES

- **Links to different kind of templates needed in “application/service”** production: Usability.gov .: http://www.usability.gov/templates/
- **Example of technical testing templates:**
  http://www.klariti.com/templates/Test-Plan-Template.shtml
  http://www.developsense.com/testing/TestPlanOutline.doc
- **Examples of different kinds of guidelines:**
  http://www.serco.com/usability/research/guidelines/index.asp
- **Graphical user interface guidelines examples:**
  Mac full detailed guideline
  http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html
- **Accessible Web Design**
  http://www.rnib.org.uk/digital/hints.htm
- Nielsen J. 1993. Usability Engineering, Academic Press Limited, London
- Nielsen J.2002/1994. Jacob Nielsen’s Web site for Usable Information Technology.
- **The Alertbox.**
  http://www.useit.com/alertbox/ &
  http://www.useit.com/papers/heuristic/heuristic_list.html.
- Shackel, B. 1991. Usability – context, framework, design and evaluation. In Shackel, B. and Richardson, S. (eds.). Human Factors for Informatics Usability. Cambridge University Press, Cambridge.
- **WWW-document.**
  http://jthom.best.vwh.net/usability/
- **Web Accessibility Initiative**
  WWW-document. http://www.w3.org/WAI/

REFERENCES

- Jakob Nielsen: Usability Engineering, 1993
- Irmeli Sinkkonen, Hannu Kuoppala, Jarmo Parkkinen, Raino Vastamäki: Psychology of Usability, 2006
- Usability.net
- http://usability.jameshom.com/