

Chapter

2

# Usability

**CHAPTER AUTHORS**

Heng Belson

Japheth Ho



**CONTENTS**

1 Introduction ..... 5

    1.1 Defining Usability..... 5

    1.2 Target Audience..... 5

    1.3 Chapter Goals..... 5

2 The relationship between Usability and User Experience (UX) ..... 5

3 Importance of usability ..... 6

4 Achieving Usability..... 7

    4.1 The 5E’s to Usability..... 7

    4.2 Striking the Balance ..... 10

5 Usability Testing..... 10

    5.1 Getting Started..... 11

    5.2 Measuring Usability..... 12

    5.3 After Completing the Test..... 15

6 Common mistakes and guidelines to good website design..... 15

7 Conclusion..... 26

8 References..... 27

9 Resources ..... 28



## 1 INTRODUCTION

### 1.1 Defining Usability

Usability is the measure of how “usable” something is, but how can we define something as usable? Many books and websites state varying definitions or requirements for something to be usable. In most cases, the common consensus is that the product in question has to first be functionally correct (does what it’s supposed to do), and then be easy to use. Ease of use is again a very subjective thing, but people would generally like things to be easy to learn and remember, intuitive and fast. Some sources also state that usability is like food, in which aesthetics also play a part in deciding how good it actually may be (which we will cover later in [section 4.3](#)). All in all, usability is a big part of how pleasant the user experience in dealing with the product has been. If a user leaves happy and satisfied after using your product, you’re on the right track, but if not, you’re on the right chapter.

### 1.2 Target Audience

In this chapter, we focus on creating good User Interface (UI) in general. While big companies have an entire team dedicated to the design factor alone, in smaller companies or group projects, it is usually also the programmer who has to undertake this responsibility. For programmers especially, taking care of functionality is of more importance than usability, and they only have a very superficial idea of what good UI from the user point of view is. For example, clear navigation or having no more than 3 nested menu selections are no doubt the very basics that every programmer knows, but usability can actually be so much more than that. Hence, this chapter is targeted at all designers or programmers who have to design and code an effective UI that any user will enjoy using.

### 1.3 Chapter Goals

First of all, this chapter explains how usability figures into the entire user experience. It also touches on the importance of good UI in a program or a website. We will then study factors that designers and programmers can take to achieve a good UI design, followed by an introduction to the ways of testing usability. Finally, we take a look at common mistakes made by developers and give guidelines to what can be done better.

Through these topics, we aim to give users an overview of what usability actually involves. Readers would hopefully understand the importance of usability and how good UI is essential in contributing to that. Dr. Jakob Nielsen, author, researcher, and consultant on user interfaces, specialising in web usability states:

*“Usability is like cooking: everybody needs the results, anybody can do it reasonably well with a bit of training, and yet it takes a master to produce a gourmet outcome.”*

It is perhaps through years of experience, constant practice, trials and errors that one learns what usability actually is. Thus, while we do not claim to be able to turn the readers into usability experts, we hope that by giving a clearer picture and the things to take note of, designers and programmers can perhaps use these points that have been noted by usability experts to get a head start in creating good and effective UI for their users.

## 2 THE RELATIONSHIP BETWEEN USABILITY AND USER EXPERIENCE (UX)

User experience, abbreviated as “UX”, is probably the catchphrase that most people confuse usability with. In this section, we shall look into the distinction between these two terms, as well as their underlying relationship.



Let us first use an analogy to illustrate a big picture of “Usability versus UX”: Imagine yourself as a user of the road – a driver, and your goal is to reach your destination – point B. Along your journey, there are many factors ranging from weather to the scenery and road texture that could shape your experience. Eventually, it is the sum of all these different elements that affects your feelings and marks up your user experience.

Usability, on the other hand, can be seen as the layout of the road – whether it is wide enough to accommodate traffic such that there would be no congestion, which could delay your time to point B, and whether the signs along the road are good indications that help you find your way to your destination effectively.

In other words, usability is a narrower concept than UX since it only focuses on goal achievement. UX, in contrast, is a consequence of the presentation, functionality, system performance, interactive behavior, etc. This essentially means that UX per se is an “umbrella” term that encompasses many different skills needed to create a user experience, which include Human Factors, Design, Ergonomics, Human Computer Interaction (HCI), Accessibility, Marketing, as well as Usability. (Mifsud, 2011)

The relationship between UX and usability is that the latter is subsumed by the former, and this also marks the differences of the two where usability enables people to easily accomplish their goals, but UX design covers more than that – it is about giving people a delightful and meaningful experience. UX starts by being useful where the application works as programmed and meet the needs of users. Inherently, functionality forms the most basic aspect of UX – a common tier which most companies have already attained or rather got stuck with. This is where usability lends functionality a helping hand, pulling it nearer towards achieving a higher level of UX by taking away all things that distracts users from happiness and the end result would be a satisfying user experience that empowers what people can do.

### 3 IMPORTANCE OF USABILITY

#### *It's all about the users*

Simply put, usability is determined by the very users. Basically, if users don't like it, they wouldn't use it, and that just makes all your hard work come to naught. You could have created the fastest prime factoriser of a number in the world, or a website that is updated with the latest news on the minute, but it's all pretty pointless if no one wants to use it.

Good usability leaves a good impression on the user, and gives the user reason to want to come back again and again. Bad usability, on the other hand, would only give user headaches trying to figure out how to use the application. We use computers to improve and make things more efficient, but if that process actually takes more time then doing it manually, it's really not worth all that effort.

### *Users are choosier then before*

In the past, computers seemed to be the tool for people to play games or for project research. It is probably why not much thought was put into developing probability usability as it was more or less assumed that you had some basic computer knowledge already. However, these days, everyone from 6 to 60 years old owns a computer. From e-shopping, to updating one's social network account, to reading the news, it's a part of our everyday lives. With so many choices of sites to choose from, how does one make itself stand out from the rest? Well, through the use of good usability of course.

Also, with so much technological improvements and research into usability along the way, there's simply no reason why good usability cannot be implemented. It is cheaper and doesn't require a master's degree in computer science to create it.

### *It could make or break your business or career*

In very tangible terms here, we're talking about money. As we mentioned earlier, it's really all about the users and now more specifically, when users become much more than just a hit counter statistic. Whether your website makes money or not, could determine the next step of your life. No one would ever want to be in the shoes of the designer who had made a design fault, much less the owner of the business that's losing money.

In an article found on UIE.com (M.Spool, 2009), it talked about how by simply changing a button, an e-shopping company had managed to save \$300 million dollars a year. As amazing as it may sound, it's all true. Traffic flow wasn't a problem as lots of potential customers were visiting the site every day. Unfortunately that it was all the customers remained, just potential ones. This problem arose because of differing views to the website. While the website designers believed they were making things easier for the users, the reality was that it was actually having a detrimental effect instead. In a nutshell, the website required customers to sign up or login for an account before they could proceed to check out. From the designers point of view, this was so that customers need to re-enter payment details again during checkout, but what users felt was that the company only wanted to store their personal details for marketing research use. In the end, by removing a "register" button that was actually preventing customers from making purchases all because customers were wary of the implications of clicking on it, a huge increase in profits was made.

## 4 ACHIEVING USABILITY

Now after valuing the importance, let us study the ways to achieving usability by first understanding the characteristics that define the term. In this section, we will be focusing on five dimensions of usability – the 5E's, which is covered in a book by Michael Albers and Beth Mazur: Content and Complexity (Mazur, 2003).

### 4.1 The 5E's to Usability

#### *Effective*

*The **completeness and accuracy** with which users achieve their goals*

Effectiveness is concerned primarily with how well a work is done. It is measured by looking at whether user's goals were met successfully and whether all work is correct. If a user cannot complete the task he/she sets out to do, it probably does not matter if the experience was good or bad. At the end of the day, they have failed to complete their tasks or meet their goals.

## Efficient

*The **speed** in which users complete their tasks*

Efficiency, unlike effectiveness, means that the application should allow users to complete their tasks with reasonable amount of time or effort. Common tasks should be possible with simply one or two steps.

Most often, the metrics used include number of mouse clicks and keystrokes required for users to complete their tasks. Having said that, navigation design elements like keyboard shortcuts, menus, links and other buttons all have an impact on efficiency. If they are well-designed, the user will require less time and effort to make navigation and action choices, which would in turn maximize efficiency.

Therefore to figure out which are the right choices of navigation design elements to maximize efficiency, you have to first understand the users and how they prefer to work. For example, are they likely to use the interface once in a while, or chronic users who might learn hidden controls and shortcuts? For keyboard shortcuts, they can be extremely efficient for expert users who work with the interface intensively. But, if these shortcuts are the primary interaction tool, they can slow down users who are unfamiliar with them, or with the application. Likewise, an interface with a series of ordered choices may be the optimal solution for one-time or infrequent users, but on the other hand, it might be annoyingly slow as the only way of interacting for a frequently used application.

Waiting or computer response time also plays a part. During an interview (Maven, 2007) with Jakob Nielsen, the usability guru, he stated that there are 3 rules for response times which are based on fundamental human characteristics. He said:

*“The rules are; if it is faster than one tenth of a second, you don’t feel like you are waiting at all. If it is more than one tenth of a second, you can tell you are waiting, but up to one second, it still feels like smooth navigation. Between one and ten seconds is the limit for your attention.”*

The rule of thumb being that waiting times should be under 1 second as far as possible. Anything after 10 seconds and your mind would start to wonder off and start to question if the site is even functional or not.

A study done in the University of Nebraska-Lincoln (Nah, 2004) on the tolerable waiting time also seems to support this theory, with results showing that the tolerable waiting time for a response is around 2 seconds. Some feedback presented during the waiting time like a progress bar would be able to prolong this time slightly.

## Error tolerant

*The ability of the interface to **prevent** errors or help users **recover** from those that occur*

Error tolerance represents how well an application prevents errors caused by user interaction and helps users recover from any errors that do occur. No doubt that the greatest aim for an application is to have no errors, but that is rather ambitious. Developers are still human, and computer systems are far from perfect. This means mistakes will nonetheless arise. Therefore, the real challenge here is how helpful the application can be when an error does occur.

## Engaging

*How **pleasant** or **satisfying** the interface is to use?*

An interface is engaging if it is pleasant and satisfying to use. Although graphic design is the most apparent element of this characteristic, a more subtle aspect of the interface also affects

how engaging it is. For example, readability of the text and the way information is organized for presentation can change a user's relationship to the interface. Likewise, interaction styles from a game-like simulation to a simple menu-command system, can affect the user's satisfaction level. However, akin to all usability characteristics, these qualities must be appropriate to the tasks, users and context.

Still, it's not about just simply making the site look good. As with most things, perception is everything. With aesthetics, there exists a condition known as the Aesthetics-Usability Effect whereby users perceive more aesthetically pleasing designs as easier to use than less pleasing ones, according to Mark Boulton (Boulton, 2005).

Mark Boulton used the analogy of 2 cars, a Skoda Octavia Estate, versus an Audi A4 Avant, both alike in every way, except design.



Images: <http://www.markboulton.co.uk/journal/comments/aesthetic-usability-effect>

It's fair to say that most of us would choose the Audi just because it looks better. It makes one feel as if the Audi would drive better than the Skoda. Unfortunately, once people have that type of impression, it gets very hard to change.

As we mentioned earlier about usability and user impressions, pleasant aesthetics would give the user a sense of trust that you would be able to assist them to complete their tasks with the same kind of high standards as you had when designing the site. It exudes professionalism. It goes without saying that things that are easy on the eyes also increase attention span making one want to explore further. This is especially important for e-commerce sites because the more time spent on the site, the higher possibility of purchases going through.

### **Easy to learn**

*Supports **initial learning** and **continued learning***

A product may be used just once, once in a while, or on a daily basis. But learning continues for as long as the usage lifespan of a product because users may need access to new functionality or modifies their requirements in future. An "easy to learn" interface allows users to build on their knowledge without any extra effort. This goes beyond the effectiveness of simply providing informative guides for difficult or advanced tasks, but allowing users to build on not only their prior knowledge of computer systems, but also any interaction patterns they have learnt through use of the existing application.

Well, it's not about creating boring and similar looking pages just so the user wouldn't have to relearn any interactions they have been accustomed to. No one likes to be called predictable. We all would like to create things that differ from the norm to be able to stand out from the masses. In computers, security algorithms should even more not be predictable. However, being predictable in web page design can actually be an advantage (Costa, 2009). How many times has one been frustrated when the design of a website we are used to changes? Search boxes should be at the top right and navigation bars should be in the centre across the screen or on the left. In

other words, there should not be any ambiguity about where and what they should click or move to next.

## 4.2 Striking the Balance

Now after studying the five dimensions of usability, you might be wondering if they are equally important. Well, it depends on the kind of applications you are developing, as well as your target group of users.

Let's take a museum website for example. For an informational website like this, the leading three aspects you should be concerned with are engaging, efficiency and effectiveness. The ultimate aim of the museum website is to have visitors making their way down to the museum exhibitions. So, they have to gain attention and make the site as pleasing/satisfying to use as possible. That is being engaging where the first impression of the web site would determine if the users would visit their galleries. Now, we all know attention span is rather short for most users on a website. Thus, you have to be efficient in conveying information such that the visitors can get any info they want quickly. Certainly, this information has to be accurate as well, which explains why the effectiveness.

Now let's look things from a different perspective – a registration form. Here, things are somewhat different where you should be more concerned with error tolerance, easy to learn, and efficient. For forms that require user interactions, error tolerance is usually the top priority because we cannot afford any “screw-up”. You must ensure a valid registration in this case, and provide help in recovering from any errors that do occur (E.g. input validation). Besides being error tolerance, the form needs to be easy to learn, which means users do not have to consult the help centre, user manual, or even go for training just to know how to use it. The time taken to fill this form should also not possibly exceed the time to actually go down to the physical site itself and have the customer service registers for the user. It certainly has to be efficient.

From these examples, you have seen how balance among the 5E's can set the direction for interface design. But, knowing these aspects of usability is only the beginning of grasping what usability “depends” on. It takes practice and effort in understanding the users to ultimately achieve usability and to this end, you will study in the following [section 5](#): usability testing – the way to understand users.

## 5 USABILITY TESTING

So after taking all the usability factors into consideration, one wonders if the website we've designed meets the required level of usability. Thus, we need to consider some testing to see how effective the website really is.

Usability testing is different from the usual black-box, white-box, or unit testing in the sense that we are not trying to find functional problems. These should have already been completed earlier on. It's also not about doing a demo for the users and proving that your website works. What we want to achieve here is to be able to learn more about potential users, know what they know, how they think, react and what they actually want to get out of the website. Using that information, we are going to tweak the website to correct usability problems.

Now, it is important to note that we refer to the 2 groups of people differently, customers and users. Usability testing should never be conducted with the customer or a fellow programmer/designer. Let's say you create an e-commerce site for a company. The customer would of course be the one who hired you, but the potential users would be almost anyone interesting in buying something off the site. Here we discuss what Soren Lauesen writes in the book, *Usability in User Interface Design: A Software Engineering Perspective* (Lauesen, 2005), and consider what it takes to plan an effect usability test.

## 5.1 Getting Started

### Users

We first need to consider who we are going to invite for the testing. Potential users' knowledge and demographics can be spread over a wide range. We would at least need to know some statistics of the potential users to be able to conduct effect tests. If the majority of potential users are going to be university students, we would expect their knowledge of computers to be higher than if it was a 60 year old grandmother. Knowing this information allows us to tweak the test tasks slightly differently to be able to fit the correct group of people.

Next, we also need to consider the number of people we want to test. Do we only get a few people, or invite as many as we can at the start? You may think that we would need to invite lots of people from the get-go, but is it possible that we only need to conduct the test on 1 person at the start? Lauesen reveals that it is actually better to start with a single person as most usability problems do not actually differ too much between potential users. What problems one experiences with the website would be shared by others. We then slowly increase the number of people we test each time. In keeping the number of users small and gradually increasing it, we are also not overwhelmed with problems to solve. We should also prioritise problems to see if they were indeed shared by many people, a small group or just 1 or 2 people. We may think about not solving the problem if too little people experience it, as any "improvements" we make might inadvertently cause a problem for someone else. In this way, we can improve our website more efficiently.

During the test, we would require the user to *think aloud*. As mentioned, we are interested in how the user thinks and interacts with the website. Thus, we would like to have the user make his or her thoughts audible so that we can be walked through the thought process the way they see it.

### Tasks

When designing tasks for the test we have to ensure we create proper and full ones. That means no made up names from cartoon or superhero characters should be used. We also want to ensure that the user is able to experience a few functions at a time. So, instead of just giving the user a task to "log in", we tell them to "buy a (specified) book off the website". As you can see, the second task allows us to watch the user navigate through the website as it involves so much more than the first task.

We should also be careful not to given hidden hints that would affect the way the users interact with the site. Instead of saying "search for the book, Romeo and Juliet, put it into the cart and check it out" we should just say "purchase the book Romeo and Juliet". This way, we can see how the user thinks and reacts to the navigation around the site. We do not want to plant the steps to the task into the users' mind. We want them to think for themselves, allowing us to see if the website is as user-friendly as we want it to be.

### Time allocation

It is also important to plan out the amount of time we spend with each user. Such usability tests are usually paid experiments, so we would not like to waste precious time with the users. We also do not want to give the users the feeling that we have not planned a proper test or experiment.

Thus, we would detail how much time is to be spent interviewing the users and explaining the procedures. We should also limit the amount of time that a user is to spend on the tasks. If the user exceeds the threshold, we would also deem that to be a usability problem as the task was not able to be completed efficiently enough.

### Role allocation

With usability testing, we do not need to overwhelm the user with a huge group of testers in lab coats waiting to note down the user's every move. Lausen recommends the optimal size of a 2-3 member team to conduct the tests with the user. We would like to make the user as comfortable as possible so that they would be able to go through the test in as normal an environment as possible.

We then allocate these 2-3 members different tasks, namely:

- Facilitator

The facilitator's job is to be the main point of contact with the users and make them feel as comfortable as possible. The facilitator would conduct the opening interviews and end debriefing. During the tasks, if the user is not able to project his or her thoughts, the facilitator can help by asking the user questions like "what are you thinking now?" or "where do you think you should proceed next?" to help the user open up for the test to be more effective.

- Log Keeper

As the name states, the log keeper is there to record down every action by the user. If there was a problem, moment of hesitation, area stuck or even a slight mis-click it could be a potential problem in usability. The log keeper would have to be alert to ensure nothing is missed.

- Observer

The observer is to assist both the facilitator and log keeper and act as a backup person in case anything was missed by either of them



### Methods of recording

We can always use trustworthy pen and paper to record down problems or areas where the user gets stuck for longer than we expect. But there are also other ways like the use of audio or video recording to tape down the entire process so that it can be replayed for re-evaluation later (be sure to seek the user's permission first though). One can also consider software or online tools like [Chalkmark](#) to detect the areas where the user has clicked.

## 5.2 Measuring Usability

In [section 4.1](#), we looked at the five dimensions in the usability world. If we want to say how far we are in achieving these dimensions, we must be able to measure the factors.

### Task time

(Adapted from *Usability in User Interface Design* (Lauesen, 2005))

The classical way of measuring usability is to calculate the time taken by the users to carry out their tasks. It can measure many usability factors such as efficiency and ease of learning.

When you want to buy a system, you can run usability testing to see how fast the users can learn to complete their tasks. In most situations, such existing system is already widely in used. So, you might even get hold of experienced users and measure how fast they carry out their tasks. However, if it is a new system that you are developing, using time as a metric is less suited in practice. This is primarily because a working version of the system is required to measure task time, which is only possible in a later stage of development, and at that point it is already very expensive to change the system to improve user performance. Besides, you would not have experienced users for the system, which means you cannot measure the level of efficiency or ease of remembering.

Most often when dealing with task time, the number of mouse clicks or keystrokes are used for measurement. However in principle, we do not care much about the numbers but the resulting total task time. That means mental processing time is the determining factor you should be concerned with. This means, you will need to find more users to participate in the usability test because mental processing time differs across individuals, and the more participants involve in the test the more accurate the result.

### **Number of problems**

**(Adapted from *Usability in User Interface Design (Lauesen, 2005)*)**

Instead of measuring task times, we can also use think-aloud tests and record a list of all the usability problems encountered by the users, which gives a better feedback to developers, and this can be carried out early in development with a user interface prototype. Here, we are interested in the total number of problems faced by the users during the test. So, what kind of problems would a user possibly encounter? For example, if you are to run a usability test on a purchase of car insurance, users might face the following problems:

- User faced problem in finding out the car insurance calculator so he used the site search feature
- User is frustrated because 'compare insurance quotes' page take too much time to download
- User is not sure whether he is going to get the best deal as he is not satisfied with the insurance quotes
- User got frustrated as he had to enter the email address twice while checking out
- User doesn't know the chassis number of his car so he can't make the purchase

Source: <http://www.seotakeaways.com/pocket-guide-to-usability-testing/>

One interesting point to note is the word "frustrated". The emotions level of the participating user should be observed at all times as they can be indicative measures of a problem. If the user is suddenly feeling angry, frustrated or even sad while doing a task, something must be wrong somewhere! This is why you should record the incident down for further analysis.

### **User opinions**

**(Adapted from *Usability in User Interface Design (Lauesen, 2005)*)**

Gathering opinions from users are the classical way of measuring the level of satisfaction. This is usually done in the form of surveys or questionnaires. With this technique, you can ask users about their opinions using a scale, for example, on simple scale: [agree, neutral, disagree], or a more precise scale of more steps: [agree, partly agree, neutral, partly disagree, disagree].

Surveys/Questionnaires are typically segregated into two sets: Pre-test and Post-test. As the name implies, pre-test surveys are done before usability testing starts and are after the opinions a user has before even using the system. For such surveys, they usually comprises of psychological or “get-to-know-you” questions that aim to determine which user group does the participant falls under based on the ways they perceive things.

Post-test surveys, on the other hand, are done after the user has used the system or complete a certain test task. We can ask about the user’s opinion on many matters like the following example:

Questionnaire	Agree	Neutral	Disagree
The system was easy to learn			
The system is easy to use			
The system helps me to ...			
It is fun to use			
I will recommend it to others			

In general, we would ask many users to complete the surveys/questionnaires in order to get a more accurate statistical result, but this is certainly only beneficial when the system has been widely used for some period of time.

Many companies consider opinion from users as the best measure for usability, which gives consistent results from user to user. This may be true; however, you must also note that there is little correlation between subjective satisfaction and objective performance. Subjective satisfaction is heavily influenced by factors that are out of developers’ control such as management style or discounts/promotions. Another issue is that they are difficult to measure during development and we get more reliable results only after the system is widely used. If the results show that a high percentage of users are satisfied, we cannot be sure that the system is indeed easy to use. If the results show otherwise, we still do not know what the problem is and we have to talk to the users to find out what the real problems are.

### Expert opinions

(Adapted from *Usability in User Interface Design (Lauesen, 2005)*)

Opinions, other than from users, can be enquired from an expert as well. This is more commonly known as usability evaluation, which is based on the set of guidelines the expert has. Typically, you would ask a usability expert to give critiques on the user interface and identify the deviations from the guidelines. It is always better to have a minimum of two experts reviewing the interface and agree on the significant deviations.

Although following guidelines is a good practice, it does not necessarily guarantee high usability. In most cases, the percentage of hit rate is only 50% (Lauesen, 2005) - that is half of the actual problems being predicted with another half being false problems. In other words, this is certainly not enough to ensure the system usability – you still have to find users to participate in usability testing. Besides, experience shows that it is quite hard for developers to follow the guidelines closely, especially when they consist of dozens of pages. This is why in [section 6](#); we will selectively compile a list of the top 10 guidelines built on examples for easy reference.

### 5.3 After Completing the Test

The job is certainly not done once the user finishes the tasks. We also have to debrief the user and do the report analysis on the findings of the test.

#### Debriefing

The facilitator will conduct a debriefing with the users to finalise their thoughts on the entire website. If there was anything missed out during the test or if the users have suggestions how they would like to see the website improved, the facilitator can make use of this time to get these information from the users. It would also bring the entire testing process to a suitable end.

#### Report analysis

Since the testing team only comprises of 2-3 members, it is also possible that the design team is made up of a more people. In any case, Lauesen recommends that the testing team consolidates and present their points to the rest of the team within 12 hours. It is essential not to wait too long after the test before consolidating the points as the mind is not able to retain information for too long. Essential bits of usability information may be lost after the 12 hour mark.

After going over the points with the team, it is also important not to head straight into solving the problems. The team should classify the problems into different areas like response or navigation issues and also like we mentioned earlier, prioritise them into levels of severity. The team then decides what needs to be solved immediately or can be further tested. In this way we can achieve solutions more efficiently and effectively.

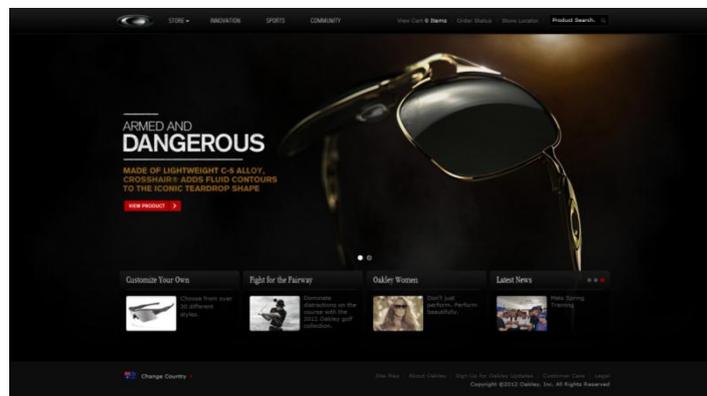
## 6 COMMON MISTAKES AND GUIDELINES TO GOOD WEBSITE DESIGN

There is always value in reminding ourselves of the common mistakes in usability design. So now that you have learnt about usability, let us take a look at the list of design stupidities that irked users the most along the years and what can you do to avoid/improve them.

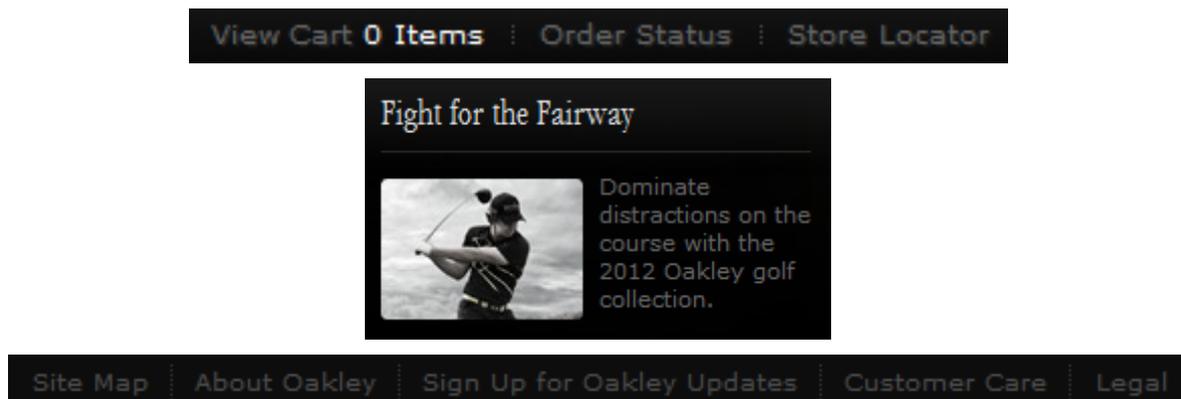
#### Writing legibly

(Adapted from *What's The Best Font For Websites and Blogs?* (Bravery, 2011))

Bad fonts! Be it font type, font size, or font colour, readability is important! Let's take [www.oakley.com](http://www.oakley.com) for example. Here is Oakley's homepage for the sunglasses people:



If you like black, you probably have found your website! But if you actually want to read text, be prepared to strain:



Low contrast between text and background is a no-go! The hyperlink's text only gets highlighted to white on mouse-over but what's the point when you even have problem locating it from the dark.

The bottom line is – Your font choice can make or break your website! So what should you take note of? First, you have to choose a font that is legible and web-safe. Remember: a font will display differently in print than on the screen.

There are four types (or families) of web-safe fonts:

- Serif
- Sans serif
- Fantasy or cursive
- Monospace

Serif typed fonts are only good for prints, and Fantasy or Cursive fonts are too difficult to read in chunks. Therefore, when it comes to web, Sans serif is the best font type to use as they are clearer, crisper and bolder. Monospace fonts, commonly known as typewriter text are typically used for displaying codes samples on the web.

In short, use sans serif for online, serif for print, monospace for code, and fantasy for accents. The following are some good examples of Sans serif fonts:

<b>Arial, Helvetica</b>	<code>font-family: Arial, Helvetica, sans-serif;</code>
<b>MS Sans Serif, Geneva</b>	<code>font-family: 'MS Sans Serif', Geneva, sans-serif;</code>
<b>Trebuchet MS, Helvetica</b>	<code>font-family: 'Trebuchet MS', Helvetica, sans-serif;</code>
<b>Verdana, Geneva</b>	<code>font-family: Verdana, Geneva, sans-serif;</code>
<b>Lucida Sans Unicode, Lucida Grande</b>	<code>font-family: 'Lucida Sans Unicode', 'Lucida Grande', sans-serif;</code>
<b>Tahoma, Geneva</b>	<code>font-family: Tahoma, Geneva, sans-serif;</code>

When it comes to web and fonts, simple is always better. So don't use more than 3 fonts and try not to change the font in mid-sentence.

Now, when we come to deciding the size for fonts, don't set it too big or too small. Users are neither blind nor have microscopic eyes. If the users can't read your website, they'll just leave before even using it.

The most common trick to adjusting font size is to first set a general font size in the body selector of the CSS style sheet like `Body { font-size: 100%; }` In general, setting the base size to 1em or 100% is similar to setting the font size to user's preference. Once set, other base font size can be set by [absolute](#) sizes, [relative](#) sizes, or percentages that will increase or decrease based on the original size set in the body selector.

As for the colours, the main focus is on the contrast between the font colour and the background. You want to avoid making your users squint or feel the need to put on sunglasses. In a nutshell, use dark-coloured text on an off-white background or vice versa.

### Saying NO to flash

Most of the flash that users encounter on the web is bad flash with no purpose other than annoying people. Let's take [www.arcticsunwest.com](http://www.arcticsunwest.com) for example:



First things first – Come on! Splash screens and Flash intros are almost extinct. They are so bad that even the suckiest web designers wouldn't recommend it. This is a flash-based website but in fact, there's very little here that requires the use of flash except for videos.



And for most of their pages, you can't even select the text! Moreover, it's damn annoying to read text while there's something spinning/moving around.

Honestly, flash isn't the best solution for websites! In fact, Adobe has already announced the [roadmap for Flash](#). Use it for what it's meant to be used – games and videos. Besides, now more than ever, there's jQuery that simplifies JavaScript if you want dynamic content.

Flash should not be used to simply add zest to a page. If your content is dull, rewrite it to make it more exciting and hire a professional photographer to shoot better photos. Don't make your pages move or include irritating and useless animations. It doesn't increase users' attention, but drives them away!

### Writing for the web

We are not writing research paper or a hundred page thesis. Let's take [www.walloftext.net](http://www.walloftext.net) for example:



The domain name suits this kind of website perfectly – wall of text (WOT) it is! Well, even though it's a blog entry, certainly most visitors will go “too long; didn't read” (tl;dr)

We are writing for the web and that means your content should be short and sweet, scan-able and straight to the point. A wall of text is deadly on web usability – intimidating, boring, and painful to read. Keep it short. To draw users into text and support “scan-ability“, use these tricks:

- Sub-headings
- Bulleted lists
- Highlighted keywords
- Short paragraphs
- Simple writing style

### Proper Search-Box

(Adapted from *Designing The Holy Search Box: Examples And Best Practices (Fekete, 2008)*)

Search is a fundamental component for good usability and it is an important part of any big website. Certainly, it can't look like the following:



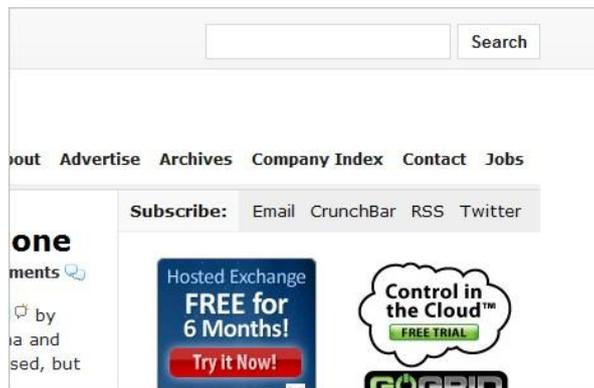
Source: bridge55.com from year 2008

Can you even locate where the search box is from the above website? NO, it's playing hide-and-peek with you! Even if you find the box, it's hard to see what you're typing in the input field. The color contrast certainly doesn't help!



Source: Wikipedia from year 2008

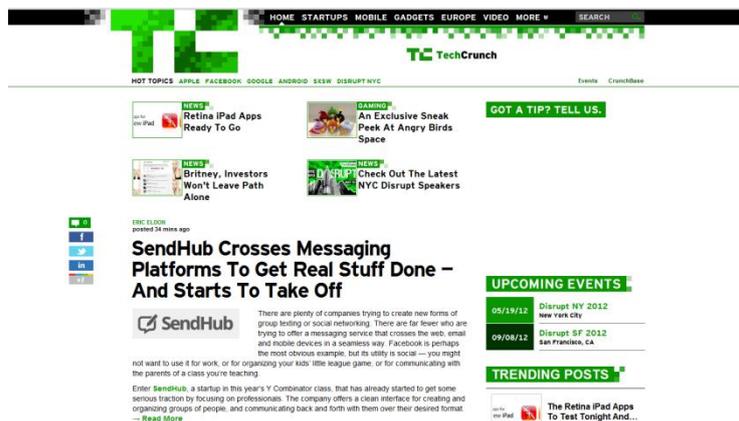
Is that a joke? Am I supposed to figure out which button to click – “Go” or “Search”? Like Steve Krug said, “Don't make me think!” (Krug, 2005 (2nd Edition)) That also means – Don't give users ambiguous options like “Go” and “Search”!



Source: TechCrunch from year 2008

No doubt, the colour scheme of the search box perfectly matches the website but there's a problem! The “Search” button, in particular, blends in just well that at first glance it's really hard to see where the search box is. It doesn't stand out and is not easy to spot – thus easily overlook!

Let's see how the website gets by after 4 years:



Source: TechCrunch from year 2012



It just got even worse! The box is not even clearly visible except for the word “SEARCH”.

**Search box = Input Field + Button**

Don't be lazy and replace that search button with a shortcut "Return" key. It's not usable at all – users may be wondering where the button is!

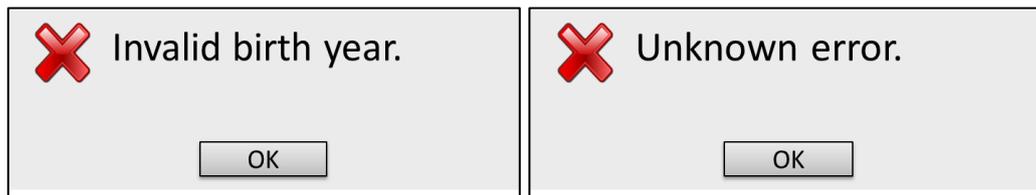
Also, take note of the text displayed in the input field. They must be readable at all times. So, use contrast and give a reasonable length so that it can accommodate multiple words without obscuring parts of user's query.

In general, the search box must be clearly visible, quickly recognizable and easy to use! The reason why a search box should be a BOX is simply because your visitors don't read page; they scan it. When users want to search, they usually scan the page for the most common pattern – a wide box and a button. So try to avoid any kind of design such as linked text or a button without a box.

### Sensible Error Messages

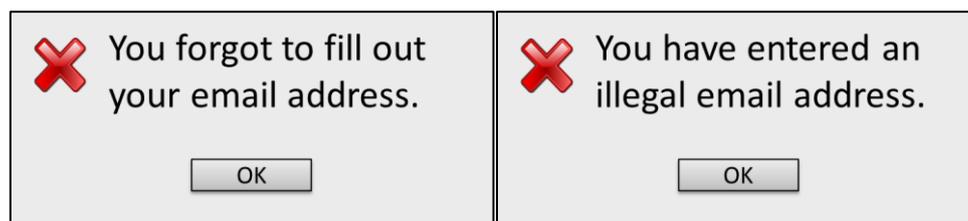
(Adapted from *4 rules for displaying error messages from a user experience perspective (Nomensa, 2010)*)

Bad error messages suck! Simply put, a bad error message is one that doesn't make sense to anyone but the person who programmed it. To have a sensible error message, first you have to be specific. Don't ever try the following:



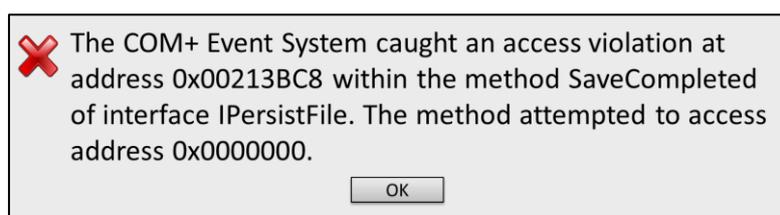
Instead, you can say something like "Please enter your birth year in 4 digits (E.g. 1987)". Give users more details about what went wrong and what they can do to fix it. The more specific your error message is, the easier it is for users to fix the error.

Next, don't ever blame the user. Try to be polite! Don't use words like illegal, or make it sound like the user has done something wrong.



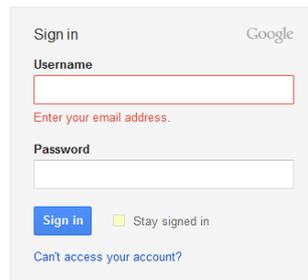
Instead of the above examples, you could simply say things like "Please fill out your email address." or "Please enter your email address according to the format (E.g. john@gmail.com)". If the user has made a mistake while filling out your form, it's not a good idea to rub his/her nose in it. Don't tell users what they did wrong but tell them what they should do to get it right.

Now, another point is to use a "user" language. This is the last thing you want your user to see:



This is way too complex. Always try to hide the complexity and show a message that would appear more user-friendly to a non-programmer. Instead of the above example, you could simply say something like “Sorry, your file was not saved. The hard drive was full.”

Now, how about placement of the error messages? Let’s take a look at the following:



Source: <https://accounts.google.com/ServiceLoginAuth>

For smaller forms like the above Google login, the usability issue may not be large whether the error message is above, to the right, or below. But for longer forms like the following, it’s a disaster:

Title:

First name:

Name cannot be empty

Last name:

Email Address:

Please enter valid password

Passwords:

Confirm Password:

Address line 1:

Enter a valid address

Address line 2:

Postcode:

Users will be confused and wonder which name is it, which address is it? So, it is better here to align the label with the error message.

In general, error messages are best to be above the field because it suits the user’s reading pattern – from top to bottom. Like the following:

Email: ✘ Please fill out your email address.

So how can you make good error message that’s sensible? According to Jakob Nielsen (Nielsen, 2005), good error messages should:

- Clearly indicate something has gone wrong
- Be in a human-readable language (NOT application error or even SQL error!)
- Be polite and not blame the users
- Describe the problem
- Give constructive advice on how to fix the problem
- Be visible and highly noticeable, both in terms of the message and how it indicates where things went wrong

- Preserve as much of the user's work as possible so that they don't have to do everything over again
- If possible, guess the correct action and let users pick it from a list of fixes
- Educate users by providing links to pages with an explanation of the problem

In layman's terms, error messages should not just say "Sorry, something went wrong". They should also say "Do this or this to continue", or even better - allow you to solve the problem immediately. When an error message is displayed, you have already wasted time. And when the message is not helpful in aiding you to complete the task you set out to do earlier on - it sucks! (Schlipf, 2008)

However, one point to note is that errors might still occur because you did not predict the complete possibilities on how a user might interact with the application. Therefore, when designing an interface, make it difficult to take incorrect/invalid/irreversible actions and plan for the unexpected. For example, avoid any technical jargons and use distinct languages that are understandable by the users. To ensure correctness/validity of user actions, always limit choices to those which are correct and provide clear examples for data entry. Last but not least, provide a means to undo or reverse actions instead of using confirmations - users become insensitive to them. Also, not to forget the most important point to note: Be polite about correcting mistakes!

### *Using Pagination Correctly*

Pagination is a good function for splitting up items onto several pages, helping to enable the page to process and download more quickly. It is especially useful when having to display multiple items, for example, products in a store, a few pictures or search results. In addition to increasing the efficiency of page loads, this also helps the site to look neater (Fadeyev, 2009).

Unfortunately, many designers often abuse this practice. We often see news articles that are being unnecessarily spread over 2 or 3 pages. In most cases, this is done just so that site can receive more page views, enabling them to earn more from advertising. From a user's point-of-view, the main purpose of clicking on the article link is to read its contents, but it's just plain frustrating to have to reload the entire page just to continue reading the article. Having to click through a few pages just to find the needed information is not just hard on the user, but for the search engines as well. As each page holds lesser tags or keywords about the topic, pagination might inadvertently have a negative impact on its ranking on the search results.

The guideline is simple, use pagination only when the need arises. If you would really like to split up your article, split it up for a good reason, for instance, if the article shifts focus to a different area. However, give the links to the other sections on top or at the side for easier navigation, don't make the user read through the entire article only to find out that the information is on another page. Or else, that may just be the last time that user ever wants to come to your site.



*State page sections clearly at the top*

The same concept can be used for search results, if possible, group search results with similar keywords together. By creating subcategories for the results, users also won't have to scroll through tons of results just to get to the most fitting one (Fishkin, 2010).

So do your site and it's users a favour, please use pagination wisely.

*Keeping registration simple and to a minimum*

Do you really need to know my address, postal code, place of work and job designation before I can have create an account with your site? I mean what are going to do with all the information of mine? All I want to do is post a review or comment on your product, why do I need to sign up?

If you've ever found yourself asking one of the above questions, then you have unfortunately been a victim of poor registration form planning. Earlier, we covered error messages being shown during the sign up process, but what we want is to reduce the need to enter information, thereby indirectly eliminating the possibilities of bad error messages. Thus, a registration form with many fields is more susceptible to errors. The registration process takes more time and becomes increasingly frustrating. Also, if an error occurs during the "submit" function, the user will have to re-enter everything from the start!

Look, the crux of the matter is this; the internet was created so that people could upload whatever information they **wanted** to share with the world. Note the key word, wanted. It was the quick and easy way one could tell the world of a new discovery, knowledge of a product or just some social interaction. Designers these days forget that users go to their sites to get information, not give it. With the abundance of sites with similar information these days, an over zealotness in wanting to get users to sign up, might just end up creating a barrier and chase users away. Users come to site because they want information from you, and not to give it to you (Owoh, 2010). It is also added effort on the user to remember the username and password (Fadeyev, 2009), and sometimes it just isn't worth it.



The image shows a registration form for 'ReadGalore'. At the top, it says 'ReadGalore' and 'Please register to view this article'. Below this, there are two input fields: 'Email' and 'Password'. A red arrow points to the 'Email' field. At the bottom of the form is a 'Submit' button.

*Making users register just to read something isn't a very good idea*

With all the news about identity thefts and personal details being hacked or sold to third-party companies for email spam, users these days are reluctant to give too much personal information out. As you can see from the \$300 million dollar case earlier, this error caused a company huge losses. So, don't lose the possibility of a good review or comment being posted just because you wanted them to sign up first.

The fact is users will sign up if they want to, thus, before you embark on creating that registration form, remember 2 things, what do you really need to know, and is it really necessary (before other tasks can be completed).

### **Bad Navigation**

It's a really simple concept that everyone knows, but the actual execution can be a tad difficult. In the past, websites used to be faulted for having navigation all over the place. These days, it's nicely placed at the top or at the side. However, the problem doesn't stop there. Vincent Flanders writes on his website, [webpagesthatsuck.com](http://webpagesthatsuck.com), that a site's navigation should be able to answer the following questions (Flanders, 2006):

Where am I? Where have I been? Where can I go next? Where's the Home Page? Where's the Home Home Page?

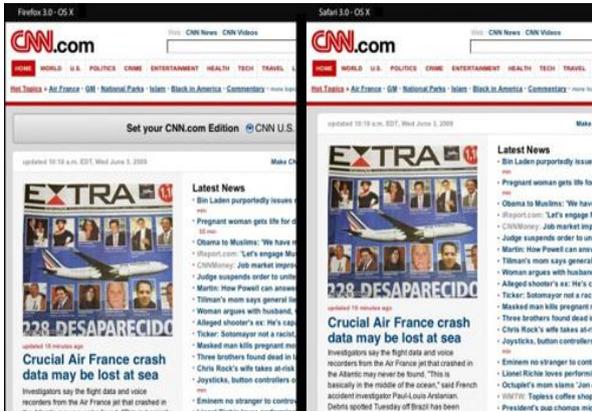
Sites commonly do not keep the style of navigation the same throughout and end up causing the user confusion while navigating. Links are also poorly worded and ambiguous. There is also no need to link back to the same page you are currently on. But the one biggest irritant of all, and yet so easy to rectify, are the frustrating 404 errors or dead links (more commonly known as "No, your page not found page is not cute at all!").

A good solution is to set the order of links from left to right or top-down based on users' needs, with the left or top link being the most commonly used ones. Links should also be textual as far as possible, but most importantly stand out from the content. If images are used, do provide text in the alt tag (Owoh, 2010). Ensure that links open in the same page, unless specified by the user, there's nothing more frustrating than having a new window pop-up (could be mistaken for an advertisement), and then making the "back" button useless.

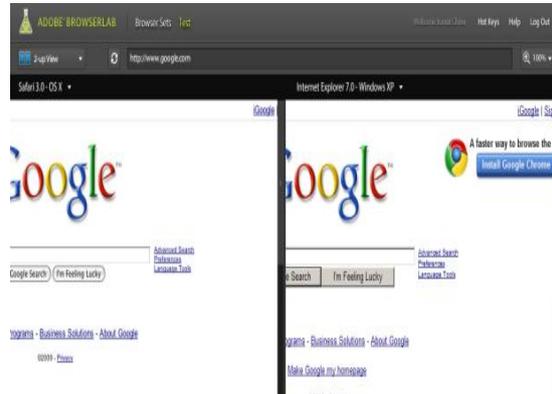
Most importantly of course, constantly check if links are still working. Rule of thumb is that most users will only stay for a maximum of 3 clicks, and if they still can't find what they are looking for, it's on to the next better site please.

Cross-Browser Compatibility

Internet Explorer, Google Chrome, Firefox, Opera – just 4 of the many browsers out there. But the thing is that they all interpret CSS and render pages slightly differently.



Source: <http://www.noupe.com/how-tos/a-guide-to-seamless-website-redesign.html>



Source: <http://googlesystem.blogspot.com/2009/06/test-your-site-in-different-browsers.html>

It gets worse when all the alignments of links and text are all out of position. So, here are some solutions to ensuring cross-browser compatibility.

- W3C standards

These are guidelines and specifications set out by the international community for browser developers to follow. By coding your website to these standards, you can ensure that there is a good chance that your website will display the same across all browsers. The W3C site also offers online tools to validate your codes for adherence to the standards.

- CSS reset

As you know, the default formatting of text is different across different browsers.



Source: <http://msdn.microsoft.com/en-us/magazine/gg257960.aspx>

The goal of CSS reset is really to take control of all default formatting tags so as to ensure that you do not have alignment problems caused by text.

```
8 <style type="text/css">
9
10 html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6
11 a, abbr, acronym, address, big, cite, code, del, dfn, em, font, img,
12 small, strike, strong, sub, sup, tt, var, dl, dt, dd, ol, ul, li, fi
13 table, caption, tbody, tfoot, thead, tr, th, td {
14     margin: 0;
15     padding: 0;
16     border: 0;
17     outline: 0;
18     font-weight: inherit;
19     font-style: inherit;
20     font-size: 100%;
21     font-family: inherit;
22     vertical-align: baseline;
23 }
```

Source: <http://www.designisphilosophy.com/css/why-a-css-reset-should-be-at-the-core-of-your-stylesheet/>

This would ensure that text at the very least doesn't cause any alignment problems for you across different browsers.

- Keep Testing

The best solution we can offer is to do constant testing. Sites like <http://browsershots.org/> allow you to view your site in a huge variety of browser and even in different versions. Also, try your best to avoid scripts as there's also no telling how they would respond.

### *Let Text Be Text and Images Be Images*

Having more images on a webpage doesn't make the site look better, it only adds to the distractions that will fill the user's eyes. Animated GIFs or Flash can be a cool way to represent a powerful message, but when overused, it gets on the nerves of the user, not to mention that it uses a lot of bandwidth to be downloaded (Owoh, 2010).

An important point to note is that, search engines and their bots crawl the internet looking at text (Flanders, 2006). So if most of your site is filled with pictures for the content, you can more or less be sure that no one would be able to find it. Also, if one wants to make additions or amendments, it would be a big hassle to do so.

A good guideline would be to offer alternatives to the users, or even the option to skip the animation if the user wishes to. Allow for alternate text to show also in the event that the images cannot be displayed properly.

## **7 CONCLUSION**

Usability, in short, is all about the users – to give them what they want! And what you have learnt in [section 5](#) is a way for you to understand your users through usability testing so that you can meet their needs and their expectations.

In [section 3](#), we have also seen how usability is important and beneficial not only to designers but programmers as well. Moreover, usability can help you progress towards achieving a good user experience, which is ultimately what most companies are chasing after over the years. Functionality can only bring you that far, but usability can bring you much further and closer to your users.

Hopefully at the end of the day, this chapter could serve as a good recipe that helps train you in cooking a usable interface along with the guidelines in [section 6](#)! We certainly look forward to the day you could also be a usability master who gives good opinions and critiques on any user interface! But before that, always remember a quote from Steve Krug – “Don't make your users think!”

## 8 REFERENCES

- Boulton, M. (2005, March 6). *The personal disquet of Mark Boulton*. Retrieved from Aesthetic-Usability Effect: <http://www.markboulton.co.uk/journal/comments/aesthetic-usability-effect>
- Bravery, R. (2011, April 2). *What's The Best Font For Websites and Blogs?* Retrieved from Integral Web Solutions: <http://www.integralwebsolutions.co.za/Blog/EntryId/832/Whats-The-Best-Font-For-Websites-and-Blogs.aspx>
- Costa, J. (2009, December 15). *Factors That Affect Usability*. Retrieved from Six Revisions: <http://sixrevisions.com/usabilityaccessibility/factors-that-affect-usability/>
- Fadeyev, D. (2009, February 18). *9 Common Usability Mistakes In Web Design*. Retrieved from uxdesign.smashingmagazine.com: <http://uxdesign.smashingmagazine.com/2009/02/18/9-common-usability-blunders/>
- Fekete, G. (2008, December 4). *Designing The Holy Search Box: Examples And Best Practices*. Retrieved from Smashing Magazine: <http://www.smashingmagazine.com/2008/12/04/designing-the-holy-search-box-examples-and-best-practices/>
- Fishkin, R. (2010, March 2). *Pagination: Best Practices for SEO & User Experience*. Retrieved from The Daily SEO Blog: <http://www.seomoz.org/blog/pagination-best-practices-for-seo-user-experience>
- Flanders, V. (2006). *Biggest Mistakes in Web Design 1995-2015*. Retrieved from Webpages That Suck: <http://www.webpagesthatsuck.com/biggest-mistakes-in-web-design-1995-2015.html>
- Krug, S. (2005 (2nd Edition)). *Don't Make Me Think*. New Riders.
- Lauesen, S. (2005). *Usability in User Interface Design: A Software Engineering Perspective*. Pearson Education Ltd, Addison Wesley.
- M.Spool, J. (2009, January 14). *The \$300 Million Button*. Retrieved from UIE.com: [http://www.ue.com/articles/three\\_hund\\_million\\_button/](http://www.ue.com/articles/three_hund_million_button/)
- Maven, R. (2007, May 15). *Interview with Jakob Nielsen*. Retrieved from Econsultancy Digital Marketers United: <http://econsultancy.com/us/blog/1192-interview-with-jakob-nielsen>
- Mazur, M. A. (2003). *Dimensions of Usability in Content and Complexity: Information Design in Technical Communication*. Erlbaum.
- Mifsud, J. (2011, July 11). *The Difference (And Relationship) Between Usability And User Experience*. Retrieved from UsabilityGeek.com: <http://usabilitygeek.com/the-difference-between-usability-and-user-experience/>
- Nah, F. F.-H. (2004). *A study on tolerable waiting time: how long are Web users willing to wait?* Lincoln, Nebraska: College of Business Administration, University of Nebraska-Lincoln.
- Nielsen, J. (2005, October 3). *Top Ten Web Design Mistakes of 2005*. Retrieved from useit.com: <http://www.useit.com/alertbox/designmistakes.html>
- Nomensa. (2010, July 1). *4 rules for displaying error messages from a user experience perspective*. Retrieved from Nomensa, Humanising Technology: <http://www.nomensa.com/blog/2010/4-rules-for-displaying-error-messages-from-a-user-experience-perspective/>

Owoh, B. (2010). *11 Common Web Design Mistakes (Blunders)*. Retrieved from Hongkiat.com, Online tips for designers and bloggers: <http://www.hongkiat.com/blog/11-common-mistakes-blunders-in-web-design/>

Schlipf, C. (2008, September 4). *Bad Error Messages Suck*. Retrieved April 10, 2012, from <http://www.yeap.de/blog2.0/archives/149-Bad-Error-Messages-Suck!.html>

## 9 RESOURCES

*Bi-weekly column by Dr. Jakob Nielsen*

<http://www.useit.com/alertbox/>

*Common mistakes in web design and how to fix them*

<http://www.webpagemistakes.ca/>

*How to write for the web?*

<http://www.useit.com/papers/webwriting/>

*W3C website*

<http://www.w3.org>