ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

# The evaluation
# Inspection

Fabio Vitali

## The evaluation

The evaluation can take place:

- Internally to the development team, while the product is being developed. This is called inspection of the design

- Externally to the development team, with the participation of potential external users. This is called testing.

## What to evaluate

- Sketches
  - Crude drawings on A4 papers are shown to the user
  - The test assistant manually and openly switches from one drawing to the next according to the user's indications
- Mock-Up on a computer
  - A wireframe of the application is shown to the user
  - The wireframe tool or the test assistant manually and openly switches from one sketch to the next
- Wizard of Oz
  - A high-quality interface of the application is shown
  - Behind the scenes, a human provides the answers and the switches from one screen to the next.
- A prototype
  - A partially working system with finalized interface is shown.
  - Only the parts that are working are tested
- The working system

## Basic rule of thumb of evaluations

*The earlier you evaluate the system, the less precise are the results, but the less expensive it is to fix the problems.*

***Therefore:***

- Evaluate early, evaluate often;
- Remove big errors early (before they cost too much to fix);
- Leave later evaluations for lesser (e.g. cosmetic) issues and for formal (i.e., contractual) assessments.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

## The inspection

## Inspection

The inspection phase takes place within the design team. For this reason, this is a cheap tool (but also very inaccurate) for the evaluation of the usability of a system.

In the inspection phase there are three relevant activities:

- Cognitive walkthrough: a fictional and step by step execution of a task, and the empirical evaluation of the likeliness of the fiction
- Action analysis: a quantitative analysis of specific actions that must be performed to play an action.
- Heuristic analysis (or guidelines application): the evaluation of interfaces based on common-sense rules derived from experience

## Cognitive walkthrough (1/2)

A cognitive walkthrough is a formalized way of imagining thoughts and actions of users when they use an interface to perform for the first time a task.

It takes the system, a prototype or even a series of drawings to try. You select a task to perform with that interface, and tell a credible story about each action that the user must execute to complete the task.

The story is credible if you can motivate each action of the user relying on general knowledge of the assumed user and on the indications and feedback provided by the interface. If you cannot tell a believable story, there is an interface problem.

Objective: To determine the plausibility of the usability of the interface for the chosen user segment.

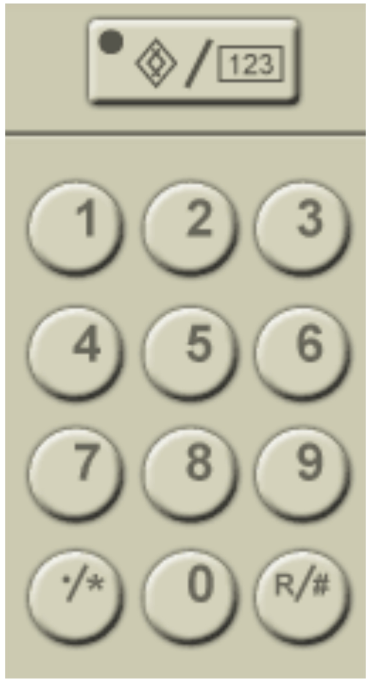
## Cognitive walkthrough (2/2)

A CW needs four ingredients:

- A description or a prototype of the interface, as detailed as possible.
- The description of a task, possibly one of the tasks described as representative in the task-based or goal-based design
- A complete and written list of the actions necessary to complete the task
  - It is sometimes called Happy Path, because it represents the ideal sequence to performing the task
- A clear description of the User and his/her skills and expectations

Based on these ingredients, task by task, you need to build a story and to evaluate its credibility.

## CW: an example (1)



Let's assess a photocopier's interface.

You are given the drawing of a numeric keypad, a "copy" button and a soft button on the back of the machine for turning it on and off. You are also told that the power turns off after five minutes of inactivity.

The task is to copy a single page, and the user is a newly hired secretary. The story we tell is:

"The secretary has to make a copy, and she knows that the copier must be turned on, so she presses the power button. She puts the sheet of paper inside the copier and presses the copy button."

## CW: an example (2)

This story is not very credible:

- how does the secretary know that the copier is turned off?
- How does she know where is the power button?
- How does she know how to insert the sheet of paper?
- Are we sure she knows that the button above the keypad means "copy"?

Warning: we must NOT evaluate the interface, but the credibility of a story ***using the interface***. This may lead to revisions both to the interface and to the story

Let’s add, for example, a display that indicates when the copier is ready, let’s add a drawing on the front to indicate how to insert the sheets, let’s move in a visible position the main switch, and let’s change the button to a "Copy" button, and repeat the experiment. And so on.

## CW: an example (3)

The CW can discover different types of problems:

- The designer assumptions on users 'reasoning ("why would the user think that the copier must be switched on?")
- Commands obvious to the designer but not obvious to the user ("the user knows that she wants to turn on the machine, but does she know where to find the switch?")
- Problems with labels and prompt ("How does she know how to insert the paper, and are we sure that she understands the icon?")
- Feedback problems ("How does she know if the copier is turned on or off?")

## CW: common errors

There are two common mistakes in the design of a CW:

- Confusing the list of actions with the walkthrough itself. The sense of CW is to credibly tell how the user performs the optimal actions to complete the task, not to describe it while she is discovering these actions.
- Confusing the CW with the actual user test: the CW identifies a class of problems that a test with 5-10 users might not identify, but the actual test with the users identifies real aspects that cannot be discovered with the CW alone.

## CW – differences with scenarios

A cognitive walkthrough is similar in many ways to a scenario or a use case, but there are major differences:

- A scenario does not include a prototype of the interface nor the description of the actions (the happy path) to carry out a task
- A scenario aims at building the interface and the happy path, not at evaluating it
- Scenarios are by construction believable, while CW become acceptable after they have become believable

## CW - Self-evaluation

Questions to ask during the self-assessment of the cognitive walkthrough results

- Is it realistic that the character will try to do this specific action?
- The control that commands this action is available?
- Is there an obvious link between control and action?
- Is feedback appropriate?

## Action analysis

The action analysis is an evaluation process that closely examines the sequence of actions to be performed to complete a task.

There are two types:

- Formal action analysis (keystroke-level analysis) is characterized by an extreme detail in the description of the actions. It can predict with a margin of 20% the actual time of completion of the task, the average time to learn an interface, the ratio between actions and errors. Unfortunately, it is very complicated and lengthy to be carried out.
- Informal action analysis (back-of-the-envelope analysis) is rather less precise and much easyer to carry out. It can highlight excessive complications, excessively long execution times, or blatant interface issues.

## Formal action analysis (1)

The formal approach to the analysis of actions is used to make *accurate predictions* of the time spent by an experienced user in performing a task.

To do this, we need to estimate the times to perform each step (physical and mental) of the task and sum them together.

The typical step is the pression of a key, so this is also called *keystroke-level analysis*.

***Objective***: specific calculation of the average times of use of a system (or, more often, of a widget).

## Formal action analysis (2)

The estimate of the time of each action is derived from a table obtained by testing hundreds of users, thousands of individual actions, in thousands of situations, and then averaging.

If a control is not described in this table, you either approximate it with something like it, or you have to run similar tests on the new control.

To obtain the time necessary to perform a task, therefore, a top-down approach must be adopted in the description of the optimal path, and then associate the appropriate times to each individual action.

# Formal action analysis (3)

A formal analysis of a complex interface is a daunting task. 10 minutes of an action may require the description of one thousand actions and the corresponding timings.

In addition, the description of the task and the actions of the users are discretion of the evaluator, and therefore there may be significant discrepancies between different analyses. For this reason, the action analysis is useful only in a few special circumstances:

- To examine very specific aspects of an interface (e.g., to examine the performance of a new widget of a GUI)
- To examine very structured and controlled tasks (e.g.: to analyze the workload of a telephone operator of an online helpdesk)

## GOMS: Action analysis for performance

Cognitive models essentially describe:

- Competence (knowledge of sequences of atomic behavior), or
- Performance (speed of execution, but only for routine tasks)

Action analysis is a performance model to describe goals and tasks.

GOMS is the earliest model of Action Analysis

- Based on
  - Goals
  - Operators
  - Methods
  - Selection rules
- Input: a detailed description of the interface and the tasks
- Output: quantitative and qualitative measures

## Average times for interface actions (1)

- [from J. Reitman GM Olson and Olson, "The growth of cognitive modeling in Human- computer interaction since GOMS," Human-Computer Interaction, 5 (1990), pp. 221-265]

### VISUAL PERCEPTION

| | |
|---|---|
| ◆ Respond to a brief light | 0.10 s. |
| • Varies with intensity, from .05 second for a light. | |
| ◆ Recognize to 6-letter word | 0.34 s. |
| ◆ Move eyes to new location on the main screen | 0.23 s. |

## Average times for interface actions (2)

PHYSICAL MOVEMENTS

- Enter one keystroke on a keyboard: 0.28 s.
  - For the first time skilled typists doing transcription, to .2 second for an average 60-wpm typist. Random sequences, formulas, and commands take longer than plain text.
- Use mouse to point at object on screen 1.50 s.
  - May be slightly lower - but still at least 1 second – a small screen and menu. Increases with larger screens, smaller objects.
- Move hand to pointing device or function key 0.30 s.
  - Ranges from .21 second for cursor keys to .36 second for a mouse.

## Average times for interface actions (3)

MENTAL ACTIONS

- Retrieve a simple item from long-term memory 1.20 s.
  - A typical item might be a command abbreviation ("dir"). Time is halved if the same item needs to be retrieved again immediately.
- Learn a single "step" in a procedure 25.00 s.
  - May be less under some circumstances, but most research shows 10 to 15 seconds as a minimum. None of these figures includes the time needed to get started in a training situation.
- Execute a mental "step" 0.07 s.
  - Ranges from .05 to .1 second, depending on what kind of mental step is being performed.
- Choose among methods 1.20 s.
  - Ranges from .06 to at least 1.8 seconds, depending on complexity of factors influencing the decision.

# KLM (Keystroke-Level Model)

One of the many evolutions of GOMS still in wide practice today.

It is used to predict or estimate how long it will take an *experienced user* to complete a *routine task* with a software tool.

The model is composed of six operators:

- K: keystroke or button press. The number of times keyboard buttons and mouse buttons are pressed. Keys, not characters: capital A is two K actions, pressing Shift and and pressing A.
- P: pointing with a mouse. Moving the mouse is a separate action from clicking the mouse (which is K).
- H: homing: moving the hands and fingers on the keyboard or other device (also: positioning). Includes moving from e.g., keyboard to mouse or moving the hands on a touch screen.
- D: manually drawing. Not frequently used.
- M: mental preparation. Time needed for thinking, planning or decision making.
- R: system response time, or wait time (also: W)

# An example using KLM

Each element has a standardized time associated with it. For our purpose, K for an average typist (40 wpm) is 0.28 seconds, B is 0.1 seconds, P is 1.1 seconds, H is 0.4 seconds, and M is 1.35 seconds.

We must identify each step in the task, separate all individual actions, and assign the time of each action, and then sum up all times

E.g.: Enter a street address (*Via Irnerio 36*) into a text field:

- Initiate the action (M)
- Find on the screen the correct text field (M)
- bring the mouse pointer to the correct field (P)
- Press mouse button (B)
- Release mouse button (B)
- Move hands from mouse to keyboard (H)
- Type “Via Irnerio 36” (14 letters, two of which are numbers and two are uppercase: 16K) – assuming numbers are the lowercase keys and remembering that uppercase letters require 2K each.

Total time = 2M + 1P + 2B + 1H + 16K = 8.88s

## Pros and cons of all types of formal Keystroke-Level Analysis

Pros:

- Practical – no need to verify data against real users.
- General - No need for knowledge of psychological models.
- Early usage - No need for working prototypes.

Cons:

- Only returning time as fundamental performance metric.
- Designed for expert users in routine tasks and not making errors;
- No room for learning, exploration, hesitation, mistakes and slips;
- Very simplified model for mental operations (M)
- Producing the full list of actions can be time-consuming for longer tasks.
- The longer the task, the more impactful is the number of mental operations (M) with respect to physical actions, and therefore more imprecise is the computed result.

Mainly useful for short and well-known tasks.

## Informal action analysis (1)

The informal analysis ignores the micro-detail and focuses on the big picture, listing a “natural” series of actions and evaluating them globally.

Instead of "taking your hand from the keyboard and grab the mouse", the actions described here are of the type: "choose the option X from menu Y".

Also called: "back-of-the-envelope action analysis"

Objective: heuristic determination of the steps with greatest weight (in terms of time and number of atomic actions) in the execution of a task, and therefore the potential sources of excessive complexity, loss of time, disorientation.

## Informal action analysis (2)

The emphasis is not on the tenth of a second to evaluate the performance of a widget, but on evaluating responses to questions such as:

- Can I execute a simple task in a simple manner?
- Can I perform a frequent task quickly?
- How many steps and facts do I need to learn before I can perform a task?
- Did we describe each step in the documentation?

## Informal action analysis (3)

Without being as precise as in formal action analysis, the informal action analysis is more robust and less subject to inaccuracies, and can be used to:

- Verify that the execution of a task does not require comparable times to doing it by hand, on paper, with a different application, etc.
- Decide whether the addition of a feature will or will not complicate the rest of the interface.
- Decide whether to add multiple ways to accomplish a task.
- Check which operations may end generating an error, and how serious this error may end up being (in terms of the time needed to fix it)

## Heuristic analysis

Both the cognitive walkthrough and the action analysis are task-oriented evaluations. User testing, described forward, is also task-oriented.

Task-oriented assessments have strengths and weaknesses:

- Appropriateness: They evaluate the characteristics of a system within a credible job flow and driven by goasl independent of the interface.
- Coverage: they let us evaluate only a few tasks, ignoring many of the others.
- Inter-task interactions: they let us evaluate how the system behaves when the user is performing many actions at the same time.

Objective: To verify the system's adherence to the guidelines identified for the project and to justify any deviation (or to modify the design).

## Heuristic evaluation

An inspection tool (thus executable by the development team) to evaluate the usability of a system regardless of the tasks it is designed for (it is therefore a domain-independent evaluation)

Comparison of the application with general, universally recognized principles.

They are based on the fundamental principle of external consistency, applied both in a positive or negative sense:

- If in other applications choice X was positive, it is probably positive also in this system.
- If in other applications choice Y was negative, it is probably negative also in this system.

This is called heuristic assessment as it provides guidelines for the discovery (heuristic) of usability problems.

## Guidelines

*<sarcasm>*
*Guidelines are beautiful is because they are so many and varied.*
*</sarcasm>*

**General guidelines** (some examples)

- The 10 heuristics of Nielsen and Molich (1994)
- The guidelines of UserFocus.co.uk (commercial, UK, 2014)

**Governmental guidelines** (some examples)

- US Research-based Web Design and Visibility Guidelines (2006)
- EU Europa Web Guide, *Rules and guidelines that apply to European Commission websites, covering editorial, legal, technical, visual and contractual aspects* (undated, current)
- EU Usability guidelines for websites and products of statistical organisations (2020)

## The 10 heuristics of Nielsen and Molich (1)

1. Visibility of system status

   The system should always keep the user informed about what happens, through appropriate feedback provided within a reasonable time

2. Match between the system and the real world

   The system should speak the user's language, with words, phrases and concepts familiar to the user rather than system terms. It must follow conventions of the real world, and make information appear in a natural and logical order.

3. User control and freedom

   Since the user often chooses system functions by mistake, he needs clearly marked "emergency exits" to leave the unwanted state without having to go through a complex dialogue. Support undo and redo.

## The 10 heuristics of Nielsen and Molich (2)

4. Consistency and standards

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

5. 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.

6. 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.

# The 10 heuristics of Nielsen and Molich (3)

7. Flexibility and efficiency of use

   Accelerators — unseen by the novice user — 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.

8. Aesthetics 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.

9. 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.

10. 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.

## The guidelines of Userfocus.co.uk (1)

A private company that gives advice on the design and evaluation of the usability of applications and websites.

On their web site you can find articles, (freely downloadable) books and the guidelines of the evaluation of usability.

247 guidelines organized in 9 chapters. There is a convenient Excel to fill-in with appropriate values during the assessment.

Automatically generates a chart of the overall usability of the application or website.

***http://www.userfocus.co.uk/resources/guidelines.html***

## The chapters of Userfocus.co.uk (1)

Usability home page

- 20 guidelines for evaluating the usability of a home page.

Task orientation

- 44 guidelines to evaluate how user tasks are supported

Navigation and IA:

- 29 guidelines to evaluate the navigation and information architecture

Forms and data entry:

- 23 guidelines for evaluating forms and data entry.

Trust and credibility:

- 13 guidelines for evaluating a credibility and trust.

## The chapters of Userfocus.co.uk (2)

Writing and content quality:

- 23 guidelines for evaluating the quality of texts and content.

Page layout and visual design:

- 38 guidelines for evaluating the page layout and the quality of the graphics.

Search usability:

- 20 guidelines for evaluating the search engine.

Help, feedback and error tolerance:

- 37 guidelines to help evaluate, feedback and tolerance to errors.

## USA: *Research-Based Web Design & Usability Guidelines*

Born for the Health and Human Service Department of the US Government in 2003, then in 2006 adopted and standardized across all US government web sites.

**https://www.usability.gov/sites/default/files/documents/guidelines_book.pdf**

18 chapters: the first and the last detailing the process, the others emphasizing one of the aspects of the design. 209 Guidelines overall.

1. Design process and evaluation (11 gls)
2. Optimizing User Experience (16 gls)
3. Accessibility (13 gls)
4. Hardware & Software (5 gls)
5. The Homepage (9 gls)
6. Page Layout (13 gls)
7. Navigation (12 gls)
8. Scrolling & Paging (5 gls)
9. Heading, Titles & Labels (8 gls)
10. Links (14 gls)
11. Text Appearance (11 gls)
12. Lists (9 gls)
13. Screen-based controls (widgets) (25 gls)
14. Graphics, Images & Multimedia (16 gls)
15. Writing Web Content (11 gls)
16. Content Organization (9 gls)
17. Search (9 gls)
18. Usability testing (13 gls)

## EU: *Europa Web Guide*

Rules and guidelines that apply to European Commission websites, covering editorial, legal, technical, visual and contractual aspects.

https://commission.europa.eu/resources-partners/europa-web-guide_en

10 rules and 4 guidelines blocks including a variety of aspects not all of which related to usability.

**Rules**

1. EU Domain and Subdomains
2. Branding
3. URL structure
4. Site categories
5. Visual identity
6. Accessibility
7. Corporate solutions
8. Data protection
9. Intellectual Property Rights
10. Archiving

**Guidelines**

1. Architecture and Navigation
2. Design recommendations
3. Content Guidelines
4. Search Engine Optimization

## EU: *Europa Web Guide guidelines*

1. Architecture and Navigation
   - Structuring Information for findability
   - Task-based Information Architecture
   - Navigation system
   - Names and labels
   - Testing methods: *Card sorting, Tree testing*
   - Role of search
   - Vocabularies and taxonomies
2. Design recommendations
   - Users first
   - Evidence over opinion
   - Prioritize task completion
   - Be inclusive
   - Design for multiculturalism
   - Design effectively
   - Provide no more than needed
   - Promote brand approach
3. Content Guidelines
   - Page layout and components
   - Types of content: *news, events, audiovisual, publications, funding*
   - Editorial style and policy
   - Web writing guidelines (10 sub-guidelines)
   - Language coverage policy
4. Search Engine Optimization
   - How to optimize content
   - Migrating a website with minimal ranking loss
   - Optimising files like PDF, PPT, DOC and XLS
   - Linking strategies (4 sub-guidelines)
   - Search Engine marketing

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

## EU: Usability guidelines for websites and products of statistical organisations

Best practices and recommendations for the design of websites and other online tools that are used for the dissemination of official statistics.

https://cros-legacy.ec.europa.eu/system/files/usability_guidelines_for_websites_and_products_of_statistical_organisations.pdf

Meant for EU offices publishing statistical tables of EU-wide impact, but contain many useful advice for general Public Administration information websites.

1. What is User Experience?
2. Layout
   1. Screen Real Estate
   2. Use of colour
   3. Typography and readability
   4. Images
   5. Icons and labels
   6. Affordances
3. Structure and navigation
   1. Cross-links
   2. Hyperlinks
   3. Navigation menu
   4. Breadcrumbs
4. Search and filter
5. Design components
   1. Buttons
   2. Headers
   3. Accordions
   4. Carousels
   5. Tabs
   6. Drop-down menus
   7. Tooltips
   8. Help and documentation
   9. In-page scrolling
6. Plain language
7. Web accessibility
8. Beyond functionality
   1. User support
   2. Providing feedback to users
   3. Collecting feedback from users
   4. Sharing on social media
   5. Layering of functionality
9. Designing for mobile screens
   1. General guidelines
   2. Visualizing data on a mobile device

## Bibliography

- Jesse Garrett, The elements of user experience, New Riders, 2011
- Philip Harris, Data Driven Design, K & R publications, 2013
- A. Cooper, R. Reimann, D. Cronin About Face 3, Wiley, 2009
- Alan Cooper, The inmates are running the asylum, SAMS, 2004
- T. Tullis, Albert B., Measuring the User Experience, Morgan Kaufmann, 2013
- Susan Weinschenk, 100 things every designer needs to know about people, New Riders, 2011
- Usability Book of Knowledge, http://www.usabilitybook.org/