Human Computer Interaction

USABILITY

By: Nguyễn Công Hoan

Reference

- DonaldNorman, **The Design of Everyday Things, MITPress, 23 Dec 2013**
- **Tutorial Teaching** of Prof. Dr. Keith Andrews, Graz University of Technology

Agenda

- Usability Engineering
- Usability Benchmarking
- Usability Inspection Methods
- UsabilityTesting Methods
- Usability in Practice

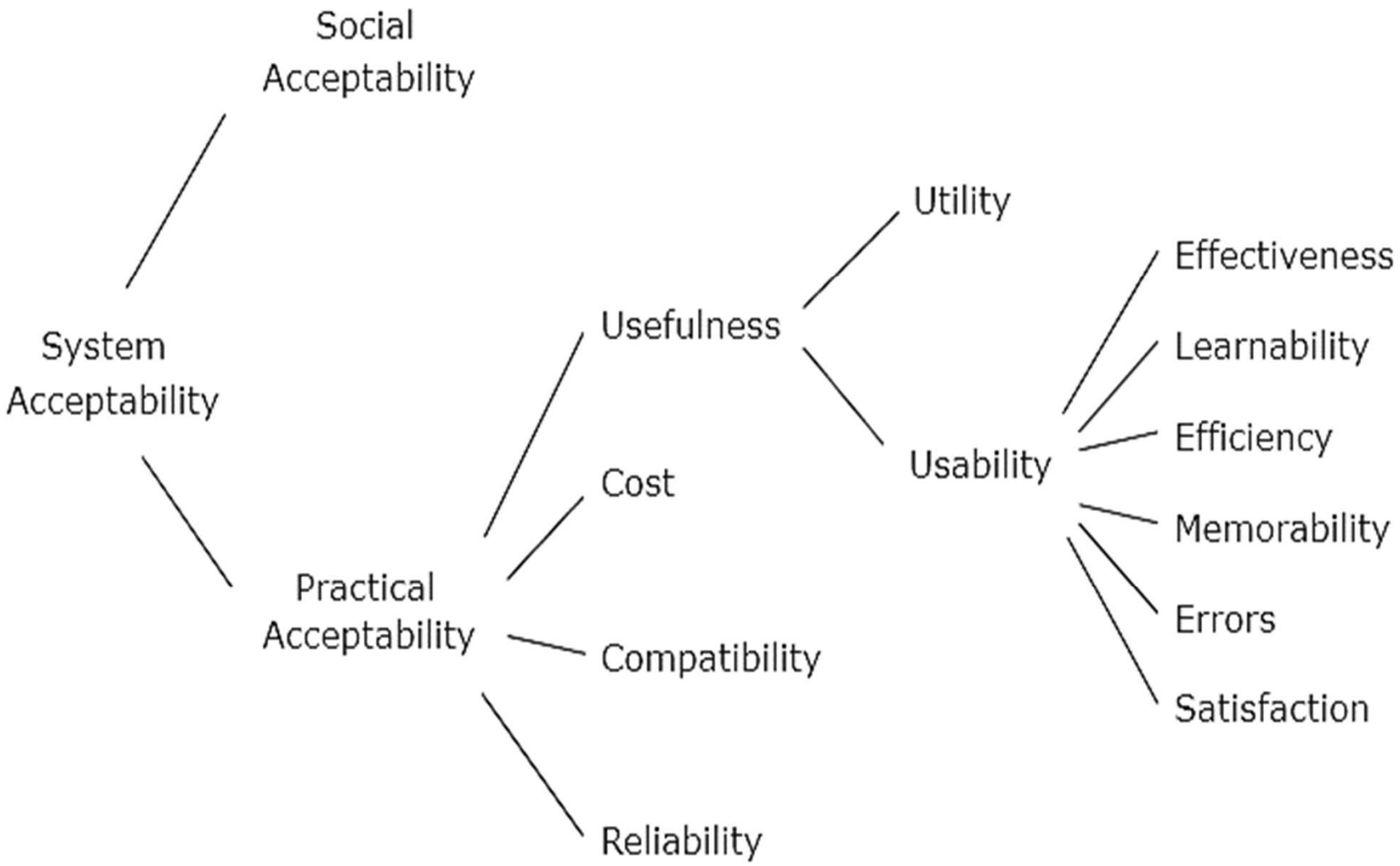
De ning Usability

The ISO de nes usability as “the extent to which a product can be used by *speci ed users* to achieve *speci ed goals* with *e ectiveness*, *e ciency* and *satisfaction* in a *speci ed context of use*.” [ISO, 1998].

The three measurable usability attributes de ned by ISO [1998] are:

- **E ectiveness**: accuracy and completeness with which users achieve speci ed goals.
- **E ciency**: resources expended in relation to the accuracy and completeness with which users achieve goals.
- **Satisfaction**: freedom from discomfort, and positive attitudes towards the use of the product.

Usability in Context



A model of the attributes of system acceptability, based on Figure1 of [Nielsen,1993b].

Six Usability Attributes

Combining the three ISO usability attributes with Nielsen's ve usability attributes, leads to the following six usability attributes:

1. **E ectiveness**: completeness with which users achieve their goal.

2. **Learnability**: ease of learning for novice users.

3. **E ciency**: steady-state performance of expert users.

4. **Memorability**: ease of using system intermittently for casual users.

5. **Errors**: error rate for minor and catastrophic errors.

6. **Satisfaction**: how satisfying a system is to use, from user's point of view.

Measuring Usability Attributes

- **E ectiveness**: decide on de nition of success. For example, number of substitution words spotted in a text, or binary measure of success (order completed or not).
- **Learnability**: pick novice users of system, measure time to perform certain tasks. Distinguish between no/some general computer experience.
- **E ciency**: decide de nition of expertise, get sample expert users (di cult), measure time to perform typical tasks.
- **Memorability**: get sample casual users (away from system for certain time), measure time to perform typical tasks.
- **Errors**: count minor and catastrophic errors made by users while performing some speci ed task. For example, number of deviations from optimal click path.
- **Satisfaction**: ask users' subjective opinion (questionnaire), after trying system for real task.

Usability Evaluation

There are four types of evaluation, according to the purpose of the evaluation:

- **Exploratory** - how is it (or will it be) used?
- **Predictive** - estimating how good it will be.
- **Formative** - how can it be made better?
- **Summative** - how good is it?

Exploratory Evaluation

Explores current usage and the potential design space for new designs.

- Done before interface development.
- Learn which software is used, how often, and what for.
- Collect usage data – statistical summaries and observations of usage.

Predictive Evaluation

Estimates the overall quality of an interface (like a summative evaluation, but a prediction made in advance).

- Done once a design has been done, but before implementation proceeds.

Formative Evaluation

Informs the design process and helps improve an interface during design.

- Done during interface development.
- Learn why something went wrong, not just that it went wrong.
- Collect process data – qualitative observations of what happened and why.
- Formative evaluation methods are “ nd and x” methods, and typically produce as output a list of problems found.

Summative Evaluation

Assesses the overall quality of an interface.

- Done once an interface is (more or less) nished.
- Either compare alternative designs, or test speci c performance requirements.
- Collect bottom-line data – quantitative measurements of performance: how long did users take, were they successful, how many errors did they make.

Modi ed Soup Analogy

Extending Robert Stake's soup analogy [Stake, 1976; Lockee, Moore and Burton, 2002]:

- When the cook tastes other cooks' soups, *that's exploratory*.
- When the cook assesses a certain recipe, *that's predictive*.
- When the cook tastes the soup while making it, *that's formative*.
- When the guests (or food critics) taste the soup, *that's summative*.

Usability Evaluation Methods

The methods of usability evaluation can also be classi ed according to who performs them:

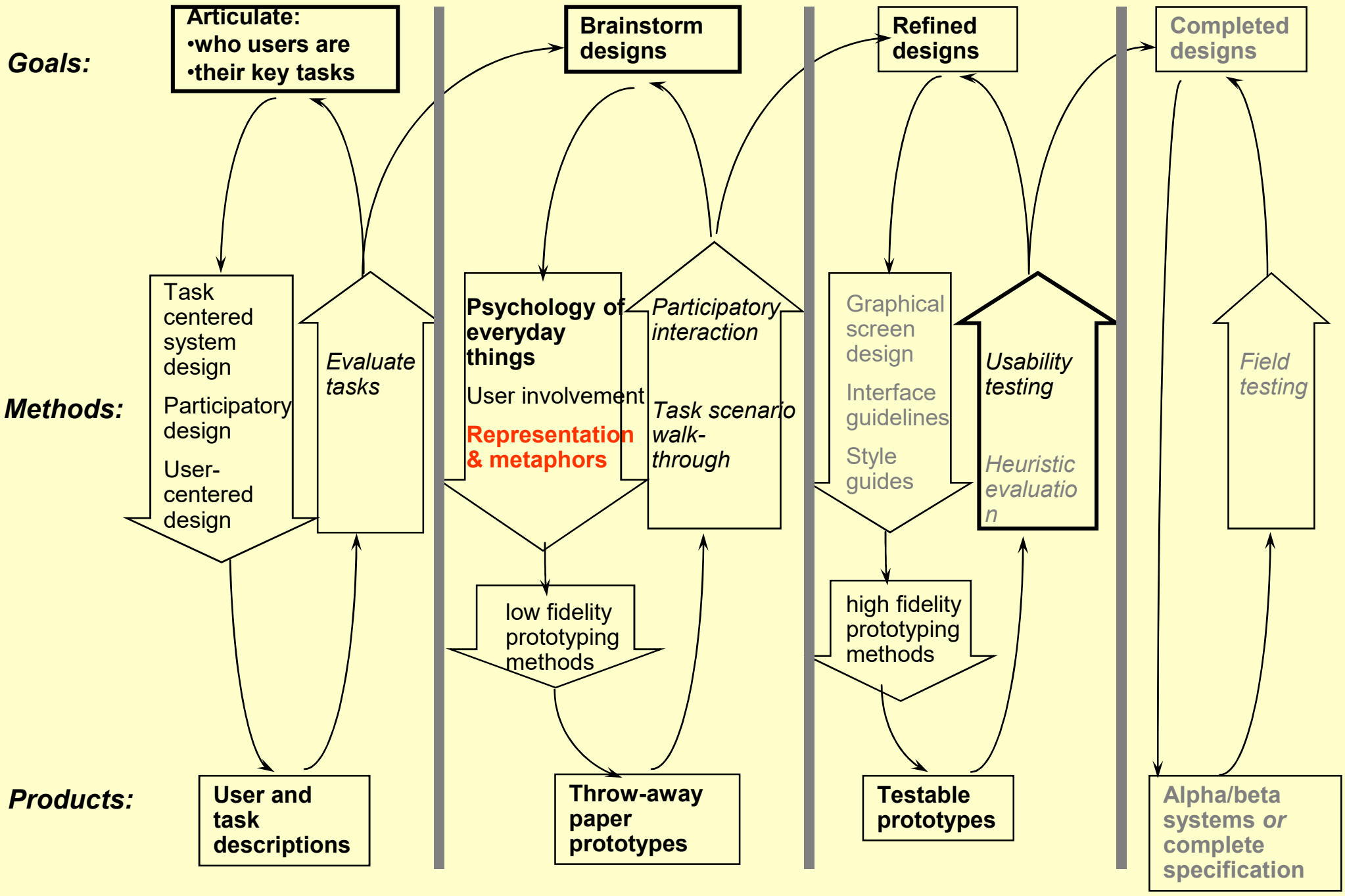
- **Usability Inspection Methods**

- Inspection of interface design by usability specialists using *heuristics* and *judgement* (notestusers).

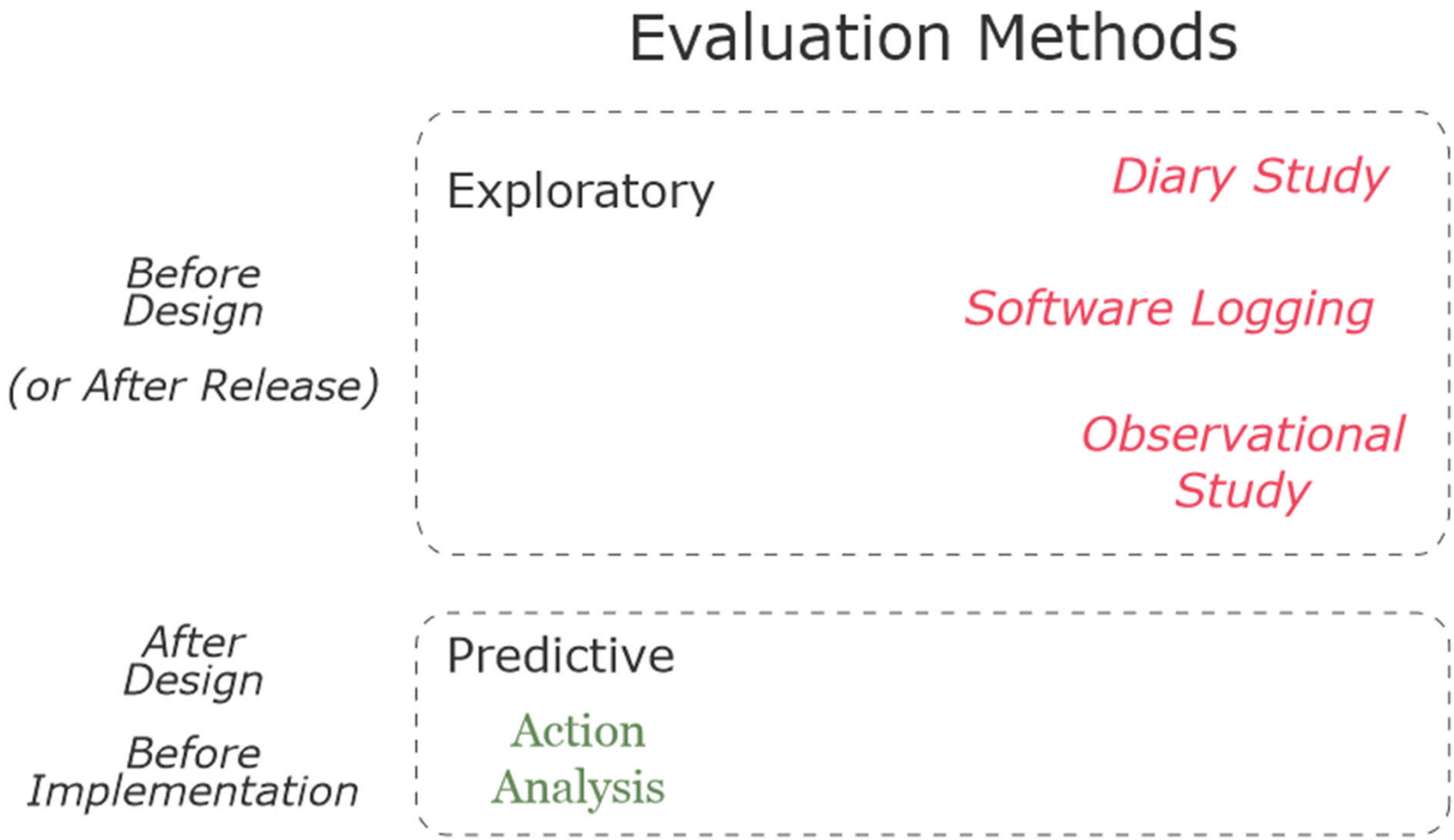
- **Usability Testing Methods**

- Empirical testing of interface design with real users.

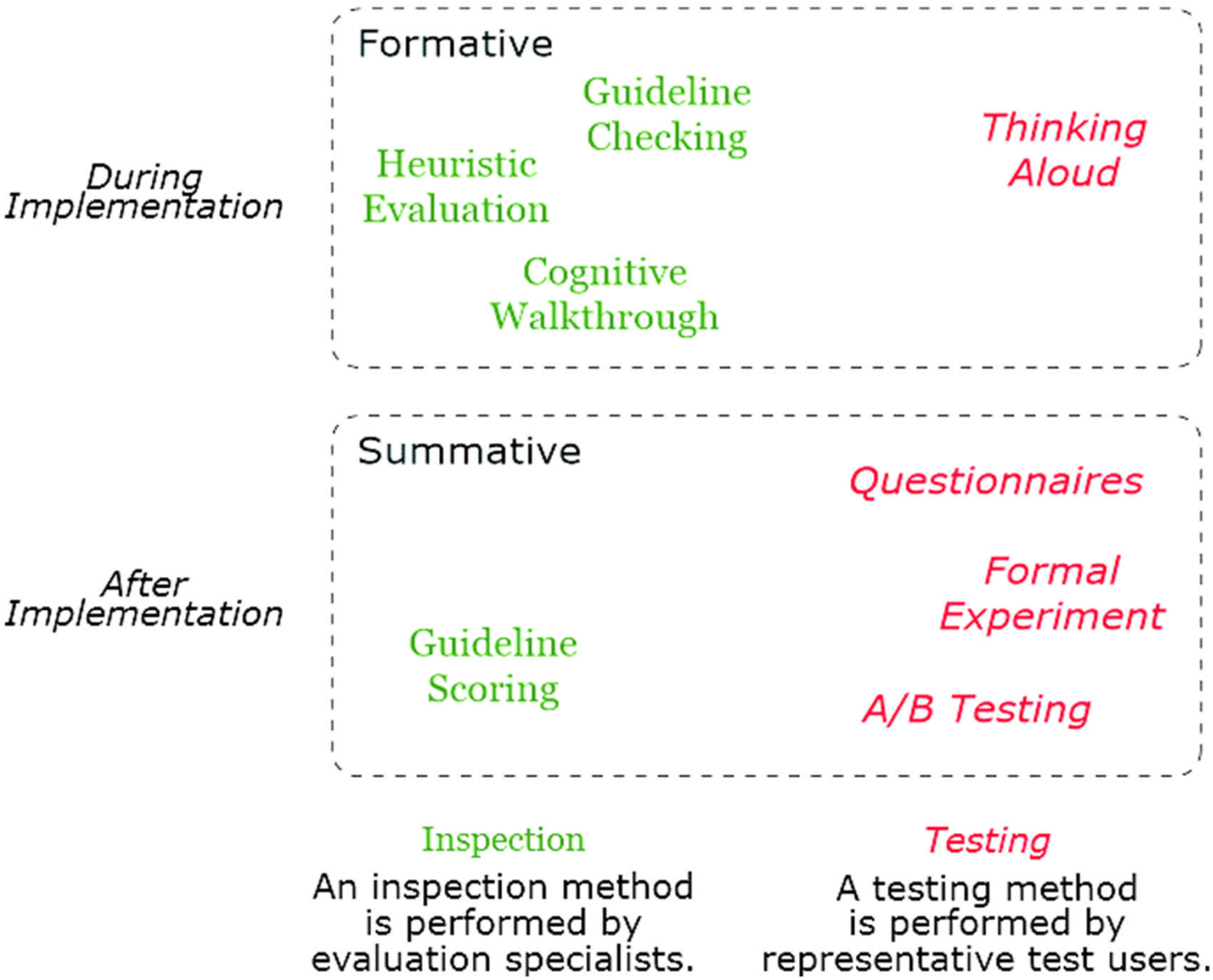
Interface Design and Usability Engineering



Usability Evaluation Methods (cont.)



Usability Evaluation Methods (cont.)

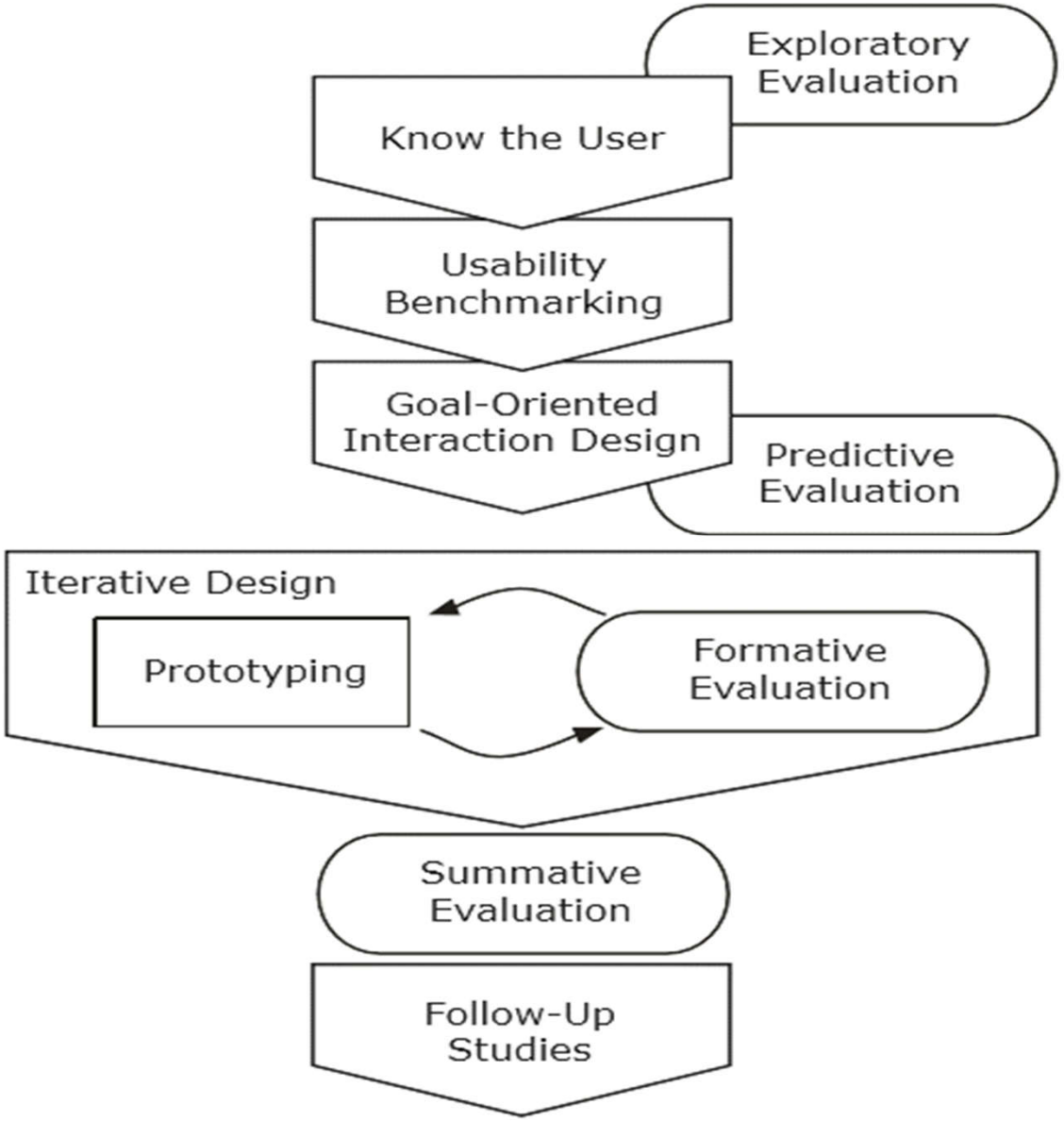


Nine common evaluation methods grouped by purpose and by who performs them.

The Usability Engineering Lifecycle

1. Know the User
2. Usability Benchmarking
3. Goal-Oriented Interaction Design
4. Iterative Design: (a) Prototyping (b) Formative Usability Evaluation (Inspection and/or Testing)
5. Summative Usability Evaluation
6. Follow-up Studies

The Usability Engineering Life cycle (cont.)



The usability engineering lifecycle. Adapted from a gure kindly provided by Martin Loitzl.

Know the User

- Qualitative research: observation of users and interviews.
- Classify users according to their characteristics.
- Draw up a user pro le for each (potential) class of user, based on behavioural and demographic variables.
- Identify user goals and attitudes.
- Analyse work ow and context of work.
- Exploratory evaluation: which software is used, how is it used, and what is it used for.
- Draw up a set of typical user scenarios.

Usability Benchmarking

- Analyse competing products or interfaces heuristically and empirically.
- Set measurable usability targets for your own interface.

Interaction Design

- Goal-oriented initial design of interface.

Iterative Design

“Design, Test, Redesign.” Build and evaluate prototype interface, then:

- Severity ratings of usability problems discovered.
- Fix problems→new version of interface.
- Capture design rationale: record reasons why changes were made.
- Evaluate new version of interface.

until time and/or money runs out. A cycle of continuous improvement.

Building Prototypes

- Verbal description.
- Paper prototype.
- Working prototype.
- Implementation of nal design.

Formative and Summative Usability Evaluation

The usability evaluation methods are described according to who performs them:

- Usability inspection methods
- Usability testing methods

Follow-Up Studies

Important usability data can be gathered after the release of a product for the next version:

- Speci c eld studies (interviews, questionnaires, observation).
- Standard marketing studies (what people are saying in the newsgroups and mailing lists, reviews and tests in magazines, etc.).
- Analyse user complaints to hotline, modi cation requests, bug reports.
- Usage studies of longer-term use of product:
- – Diary studies.
- – Software logging: instrumented versions of software→log data.
- – Observational studies.

Planning Usability Activities

1. Prioritise activities.
2. Write down explicit plan for each activity.
3. Subject plan to independent review (e.g. colleague from di erent project).
4. Perform pilot activity with about 10% of total resources, then revise plan for remaining 90%. [Always perform a pilot study!]

Usability Benchmarking

Usability benchmarking:

- how usable is the competition?
- how much better should your interface be?
- what is your likely return on investment?

Competitive Analysis

Competitive analysis of competing systems:

- Determine the current state of the art and decide how far to raise the bar.
- Analyse competing products or interfaces heuristically (run a heuristic evaluation) or empirically (run a thinking aloud test or formal experiment).
- “Intelligent borrowing” of ideas from other systems.

Set Usability Targets

Decide in advance on usability metrics and desired level of measurable usability (usability targets).

For example:

– The current system exhibits 4.5 errors per hour on average for an experienced user. The target for the new version is less than 3 errors per hour.

– From competitive analysis, on the main competing web site, novice users take 8 mins. and 21 secs. on average to book a ight. The target for our new web site is 6 mins.

Return on Investment

Estimate return on investment (ROI) by performing a nancial impact analysis:

- Compare potential savings based on ***loaded cost*** of users to to the estimated cost of the usability e ort.
- Jakob Nielsen concludes [Nielsen, 2003] that current best practices call for devoting about 10% of a project's budget to usability.

Usability Inspection Methods

Inspection of interface design using heuristic methods (based on analysis and judgement rather than experiment).

1. **Heuristic Evaluation**: A small team of evaluators inspects an interface using a small checklist of general principles and produces an aggregate list of potential problems.

2. **Guideline Checking**: An evaluator checks an interface against a detailed list of speci c guidelines and produces a list of deviations from the guidelines.

3. **Cognitive Walkthrough**: A small team walks through a typical task in the mind set of a novice user and produces a success or failure story at each step along the correct path. [analyses learnability]

4. **GuidelineScoring:** An evaluator scores an interface against a detailed list of speci c guide lines and produces a total score representing the degree to which an interface follows the guidelines.

5. **Action Analysis:** An evaluator produces anestimate of the time an expert userwill take to complete a given task, by breaking the task down into ever smaller steps and then summing up the atomic action times. [analyses e ciency]

Would You Use Untested Software?

Would you knowingly use untested software?

- How many of you have written programs that are used by other people?
- How many of you have watched or observed users using your software?
- How many of you actually evaluated or tested your interface before it was used?
- In practice, most software developers do not actually conduct any kind of usability evaluation [due to perceived expense, lack of time, lack of

Heuristic Evaluation

- Small team of evaluators (usually usability specialists) systematically checks interface design against small set of recognised usability principles (the “heuristics”).
- Online Resources
  - Jakob Nielsen; Heuristic Evaluation; http://nngroup.com/topic/heuristic-evaluation/

Usability Heuristics

Revised list of usability heuristics from [Nielsen, and Molich 1994]:

1. Visibility of System Status [Feedback]

- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- For example: busy cursor [1–10s], progress indicator [>10s].

2. **Match Between System and the Real World** [Speak the Users' Language]

- 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. Match users' mental model. Beware of misleading metaphors.

Usability Heuristics (cont.)

3. User Control and Freedom [Clearly Marked Exits]

- 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.

4. Consistency and Standards [Consistency]

- Users should not have to wonder whether di erent 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 rst place.
- For example: select le from menu rather than typing in name, con rmation before dangerous actions, beware of modes, avoid similar command names, warning if Caps Lock is activated when entering a password, etc.

Usability Heuristics (cont.)

6. Recognition rather than Recall

- Make objects, actions, andoptions 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.
- Users' short-term memory is limited. Provide examples, default values, easily retrievable instructions.

Usability Heuristics (cont.)

7. **Flexibility and E ciency of Use** [Accelerators]
   - 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.
   - Forexample: abbreviations, command keys, type-a head, edit and reissue previous commands, menu of most recently used les, macros.
8. **Aesthetic and Minimalist Design** [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. “Less is more”

Usability Heuristics (cont.)

9. Help Users Recognise, Diagnose, and Recover from Errors [Good Error Messages]

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Phrase error messages defensively, never blame the user. Multileve lmessages. Link to help system.

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.
- Liberal use of examples.

Limits on Response Times

- **0.1sec**: is the limit so that the system appears to react instantaneously. Important for direct manipulation, virtual world navigation.
- **1sec**: is the limit so that the user's ow of thought stays uninterrupted. Display a busy cursor if things will take longer than 1 sec.
- **10secs**: is the limit for keeping the user's attention on the task at hand. Display a progress indicator if things will take longer than 10 secs.

Performing a Heuristic Evaluation

- Design may be verbal description, paper mock-up, working prototype, or running system. [when evaluating paper mock-ups, pay special attention to missing dialogue elements!]
- Provide evaluators with checklist of usability heuristics.
- Optionally provide evaluators with some domain-speci c training.
- Each evaluator works alone (≈1–2 hours).

Performing a Heuristic Evaluation (cont.)

- Interface examined in two passes: rst pass focuses on general ow, second on particular elements in detail.
- Notes taken either by evaluator or evaluation manager.
- Make list of potential problems and list of positive ndings.
- Take screen shots (in PNG) as you work - they may not be reproducible later.
- Independent ndings are then aggregated into one large list (by evaluation manager). This is best done with a spreadsheet.

Performing a Heuristic Evaluation (cont.)

- The large list of potential problems is distributed to each evaluator.
- • Each evaluator now assigns severity ratings individually to each problem in the large list(unseen by the other evaluators).
- The individual severity ratings are averaged to obtain the nal severity rating for each problem.
- The long list is sorted in decreasing order of average severity. Now you know why a spreadsheet comes in handy.
- Group debrie ng session to suggest possible redesigns.

How Many Problems are Found?

- How Many Problems are Found?

Aggregated Evaluations

- Individual evaluators found relatively few problems.
- Aggregating thee valuations (merging the problem lists) of several individuals produced much better results.
- Group debrie ng session to suggest possible redesigns.

Heuristic Evaluation in Practice

When heuristic evaluation is used in standard industry practice today:

- It is often called an *expert review*.
- It is typically performed by 3–4 usability specialists.
- It might cost around MUCH

Pros and Cons of Heuristic Evaluation

- cheap
- intuitive
- usable early in development process
- nds many problems
- nds both major and minor problems
- ==> may miss domain-speci c problems

Severity Ratings

Severity ratings can help prioritise the xing of usability problems.

- After evaluation sessions, a complete aggregate list of usability problems is given/sent to each evaluator.
- Working independently, evaluators assign severity rating [on scale of 0–4] to each problem (≈ 30 mins.).
- Severity rating of single evaluator is unreliable, mean of 3–5 evaluators is satisfactory.

Five-Point Severity Scale

Score	*Severity*	*Fix Priority*
4	catastrophic problem	imperative
3	major problem	high
2	minor problem	low
1	cosmetic problem only	
0	not a problem at all	

Order of Criticality

To explicitly take problem frequency into account, assign criticality ratings.

Criticality = Severity Ranking + Frequency Ranking

Severity		*Frequency*	
4	catastrophic	4	>90%
3	major	3	51–89%
2	minor	2	11–50%
1	cosmetic	1	1–10%
0	none	0	<1%

Guideline Checking

Guidelines ...speci c advice about usability characteristics of an interface.

- An evaluator checks an interface against a detailed list of speci c guidelines and produces a list of deviations from the guidelines.
- Whereas heuristic evaluation employs 10 broad principles, guide line checking often involves dozens (or hundreds) of more speci c individual items on a checklist.

Example Sets of Guidelines

- Sidney Smith and Jane Mosier; Design Guidelines for Designing User Interface Software; The MITRE Corp., 1986. [944 guidelines] ISBN 9992080418 (com, uk) ftp://ftp.cis.ohio-state.edu/ pub/hci/Guidelines
- C. Marlin Brown; Human-Computer Interface Design Guidelines; Ablex, NJ, 1988. ISBN 0893913324 (com, uk) [302 guidelines]
- Deborah Mayhew; Principles and Guidelines in Software User Interface Design; Prentice-Hall, 1991. ISBN 0137219296 (com, uk) [288 guidelines]

Pros and Cons of Guideline Checking

- cheap
- intuitive
- usable early in development process
- - time-consuming
- - can be intimidating
- – often hundreds or thousands of speci c guidelines.

Cognitive Walkthrough

Task-oriented walk through of interface, imagining novice users' thoughts and actions. Focus esexplicitly on learnability.

- Design may be mock-up or working prototype.
- Analogous to structured walkthrough in software engineering.
- Based on cognitive model (CE+) of human exploratory learning.

Exploratory Learning

Rather than read manual or attend course, users often prefer to learn new system by “trial and error”→ exploratory learning [Carroll and Rosson, 1987]:

1. Start with rough idea of task to be accomplished.
2. Explore interface and select most appropriate action.
3. Monitor interface reactions.
4. Determine what action to take next.

Cognitive Walkthrough Preparation

a) Identify user population.

b) De ne suite of representative tasks.

c) Describe or implement interface or prototype.

d) Specify correct action sequence(s) for each task.

Cognitive Walkthrough Steps

For each action in solution path, construct credible “success” or “failure” story about why user would or would not select correct action. Critique the story to make sure it is believable, according to four criteria:

a) Willtheuserbetryingtoachievetherighte ect? What is users’ goal – will they want to select this action?

b) Willtheuserknowthatthecorrectactionisavailable? Is control (button, menu, switch, triple-click, etc.) for action apparent (visible)?

c) Willtheuserknowthatthecorrectactionwillachievethedesirede ect? Once users nd control, will they recognise that it is the correct control to produce the desired e ect?

d) Ifthecorrectactionistaken,willtheuserseethatthingsaregoingok? After correct action, will users realise progress has been made towards the goal (feedback)?

- Note that CW always tracks the correct action sequence. Once the user deviates from the correct path their further progress is no longer considered.

Group Walkthrough

- Performed by mixed team of analysts (designers, engineers, usability specialist).
- Capture critical information on three group displays (ip charts, overheads):
- 1. User knowledge (prior to and after action).
- 2. Credible success or failure story.
- 3. Side issues and design changes.
- Perhaps also videotape entire walkthrough.

Pros and Cons of Cognitive Walkthrough

- nds task-oriented problems
- helps de ne users' goals and assumptions
- usable early in development process
- - some training required
- - needs task de nition methodology
- - applies only to ease of learning problems
- -- time-consuming

Guideline Scoring

- The interface is scored according to its conformance against a weighted list of speci c guidelines.
- A total score is produced, representing the degree to which an interface follows the guidelines.

Web Technologies - Checklist Homepage Design / Usability

Homepage (URL)		Date		Nr.	
Tester		Size	kB	Score	%

Nr	Topic	Recommended Design	Strength	Points	You
1	Download time	50 kB (<10 sec for your customer)	✶✶✶	3	
2	Window title	Start with Company Name	✶✶✶	3	
3	Title tag line	What about	✶✶✶	3	
4	Readable URL	Hackable URL	✶✶	2	
5	Error page	Catch errors/dead links, to search	✶✶	2	
6	Meta tags	For search engines	✶✶✶	3	
7	Alt Information	Images, accessibility, Lynx	✶✶	2	
8	Page width	770 pixel (620-1024)	✶✶	2	
9	• Liquid vs. frozen layout	Liquid	✶✶	2	
10	• Page length	<2 pages (1000-1600 px)	✶✶	2	
11	Frames	No	✶✶✶	3	
12	Logo placement	Upper left	✶✶✶	3	
13	• Logo size	80x68 Pixel	✶✶	2	
14	**Search**	Yes, in a box, always	✶✶✶	3	
15	• Search placement	Upper part, right or left corner	✶✶✶	3	
16	• Search box colour	White	✶✶✶	3	
17	• Search button	Call it “Search” or “Go”	✶✶	2	
18	• Width of search box	>=25 characters (30 best)	✶✶	2	
19	• Type of search	Simple search (Link to advanced)	✶✶	2	
20	**Navigation**	4 types: left, tabs, top, categories	✶✶	2	
21	• Footer navigation links	Max. 7 links, single line	✶	1	
22	• Sitemap link	Name “Site Map”	✶✶	2	
23	Routing page	No	✶✶	2	
24	Splash page	No	✶✶✶	3	
25	Sign-In	“Account” or “Sign In”	✶	1	
26	**About** the company	Always include it	✶✶✶	3	

Web Technologies - Checklist
Homepage Design / Usability

27	• About link	Call it “About <company>”	✶✶	2
28	• Contact information	Call it “Contact us”	✶✶	2
29	• Privacy policy	If you collect data	✶✶✶	3
30	• Name of privacy link	Call it “Privacy Policy”	✶✶	2
31	Job opening	Call it “Jobs” if you have it	✶✶	2
32	Help	If it is a complex site	✶	1
33	• Help placement	Upper right	✶✶	2
34	Auto-playing music	No	✶✶✶	3
35	Animation	No	✶✶	2
36	Graphics/illustration	5-15%	✶	1
37	Advertising	<= 3 ads	✶✶	2
38	**Body** text colour	Black	✶✶	2
39	• Body text size	12 points	✶	1
40	• Body text size frozen	No	✶✶✶	3
41	• Body text typeface	Sans-serif	✶	1
42	• Background colour	White	✶✶	2
43	• Link colour (unvisited)	Blue	✶✶	2
44	• Link colour (visited)	Purple	✶	1
45	• Link colour different	Yes (not light grey)	✶✶✶	3
46	• Link underlining	Yes (except in navigation bar)	✶✶	2
	Score of URL:			100

✶	Default Recommendation
✶✶	Strong Recommendation
✶✶✶	Essentail Recommendation

from Nielsen, Homepage Usability
adapted by alexander@nischelwitzer.com V1.3

Pros and Cons of Guideline Scoring

- cheap
- intuitive
- - must select and weight guidelines
- - guidelines or weightings often domain-dependent

Action Analysis

- Quantitative analysis of actions to predict time skilled user requires to complete tasks, based on time estimates for typical interface actions. Focuses on performance of skilled user (e ciency).
- Two avours (levels of detail):
- a) Formal or “Keystroke-Level”
- b) Informal or “Back-of-the-Envelope”

Usability Testing Methods

- “Would you y in an airplane that hasn’t been ight tested? Of course not. So you shouldn’t be using software that hasn’t been usability tested. ”

[Ben Shneiderman, The Front Desk, BBC Video, 1995.]

Usability Testing Methods (cont.)

Empirical testing of interface design with representative users.

- Thinking Aloud: Test users verbalise thoughts while performing test tasks.
- Co-Discovery: Twotestusersexploreaninterfacetogether. Insightisgainedfromtheirconversation while performing test tasks.
- Formal Experiment: Controlled experiment, face-to-face with test users, measurements and statistical analysis.
- A/B Test: Controlled experiment on (part of) actual user population, typically (remote) web site users, with measurements and statistical analysis.
- Query Techniques: Interviews and questionnaires.
- Usage Studies: Usage data is collected from a small number of users working on their own tasks in their natural environment over a longer period.

Why do Usability Testing?

- More often than not, intuitions are wrong!
- People believe they understand behaviour of others based on their own experiences.
- This belief only lost through prediction then measurement (→usability tests).
- Designers of system nd it very easy to use.
- Experience changes one's perception of the world.
- It is almost impossible to "forget" an experience and put oneself in position of someone not having had the same experience.

Preparing for Usability Testing

Test Environment

Ensure comfortable test environment:

- Organise quiet room.
- Put up sign “User test in progress – Do not disturb”.
- Disable telephones (xed line and mobile).
- Ensure adequate lighting.
- Provide (non-alcoholic) refreshments.

Test Equipment

Roles in the Test Team

- Test Facilitator (Administrator, Moderator, Manager, Monitor). In overall charge of test, responsible for all interaction with test user (introduction, test, debrie ng).
- Data Logger(Scribe)
- Video Operator
- Computer Operator

Preparing for Usability Testing (cont.)

- **Test Users**
  - Test Participants or TestUsers
    - Users taking part in the test.
    - Never ever call them test subjects!

Six Stages of Conducting a Test

1. Develop the Test Plan
2. Select and Acquire Participants
3. Prepare Test Materials
4. Run a Pilot Test
5. Conduct the Real Test
6. Analysis and Final Report

Usability in Practice