

A short introduction to

designing
user-friendly
interfaces //

usable machine



Usability
is often ignored
until it becomes
a problem

Introduction

This booklet is about usability and user interface design. It is aimed at anyone working with design and development of applications - whether it is for pc, web or mobile devices. / 3

The booklet contains a collection of the best techniques for achieving a user-friendly design. They are presented in a compact and easily accessible format.

If you are interested in learning more, there is a bibliography on the last page.

If you have any comments or questions, please send an email to ingrid@usablemachine.com

What is usability?

- 4 / Usability is not an objective measure. While one person may easily use a product, somebody else might find it very difficult to use the same product. Therefore, we speak of an estimate of the *overall usability*.

There is a high degree of usability when:

1. The user is able to use the product with little or no prior training
2. The user experiences the product as easy to use and “logical”
3. The user makes no or few errors and is able to correct the errors without help
4. The user can use the product at a relatively high speed

How is usability measured?

If you want to improve usability, you need more than just a sense of what works and what doesn't. You need metrics. / 5

The best way to measure ease of use is to “test” the product by letting users use it under observation – a bit like a scientist observing mice in the laboratory.

Such a test is called a usability test.

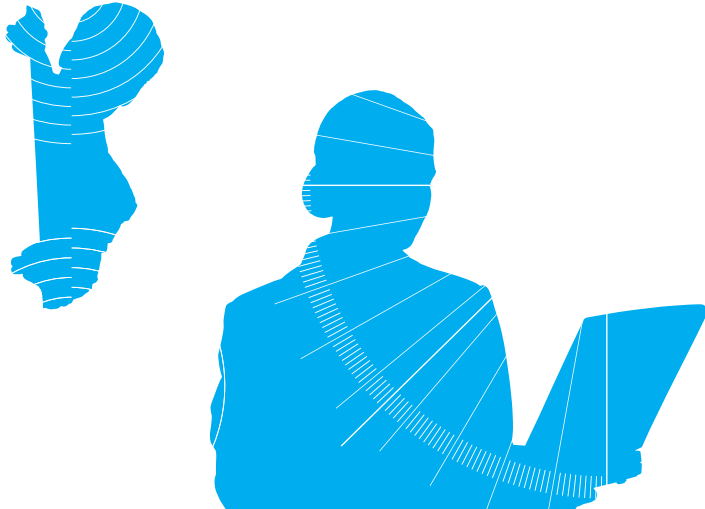


Usability test

6 / Recruiting participants

Participants should represent typical users, who might choose to use the application in their own life.

Avoid testing with existing users, since they already know the application and will not make the same mistakes and misunderstandings as when they first began to use it.



How to run a test

Invite 5-8 people to an individual session. Give them written instructions asking them to solve a series of tasks in your application. Along the way they are asked to “think aloud” so you can follow their reasoning.

Do not offer your help

Do not help the participants during the test. They need to try to solve the tasks themselves, so you can see what they are able to do on their own and what they don’t understand.

Results

A usability test reveals the problems that users encounter when they learn to use the application. It reveals what is difficult to understand and what is easy to use.

During the usability test you can also measure time per task and ask questions that help you understand how the user thinks.

Four tests you can't trust (by themselves)

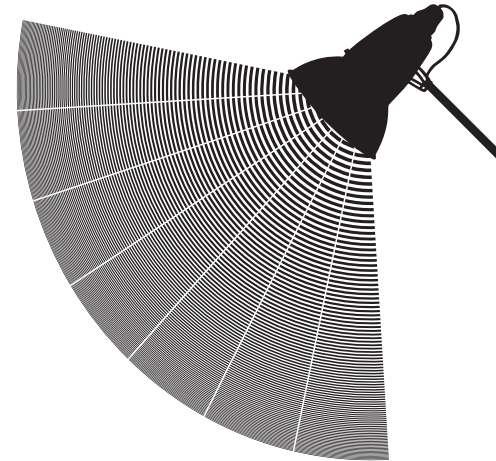
- 8 /
- 1. The user evaluates the level of usability**
For example by giving it a rating. This method is not reliable because the user does not always understand how the application *really* works.
 - 2. Counting the number of clicks**
It doesn't necessarily follow that few clicks equals a high degree of usability. A function may require few clicks, but still be poorly understood.
 - 3. The development team tests the usability**
Programmers and designers are typically more technically inclined than "real users" and know the application too well. They are rarely typical users.
 - 4. User comments submitted online**
For example via a user forum. This is not comprehensive. Users do not report all the problems they encounter.

How is usability achieved ?

A high degree of usability can be achieved in many different ways, and each person has his or her favorite methods. / 9

The following methods are some of the most recognized:

- User studies
- Personas
- Scenarios
- Wireframes
- Storyboarding
- Usability testing



User studies

10 / Make sure you get to know your users before you design for them. This is done by gathering information about their lives, background, working practices, problems, goals, etc.

- **Interviews**
Ask users how they live, work, think, shop, use their computer, communicate, etc.
- **Focus groups**
A group of users discuss your application or the topic it deals with.
- **Questionnaires**
Questionnaires on paper or online are an effective way to collect large volumes of data.
- **Observation**
Observe a user for an hour or a day and see how he/she behaves and what problems he/she encounters.

Personas

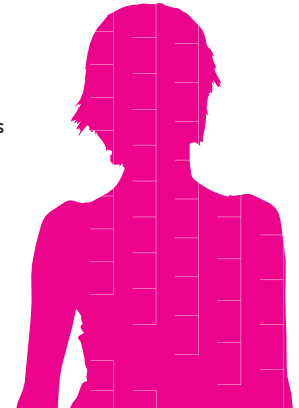
A persona is a description of a typical user. He or she is portrayed as a “real person” with name, age, interests, concerns, preferences and personal attributes. / 11

Personas are used in the design process to continually validate if the design meets the users' needs. This is a great tool to ensure a user focus.

Create one or more personas for your application and add as much detail as possible.



Natalie / 28 years / Waitress
Natalie works at a French restaurant.
She lives alone in an apartment in Copenhagen. She loves dogs, running and going out with friends. She mostly uses her computer for email and Facebook.



Scenarios

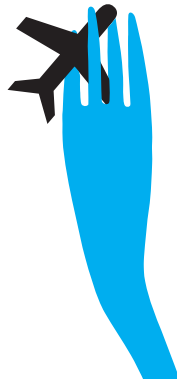
12 / A scenario describes a process involving the application, seen from the user's perspective. It tells about the overall flow without going into details and without mentioning the technical solution.

For each scenario describe the user, environment, motivation, objectives and actions - step by step. Create a scenario for each of the key processes in the application.

This method ensures that the design is based on the user's experience of the application.



Scenario example / John
John is travelling to China with his wife and he wants to secure good seats for himself on the plane. He goes to the airlines website and finds his plane. He logs in by using his credit card number. He chooses 2 seats in the middle of the plane. He also orders a vegetarian meal for his wife.



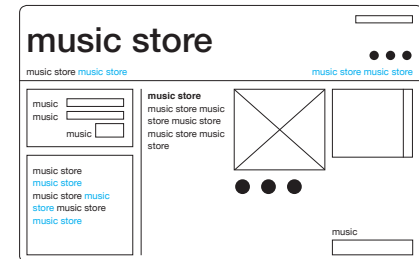
Wireframes

The user interface should be visualized before it is programmed. However, there is no reason to create a finished graphic design with complete data only to demonstrate how the application should work.

/ 13

The best technique for making a quick outline of an interface is called wireframes.

A wireframe is a sketch indicating the fields, buttons, pictures etc. It shows the layout of a screen, but without graphic design or "real" content.



Storyboarding

14 / Storyboarding is a technique known from movie production - every scene is shown with a picture, much like a cartoon.

When doing interface design, storyboarding is used to visualize how the finished design and flow will look. The storyboard shows a complete process or scenario - screen by screen.

Storyboards enable the development team as well as users to give feedback to the design and improve it before a single line of code has been written.

How it is done

First, draw the screen as it looks when the application opens. Use the wireframe technique to make it faster.

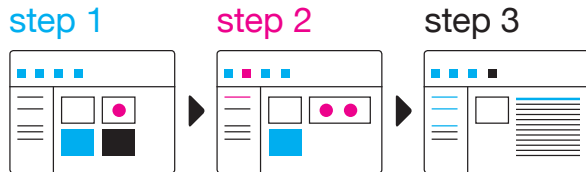
Then, draw the screen showing what the user does. Show for example where the user clicks or presses with his/her finger.

Lastly, draw the screen showing the system feedback.

Illustrate an entire process this way.

It can be quite time-consuming to storyboard. But it is much cheaper than having to recode the application if the usability issues are found at a later time.

/ 15



Usability testing

16 / Doing a usability test towards the end of the development process is not effective and will only leave time to make small, superficial changes to the design.

Therefore, it is important that you test with users as early as possible. You can, for example, test your design by making users work with a storyboard as a non-fully functional prototype.



An iterative design process

Iterative means “to repeat something,” and an iterative design process can look like this:

17 /

- Create a design
- Build a prototype, for example a storyboard
- Evaluate the design with users or the development team
- Continue the design activity and include the feedback obtained through evaluation
- Built a prototype... and so forth

With each iteration the design becomes more user-friendly and thoughtful.

7 design principles

18 / There is no formula for how to create a user-friendly interface. However, by following best practices you can get really far.

Here are 7 design principles to help you on the right track:

1. Simplicity
2. Show the system status
3. Be consistent
4. Design to avoid mistakes
5. Write clearly and concisely
6. Reflect the users' reality
7. Ask the users

1. Simplicity

19 / Interfaces are often difficult to use because they show too much information at one time. The user loses oversight and does not know where to focus his or her attention.

Here are some techniques to simplify the interface:

- Hide detailed information in secondary menus
- Show only the most important information and show a link to the rest
- Organize and group content for better oversight
- Let the screen focus on one task at a time



2. Show the system status

20 / It is important that the user knows the status of the application. Therefore you should always provide the user with feedback when he or she does something.

The user can usually not figure out or remember what the application has done, what data is updated, or whether something has failed or succeeded - unless he/she gets a clear message about this.

So if there is anything the user needs to know about a process that has begun, ended or failed, you should show it on the screen.



The transfer has
completed successfully

3. Be consistent

Be consistent in everything you show in the interface. This applies to layout, icons, text, functionality and menus. / 21

Users will always try to relate what they see on the screen to something they already know.

And if they recognize it, they will feel safe and know exactly how to use it.

So it is often better to be consistent with the leading products on the market rather than inventing a new way of doing something. Even if your own idea seems a little smarter.

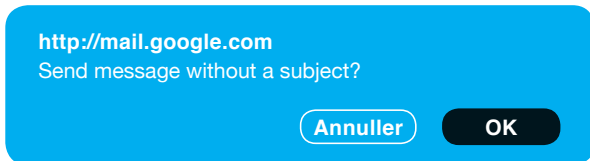


4. Design to avoid mistakes

22 / Nobody likes to be told that they made a mistake whether by receiving an error message or by the application crashing. So the best choice is to help users avoid making mistakes altogether.

You do this by offering help, instructions and warnings at the right time.

If you still need to display an error message, make it easy to understand and tell the user how he/she can correct the error.



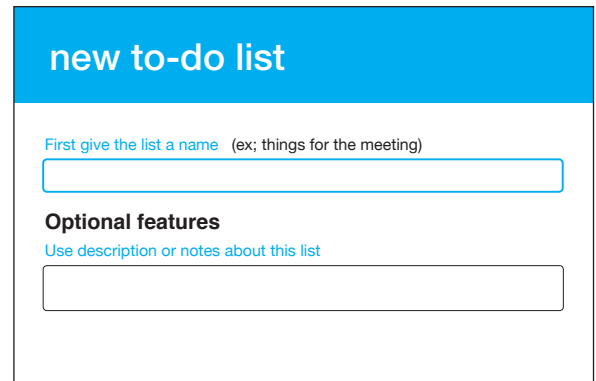
5. Write clearly and concisely

The vast majority of users do not read the text on the interface if it is long or difficult to understand.

/ 23

It is still a good idea to display text in order to explain what the user should do. But the text should be as clear and concise as possible.

This applies to all kinds of texts, such as labels, headlines, help text, and error messages.



A form titled "new to-do list" with a blue header. Below the header, there is a text input field with the placeholder text "First give the list a name (ex; things for the meeting)". Below this field, there is a section titled "Optional features" with a text input field and the placeholder text "Use description or notes about this list".

6. Reflect the user's reality

24 / The user has a picture in his or her head of how his or her world looks – this is also called a “mental model”.

If the application design does not match the user's mental model, he/she will become confused and will spend a long time learning how the application works.

Therefore it is important to reflect the user's reality:

- Use concepts and words that the user already knows
- Organize elements as they are organized in reality
- Use metaphors that make sense in reality
- Follow real workflows

7. Ask the users

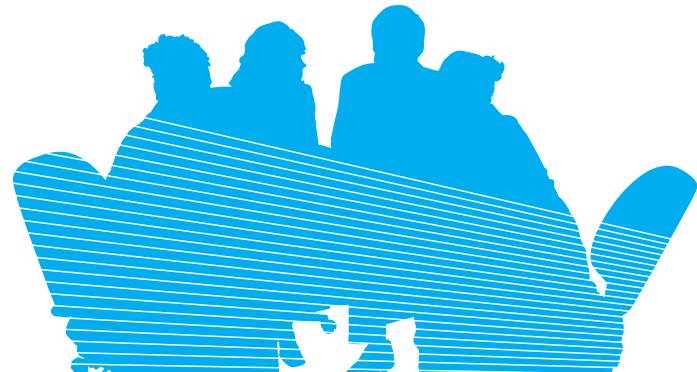
This rule is important. It's about the mindset needed to create user-friendly applications: When in doubt, ask the users!

/ 25

You can do this by calling a typical user you know, or establish a user panel that can respond to questions throughout the development process. Or invite users for usability tests and interviews.

The more you ask, the better your application will suit the users' needs and requirements.

In short: You achieve usability by being in dialogue with users.



Bibliography

26 / Would you like to learn more about usability and interface design?

Here is a list of great resources you can start with:

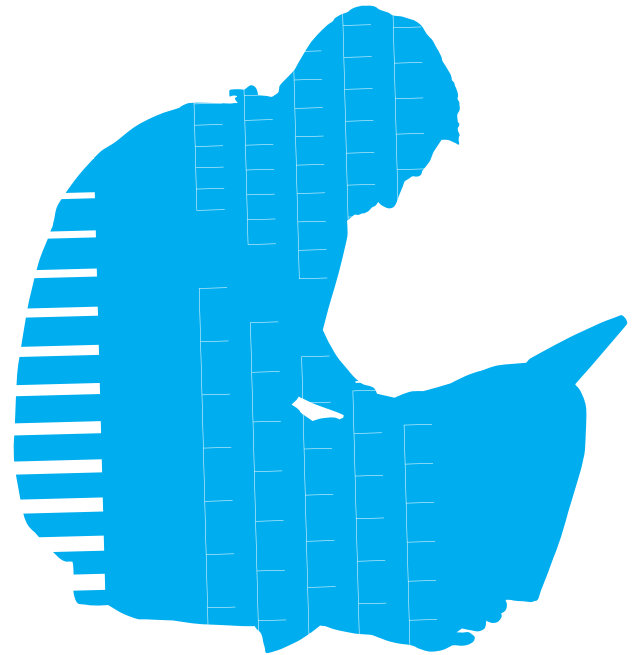
[Don't Make Me Think / A Common Sense Approach to Web Usability](#) by Steve Kruger

[The design of everyday things](#) by Donald A. Norman

[Designing for Interaction / Creating Smart Applications and Clever Devices](#) by Dan Saffer

[About Face 3 / The Essentials of Interaction Design](#) by Alan Cooper et al

www.useit.com by Jakob Nielsen



This booklet is about usability and interface design

It provides a brief introduction to techniques such as usability testing, user studies, personas and wireframes.

It presents 7 design principles that make your design more user-friendly.

The booklet is written by Ingrid Haug. She is the founder of Usable Machine, a company that designs applications for pc, web and mobile devices.

Read more at www.usablemachine.com