Usability Inspection

«UCD: User-Centered Software Development»

Prof. Dr. Cl. Müller-Birn, Institute for Computer Science, HCC.lab

June 1, 2015

Usability Inspection

Introduction

Context for today

Last week

Bringing interaction and interface design together and evaluating your designs

This week

Usability Inspection

- Action Analysis
- Heuristic Evaluation
- Cognitive Walkthrough

How to use existing insights to improve your design?

Outline

1. Action Analysis
2. Heuristic Evaluation
3. Cognitive Walkthrough

Learning goals for today

- You can describe the idea behind usability inspection and when it can be used.
- You can explain three inspection methods with their general characteristics.
- You can tell the advantages and disadvantages of using heuristic evaluation.
- You can autonomously apply usability heuristics on a practical example.

Hands-On: Order the chaos

Single person working (10 minutes)

On the next slide you will find all the methods and approaches we have discussed in this course so far. Unfortunately there are somehow mixed up.

Please bring them into the correct order based on their numbers.

Expected result:

Sequence of numbers such as, 5-7-1-…

(5) Collecting information

(2) Defining data elements by primary nouns

(7) Designing the presentation of information

(9) Designing interface elements for user's doing

(8) Defining a persona

(12) Carrying out interviews

(4) Designing interface elements for user's moving

(10) Constructing major task flows

(3) Describing a scenario

(1) Determining user groups

(11) Developing application flows to facilitate user tasks

(6) Structural design for intuitive access to content

Defining Usability Inspection

Defining usability inspection

"Usability inspection is the generic name for a set of cost effective ways of evaluating user interfaces to find usability problems. They are fairly informal methods and easy to use." (Nielson 1994)

Approaches

» Action analysis
» Heuristic evaluation
» Cognitive walkthrough

Usability Inspection

Action Analysis

What is action analysis?

Action analysis is an evaluation procedure that forces you to take a close look at the sequence of actions a user has to perform to complete a task with an interface.

We discuss only "formal" action analysis, is often called "keystroke-level analysis" in HCI. This approach is characterized by the extreme detail of the evaluation.

Phases

1. Decide what physical and mental steps a user will perform to complete one or more tasks with the interface.
2. Analyze those steps, looking for problems.

Keystroke-level model for User Performance Time

The basic idea was initially developed by (Card et al. 1983).

The goal is to make accurate predictions of the time it takes a skilled user to complete tasks.

To predict task times, the times to perform each small step of the task (physical or mental) are estimated, and those times are added.

One model is the following discussed ***keystroke level model*** (KST)

» Quantitative analysis tool for predicting performance

» The timing of different gestures has been developed in laboratory studies

» Original nomenclature consists of five values

Excursion: Performance

Time. How long does it take a user to accomplish a given set of tasks using the system?

Errors. How many errors does a user make and how serious they are?

Learning. How long does it take a novice user to learn how to use the system to do a given set of tasks?

Functionality: What range of tasks can a user do in practice with the system?

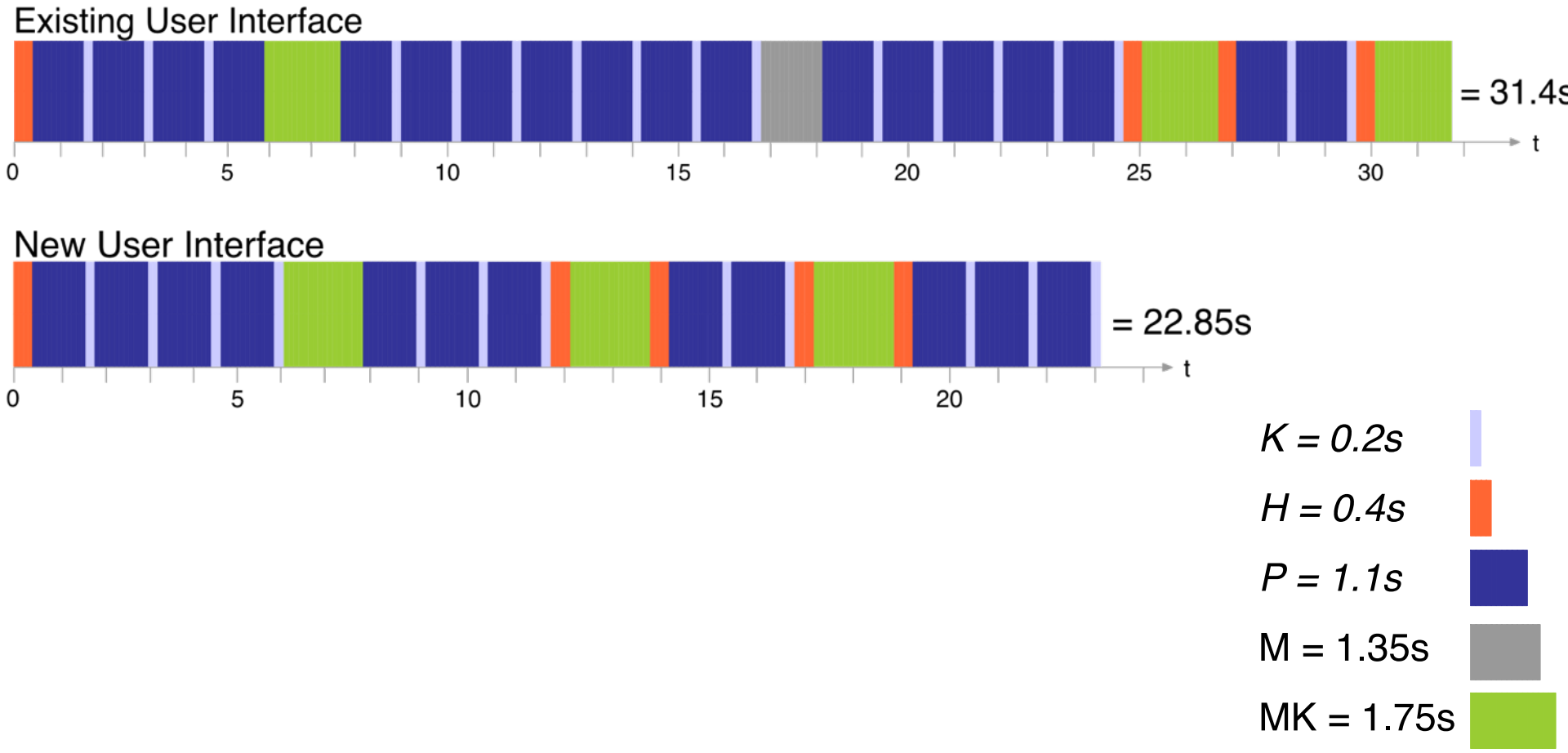
Recall: How easy is it for a user to recall how to use the system on a task that she has not done for some time?

Timing

Keying	K=0.2 sec	The time it takes to tap a key on the keyboard
Pointing	P=1.1 sec	The time it takes to point to a position on a display
Homing	H=0.4 sec	The time to move from the keyboard to mouse or vice-versa
Mentally preparing	M=1.35 sec	The time it takes a user to prepare mentally for the next step
Responding	R=f(system)	The time a user must wait for a computer to respond to input

A more differentiated description is given in: Judith Reitman Olson and Gary M. Olson,
"The growth of cognitive modeling in human- computer interaction since GOMS," Human-Computer Interaction, 5 (1990), pp. 221-265.

Example: comparison of an existing with a new designed interface



Usability Inspection

Heuristic Evaluation

What is heuristic evaluation?

» Heuristic evaluation is a systematic inspection of a user interface design for usability

» The goal is to find the usability problems in a user interface design so that they can be attend as part of an iterative design process

» It involves a small set of evaluators examine the interface and judge its compliance with recognized usability principles

What is heuristic evaluation?

» Heuristic evaluation is a **systematic inspection** of a user **interface design** for usability

» The goal is to find the **usability problems** in a user interface design so that they can be attend as part of an iterative design process

» It involves **a small set of evaluators** examine the interface and judge its compliance with recognized **usability principles**

Heuristic Evaluation

Heuristics Principles

Selected sets of heuristics principles

Nielsen, J., and Molich, R.: Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256. (1990).

Shneiderman, B.: Designing the User Interface - Strategies for Effective Human-Computer Interaction (Addisson-Wesley 1998)

Tognazzini, B.: First Principles of Interaction Design. http://www.asktog.com/basics/firstPrinciples.html (2003)

The principles behind heuristics can come from a variety of disciplines:

» Cognitive psychology
» Typography and graphic design
» Human computer interface studies
» Cultural studies

Nielsen's heuristics

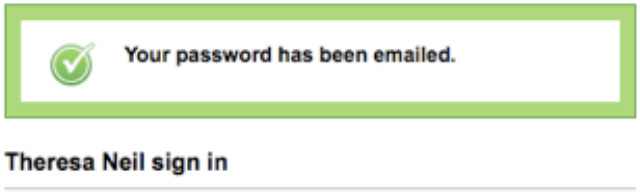
1. Simple and natural dialogue
2. Speak the user's language
3. Minimize the user's memory load
4. Consistency
5. Feedback
6. Clearly marked exits
7. Shortcuts
8. Good error messages
9. Prevent errors
10. Help and documentation

Nielsen, J., (1994). Heuristic evaluation. In J. Nielsen & R. L. Mack (Eds.), Usability inspection methods (pp. 25-62). New York: John Wiley & Sons.

Nielsen's heuristics (1)

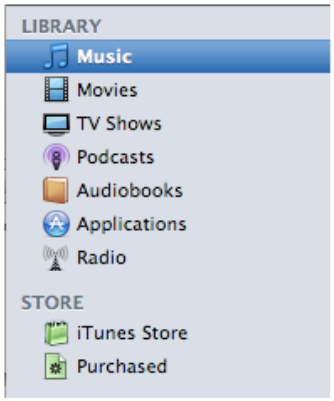
Visibility of system status

» The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.



Match between system and the real world

» The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



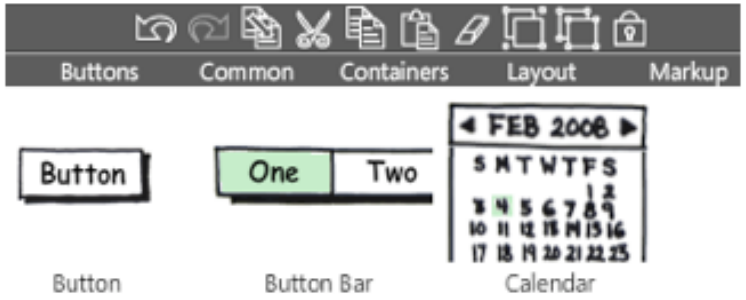
Examples taken from: http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

[Molich and Nielsen 1990; Nielsen and Molich 1990]

Nielsen's heuristics (2)

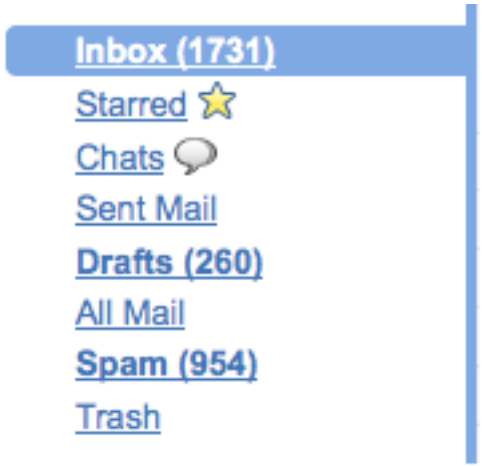
User control and freedom

» Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.



Consistency and standards

» Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.



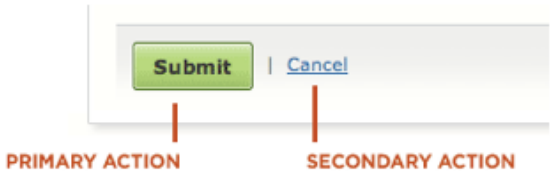
Examples taken from: http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

[Molich and Nielsen 1990; Nielsen and Molich 1990]

Nielsen's heuristics (3)

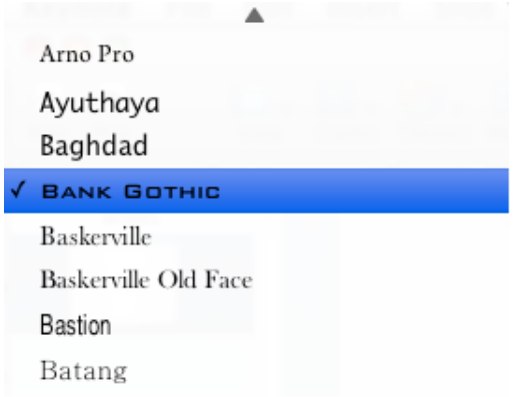
Error prevention

» Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.



Recognition rather than recall

» Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.



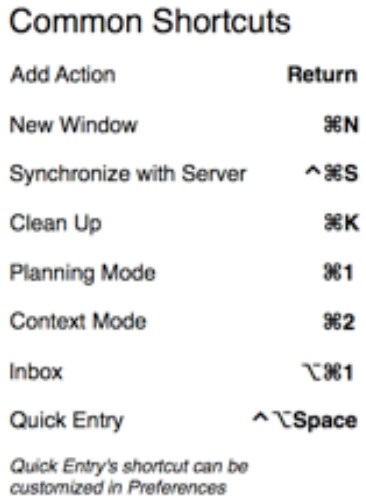
Examples taken from: http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

[Molich and Nielsen 1990; Nielsen and Molich 1990]

Nielsen's heuristics (4)

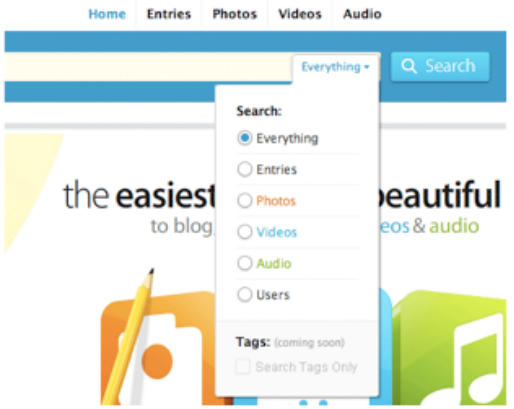
Flexibility and efficiency of use

» Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.



Aesthetic and minimalist design

» Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.



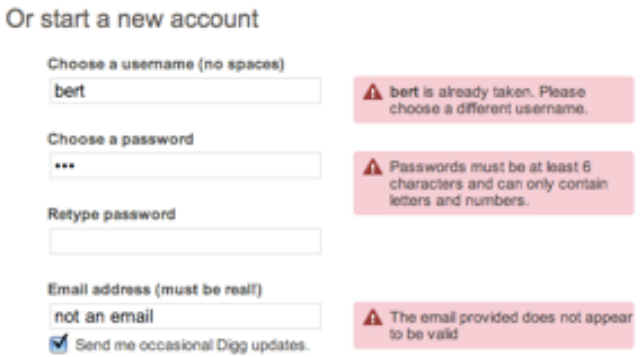
Examples taken from: http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

[Molich and Nielsen 1990; Nielsen and Molich 1990]

Nielsen's heuristics (5)

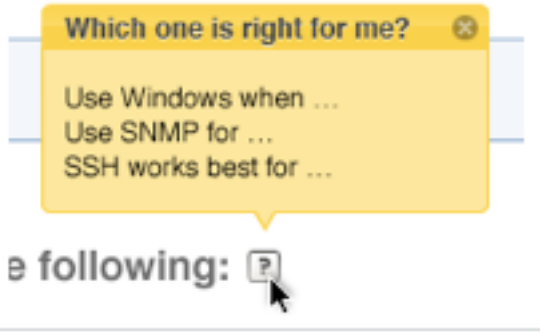
Help users recognize, diagnose, and recover from errors

» Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.



Help and documentation

» Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



Examples taken from: http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5

[Molich and Nielsen 1990; Nielsen and Molich 1990]

Shneiderman's Eight Golden Rules of Interface Design

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

Shneiderman, B. (1998). Designing the user interface: Strategies for effective human-computer interaction. Reading, MA: Addison Wesley Longman.

Tognazzini's First Principles of Interaction Design

1. Anticipation
2. Autonomy
3. Color Blindness
4. Consistency
5. Defaults
6. Efficiency of the User
7. Explorable Interfaces
8. Fitts' Law
9. Human Interface Objects
10. Latency Reduction
11. Learnability
12. Metaphors, Use of
13. Protect Users' Work
14. Readability
15. Track State
16. Visible Navigation

Tognazzini, B. (2003). First principles of interaction design. Retrieved May 1st, 2013, from http://www.asktog.com/basics/firstPrinciples.html

Hands-on
Learn about Tog's First Principles.

Hands on: Learn about Tog's First Principles (1)

Round 1 (15 minutes, individually):

» You will get one page (one page might consist of two principles or one principle is two pages long).
» Read the page and think about it. If you don't understand what you are reading, carry out some research.
» Describe it in your own words to yourself.
» Find a positive or negative example in the web to support your heuristic.

Result: After this round, you should know at least ONE principles and you should have at least one example.

Hands on: Learn about Tog's First Principles (2)

Round 2 (15 minutes, group of two):

» Find another student and explain her/him your principle and your example.
» Learn from the other student his/her principle/s.
» Now, you should know at least TWO principles and you should have at least two examples.
» The next step is to read together the additional page provided by the lecturer.
» Read the page together and discuss it.
» Find another example together.

Result: Now you should to know at least THREE heuristics.

Hands on: Learn about Tog's First Principles (3)

Round 3 (20 minutes, group of 10):

» Now all together ☺
» Build a group of ten students around one of the two flip charts. So we should have two groups, each with 10 students.
» Make sure that you have all letters from A to K represented in your group.
» Explain your principles and examples to each other, discuss them and document your insights on flipchart paper (principle's name, keywords, example).

Results: Present your results to the other groups. Are there differences?

Heuristic Evaluation

Applying usability heuristics

Pros and cons of heuristic evaluation

PROS

- Useful for evaluating a system without needing to involve users
- Identifies major problems well
- Easy to learn to do
- Can train developers to do it
- Cheap
- Gives pointers for improvements – but is not explicit
- Can be applied to interfaces in varying states of readiness, including paper prototypes

CONS

- Relies on interpretation of heuristics
- Only as good as the evaluator applying them
- Broad-based guidelines – but sometimes it is necessary to break the rules
- Can't cover every type of interface that might get built – different application areas might have very different concerns
- Doesn't involve users!

How to do a heuristic evaluation

A heuristic evaluation session for an individual evaluator typically lasts one or two hours.

Basic steps

1. The evaluator inspects the user interface thoroughly
2. The evaluator judges the interface on the basis of the heuristics
3. The evaluator makes a list of the usability problems found
4. The evaluator explain and justifies each problem with heuristics

Hints for doing a “good” heuristic evaluation

» Your evaluation should be grounded in known usability guidelines. You should justify each problem you list by appealing to a heuristic, and explaining how the heuristic is violated.

» List every problem you find. For example, if a button has several problems with it – inconsistent placement, bad color combination, bad information scent – then each of those problems should be listed separately.

» Inspect the interface at least twice.

Formal heuristic evaluation process

1. **Preparation:** Decide which aspects of a product and what tasks you want to review. Create a list of representative tasks for the application or component you are evaluating. Decide which heuristics will be used.
2. **Training:** If possible select a team of three to five evaluators and give them some basic training on the principles and process.
3. **Evaluation:** Ask each evaluator to perform the representative tasks individually and list where the product violates one or more heuristics (focus on problems).
4. **Rating:** Ask evaluators to prioritize all problems found (not just their own). Take the mean of evaluators' ratings.
5. **Debriefing:** The design team and the evaluators meet again to discuss the results. Categorize and report the findings so they can be presented effectively.

Severity ratings

Factors that contribute to the severity of a problem:

- » Frequency of occurrence: common or rare
- » Impact on users: easy or hard to overcome
- » Persistence: does it need to be overcome once or repeatedly

Severity scale

1. Cosmetic: need not be fixed
2. Minor: Needs fixing but low priority
3. Major: Needs fixing but high priority
4. Catastrophic: Imperative to fix

Possible adaptions/extensions of heuristic evaluation

» Use more than one evaluator (the sweet spot for cost-benefit is 3-5 evaluators, cp. Nielson, 1992)

» Alternate heuristic evaluations with user testing within an iterative design cycle

» The observer (if there is one) can help the evaluator if she gets stuck in a confusing interface (as long as the usability problems is known)

Heuristic evaluation is not user testing

- Evaluator is not the user either
  - Maybe closer to being a typical user than you are, though
- Analogy: code inspection vs. testing
- HE finds problems that UT often misses
  - inconsistent fonts
  - Fitts's law problems
- But UT is the gold standard for usability

Writing good heuristic evaluations

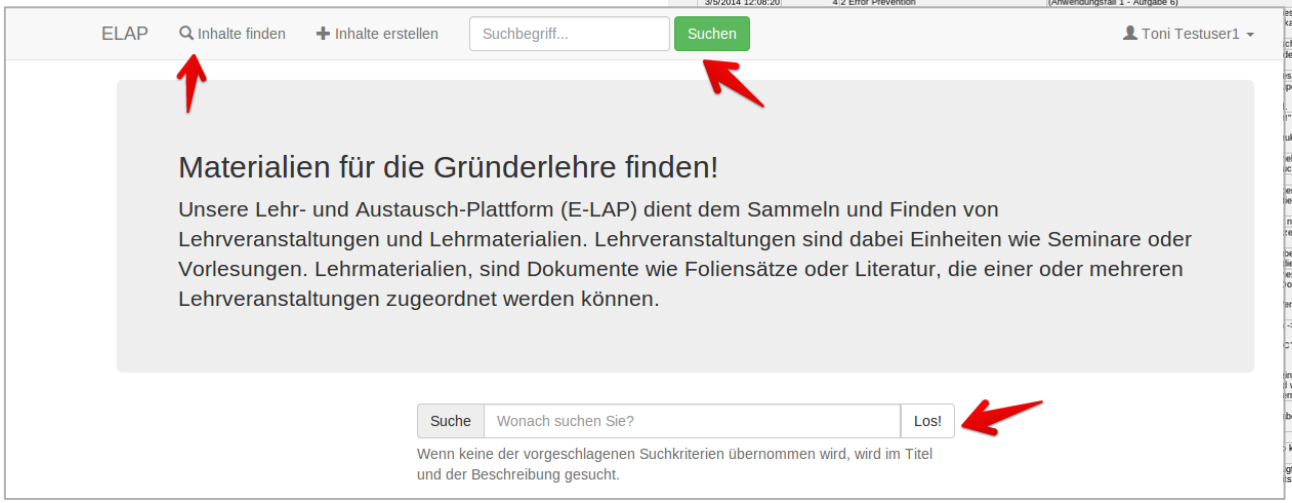
- Heuristic evaluation must communicate well to developers and managers
- Include positive comments as well as criticisms
  - “Good: Toolbar icons are simple, with good contrast and few colors (minimalist design)”
- Be tactful
  - Not: “the menu organization is a complete mess”
  - Better: “menus are not organized by function”
- Be specific
  - Not: “text is unreadable”
  - Better: “text is too small, and has poor contrast (black text on dark green background)”

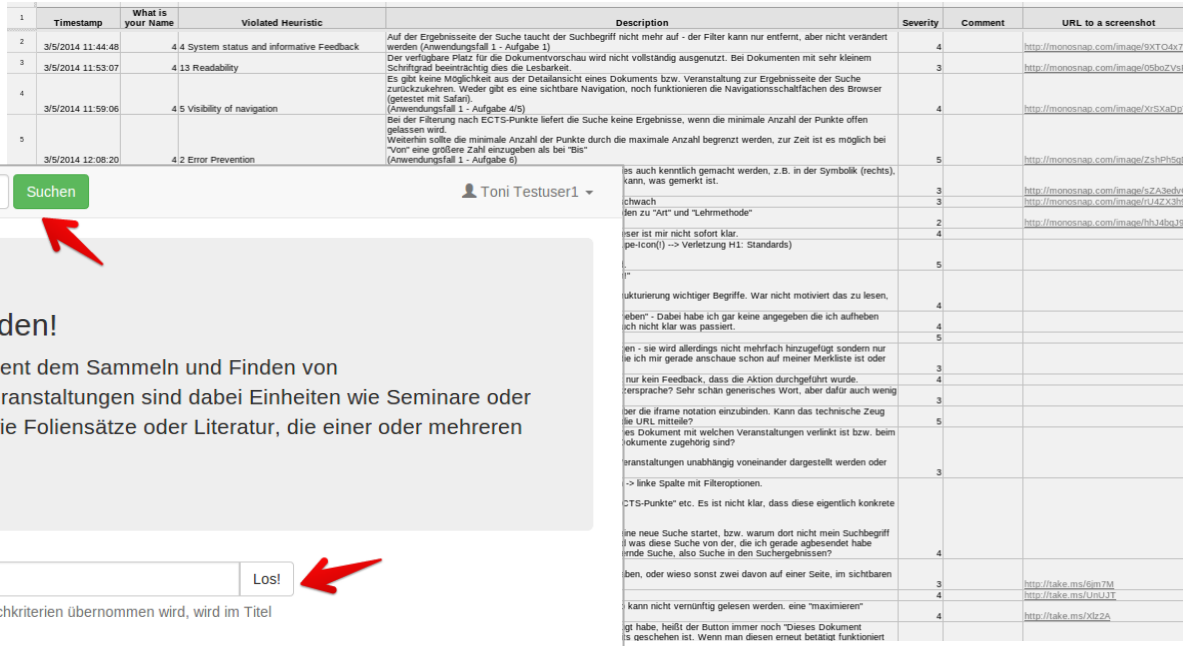
Suggested Report Format

What to include:

- Problem
- Heuristic
- Description
- Severity
- Recommendation (if any)
- Screenshot (if helpful)

Example





	Timestamp	What is your Name	Violated Heuristic	Description	Severity	Comment	URL to a screenshot
2	3/5/2014 11:44:48	4	4 System status and informative Feedback	Auf der Ergebnisseite der Suche taucht der Suchbegriff nicht mehr auf - der Filter kann nur entfernt, aber nicht verändert werden (Anwendungsfall 1 - Aufgabe 1)	4		http://monosnap.com/image/9XTO4x7
3	3/5/2014 11:53:07	4	13 Readability	Der verfügbare Platz für die Dokumentvorschau wird nicht vollständig ausgenutzt. Bei Dokumenten mit sehr kleinem Schriftgrad beeinträchtigt dies die Lesbarkeit.	3		http://monosnap.com/image/05boZVs
4	3/5/2014 11:59:06	4	5 Visibility of navigation	Es gibt keine Möglichkeit aus der Detailansicht eines Dokuments bzw. Veranstaltung zur Ergebnisseite der Suche zurückzukehren. Weder gibt es eine sichtbare Navigation, noch funktionieren die Navigationsschaltfächen des Browser (getestet mit Safari). (Anwendungsfall 1 - Aufgabe 4/5)	4		http://monosnap.com/image/XrSXaDp
5	3/5/2014 12:08:20	4	2 Error Prevention	Bei der Filterung nach ECTS-Punkte liefert die Suche keine Ergebnisse, wenn die minimale Anzahl der Punkte offen gelassen wird. Weiterhin sollte die minimale Anzahl der Punkte durch die maximale Anzahl begrenzt werden, zur Zeit ist es möglich bei "Von" eine größere Zahl einzugeben als bei "Bis" (Anwendungsfall 1 - Aufgabe 6)	5		http://monosnap.com/image/ZshPh5q

Problem: Suche

Heuristik: Consistency/Standards

Beschreibung: Suchmöglichkeiten tauchen mehrfach auf und für die Evaluatoren war nicht einsichtig, worin die Unterschiede bestehen. Es kann über die Menüleiste am oberen Rand gesucht werden und über die zentrale Suchleiste in der Mitte der Seite, zudem sehen beide Suchen nicht gleich auch, obwohl sie die gleichen Funktionen bereitstellen

Severity: 5

Result


ELAP
Entrepreneurship Lehr- und Austauschplattform
ALLE INHALTE
HILFE
ANMELDEN
INHALTE ZUM THEMA GRÜNDUNGSLEHRE FINDEN!
z.B. Geschäftsmodell oder Design Thinking
Alle Inhalte anzeigen
Businessplan
Business Model Canvas
Finanzierung
Geschäftsmodelle
Marketing
Produkt und Dienstleistung
Rechtsformen
Praxisgründung
Faltin
Evaluation
Gründergarage
Marktanalyse
Elevator-Pitch
Organisation
Ideenpitches
Ideengenerierung mit der Empathy Map (Funpreneur Wettbewerb)
SUCHEN UND FINDEN
Suchen Sie Lehrveranstaltungen, wie Seminare oder Vorlesungen und suchen Sie nach Lehrmaterialien in Form von Foliensätzen, Video, Audio oder Text.
Suchen und Finden
INHALTE BEREITSTELLEN
Sie können ihre eigenen Lehrveranstaltungen und Lehrmaterialien anderen Nutzern zur Verfügung stellen.
Inhalte bereitstellen
ÜBER E-LAP
Unsere Lehr- und Austausch-Plattform (E-LAP) dient dem Sammeln und Finden von Lehrveranstaltungen und Lehrmaterialien.
Über E-LAP

Usability Inspection

Cognitive Walkthroughs

What is a cognitive walkthrough?

The cognitive walkthrough is a formalized way of imagining people's thoughts and actions when they use an interface for the first time. It is focused on learnability.

Where heuristic evaluation is focusing on individual elements in the interface, a cognitive walkthrough focuses on individual actions in the sequence, asking a number of questions about the learnability of each action.

For each action the evaluator asks:

- » Will user know what sub-goal they want to archive?
- » Will user find the action in the interface?
- » Will user recognize that it accomplishes the sub-goal?
- » Will user understand the feedback of the action?

Wrapping up

We talked about difference between user testing and usability inspection.

We discussed three methods: action analysis, heuristic evaluation, cognitive walkthrough.

You know a lot of heuristics now and you are able to plan and carry out a heuristic evaluation (because this is your homework).

References

Jakob Nielsen. 1994. Usability inspection methods. In Conference Companion on Human Factors in Computing Systems (CHI '94), Catherine Plaisant (Ed.). ACM, New York, NY, USA, 413-414.

http://www.interaction-design.org/encyclopedia/usability_evaluation.html

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33 , 3 (March), 338-348.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.

Card, S., Moran, T., and Newell, A. "The Psychology of Human-Computer Interaction." Hillsdale, New Jersey: Lawrence Erlbaum, 1983.