IDI Course Notes | P

IDI – Common Design Problems

Pere-Pau Vázquez, Dep. LSI - UPC

Contents

1 The problems of a bad design

1.1 Introduction

One way to improve your design skills is to learn from other's mistakes. Of course, you will also have to learn from your own mistakes, but this is more painful. Therefore, we will start with some web pages and applications that have notable design errors. Then, we will introduce a long list of the 10+ most common design mistakes, according to Jakob Nielsen, with some examples of the mistakes in action.

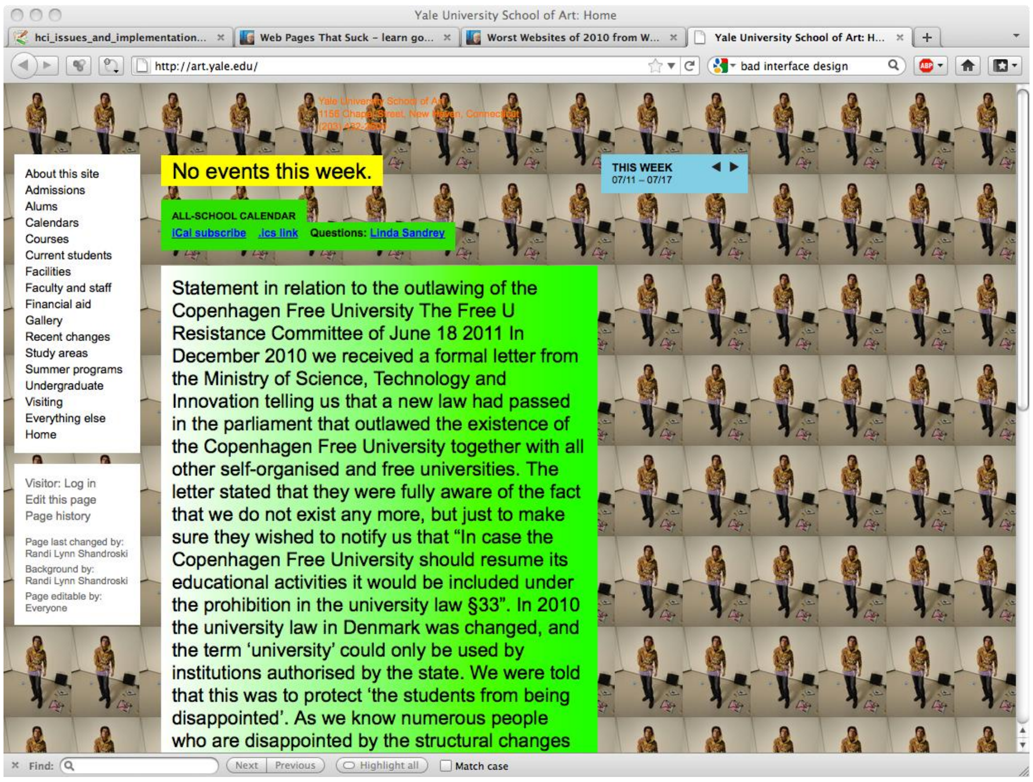
For the web pages, we will mainly consider visual design: We will show a snapshot from the page, and then point some of the problems. For the applications, we will also comment, when possible, on the behaviour, if it is also problematic.

1.2 Bad design in web pages

We all have experienced problems when using an application or webpage. If we are lucky, we will end up wasting a portion of our time and maybe will get angry. If we are unlucky, we may end up with less money than before and a rage attack.

As you may imagine, you can google for blogs or sites that show images or links to examples of bad design. One of the examples you can find, is one devoted to bad web pages: http://www.webpagesthatsuck.com/, from where we took the following examples.

1.2.1 Bad colours, fonts, and overall organization



No events this week.

Statement in relation to the outlawing of the Copenhagen Free University The Free U Resistance Committee of June 18 2011 In December 2010 we received a formal letter from the Ministry of Science, Technology and Innovation telling us that a new law had passed in the parliament that outlawed the existence of the Copenhagen Free University together with all other self-organised and free universities. The letter stated that they were fully aware of the fact that we do not exist any more, but just to make sure they wished to notify us that "In case the Copenhagen Free University should resume its educational activities it would be included under the prohibition in the university law §33". In 2010 the university law in Denmark was changed, and the term 'university' could only be used by institutions authorised by the state. We were told that this was to protect 'the students from being disappointed'. As we know numerous people who are disappointed by the structural changes

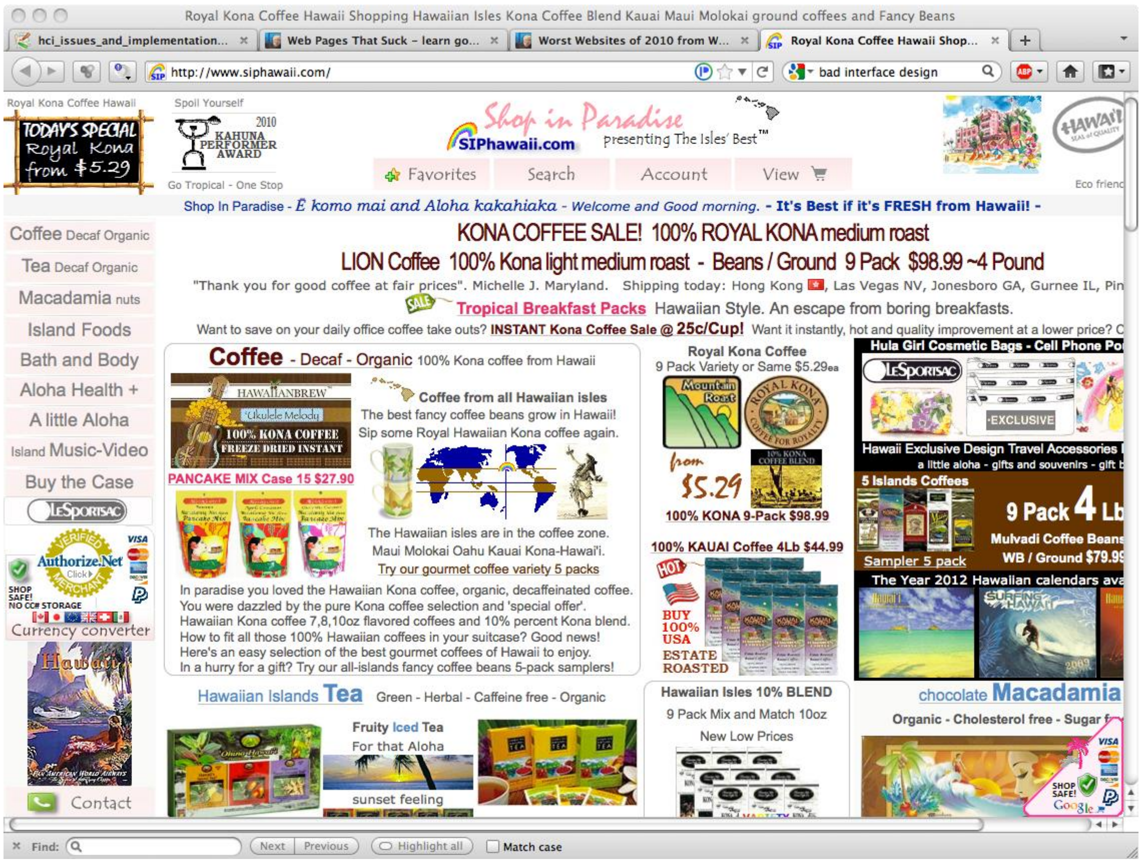
Figure 1: The Yale School of Art.

The first example is the webpage of the Yale School of Art, shown in Figure 1. You can visit the web by going to http://art.yale.edu/.

The page shown is full of flashing colours, images, and texts overlapping the images that are therefore difficult to read. It seems that the members of the institution can modify the page freely and this, together with the lack of a person officially responsible on the final look, has lead to this horror.

1.2.2 Where is white space?

The following example comes from the *Shop in Paradise* website, shown in Figure 2. This web can be visited through: http://www.siphawaii.com/.


Royal Kona Coffee Hawaii Shopping Hawaiian Isles Kona Coffee Blend Kauai Maui Molokai ground coffees and Fancy Beans

http://www.siphawaii.com/

Shop In Paradise - E komo mai and Aloha kakahiaka - Welcome and Good morning. - It's Best if it's FRESH from Hawaii! -

KONA COFFEE SALE! 100% ROYAL KONA medium roast

LION Coffee 100% Kona light medium roast - Beans / Ground 9 Pack $98.99 ~4 Pound

"Thank you for good coffee at fair prices". Michelle J. Maryland. Shipping today: Hong Kong, Las Vegas NV, Jonesboro GA, Gurnee IL, Pin

Tropical Breakfast Packs Hawaiian Style. An escape from boring breakfasts.

Want to save on your daily office coffee take outs? INSTANT Kona Coffee Sale @ 25c/Cup! Want it instantly, hot and quality improvement at a lower price? C

Coffee - Decaf - Organic 100% Kona coffee from Hawaii

Coffee from all Hawaiian isles
The best fancy coffee beans grow in Hawaii!
Sip some Royal Hawaiian Kona coffee again.

The Hawaiian isles are in the coffee zone.
Maui Molokai Oahu Kauai Kona-Hawai'i.
Try our gourmet coffee variety 5 packs

In paradise you loved the Hawaiian Kona coffee, organic, decaffeinated coffee.
You were dazzled by the pure Kona coffee selection and 'special offer'.
Hawaiian Kona coffee 7,8,10oz flavored coffees and 10% percent Kona blend.
How to fit all those 100% Hawaiian coffees in your suitcase? Good news!
Here's an easy selection of the best gourmet coffees of Hawaii to enjoy.
In a hurry for a gift? Try our all-islands fancy coffee beans 5-pack samplers!

Figure 2: Shop in Paradise webpage.

When you visit this website, the first thing you note is that there is not white space. All the text is cluttered. One can imagine that is a *trick* to be able to place a higher range of products in the same space. However, it is difficult to find anything in such a page.

1.2.3 Too much flash

One of the typical problems in web design is the abuse of Flash animations. Many researchers have reported this problem by now, and we may find a typical example in Figure 3. The website can be visited through http://www.vacaway.com.

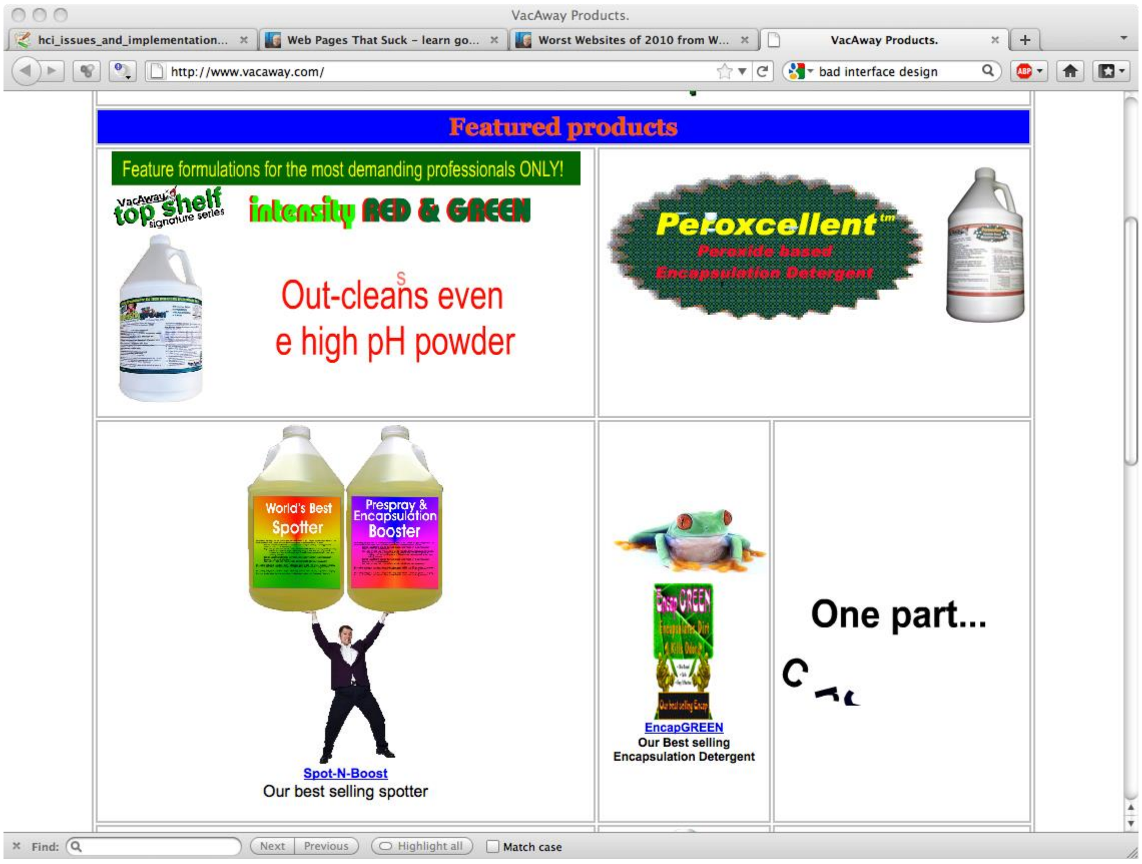


Figure 3: The webpage of Vacaway.

This webpage sells different of cleaners. The problem is that, when you visit it, you obtain a set of animations all around. It is completely uncomfortable. The animations are so distracting that you will probably not buy what you were looking for.

1.3 Applications with deficient UIs

Not only web pages suffer from bad design decisions. Applications commonly have some flaws. Since the arrival of smartphones and app stores, our exposure to a larger set of applications has grown. And, seriously, many of them are not designed properly. We will show here some examples of applications with design mistakes. Many of them are applications developed for portable devices, such as smartphones or tablets. Some of these examples and a lot more can be found at http://www.buigallery.com/.

1.3.1 Unnecessary introductory screen

In Figure 4 we can see the kind of introductory page that does is completely unnecessary. It doesn't provide any critical information, it doesn't distract you while data loads—it just brings a bad UI practice from the web to your iPhone, and it requires you to acknowledge it by tapping the *Open* button. This is a bad practice inherited from the old Internet: web pages with an introductory video or a flash animation that serves for nothing and the users only want to skip.

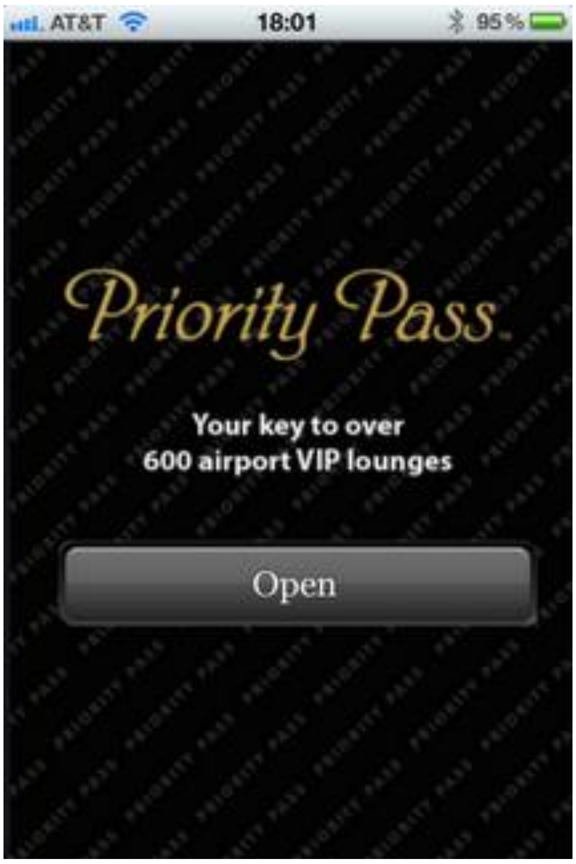


Figure 4: Priority Pass app for iPhone: Unnecessary introductory screen.

Apart from that, the button contains a label, *Open*, that has not been chosen properly: we are not going to open anything, better titles could be *Ok* or *Find a lounge*. Not happy with this, the button behaves in a non-standard fashion, as it does not visually respond to touch (changing colour or appearance) when clicked.

The best solution would be to eliminate this screen completely.

1.3.2 Widgets used incorrectly and with improper design

The old iPrint application for the iPhone was intended to let you print some work in any wireless Epson printer. Although simple, it had a higher number of design problems. The current version looks much better, with several design problems corrected. We can see an image of the previous version in Figure 5.

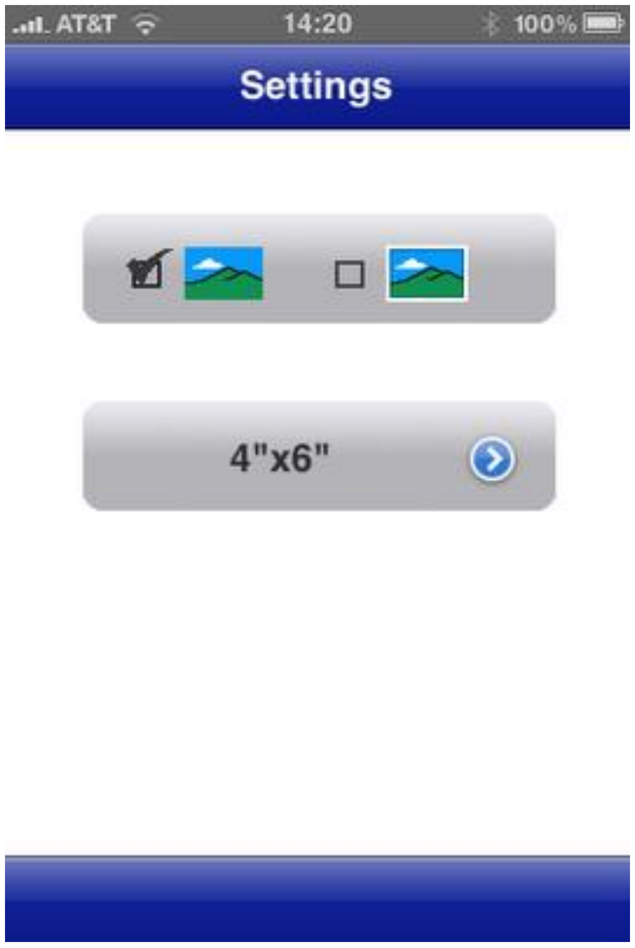


Figure 5: Epson iPrint app for iPhone.

This application has different issues:

- We cannot go to the *Settings* and *Select Printer* pages if we have not previously selected a photo.
- *Settings* and *Select Printer* pages do not provide a *Back* button.
- The options on the *Settings* page are designed like buttons, although they do not start actions. They should be separate, grouped lists.
- In order to toggle the *Border* option, you have to touch the *Border* "button" in two different places. This violates the expected behaviour of a control that is designed to look like a button.
- The check boxes in the *Border* "button" are graphical, instead of standard checkboxes, which is completely unnecessary for an iPhone app.
- The detail disclosure indicator next to the paper size does not bring you to anything related to changing a size.

1.3.3 Disrupting introductory screen

The Weather Channel app for the iPad (see Figure 6) shares some problems with the first example; it has an unnecessary, in this case beautifully designed, introductory page. It displays one weather-related picture out of a gallery with an accompanying large bar with six buttons. In order to start, you have to select one of those. However, there is little information the user can get from this initial screen that provides enough context for him or her to decide where to go.

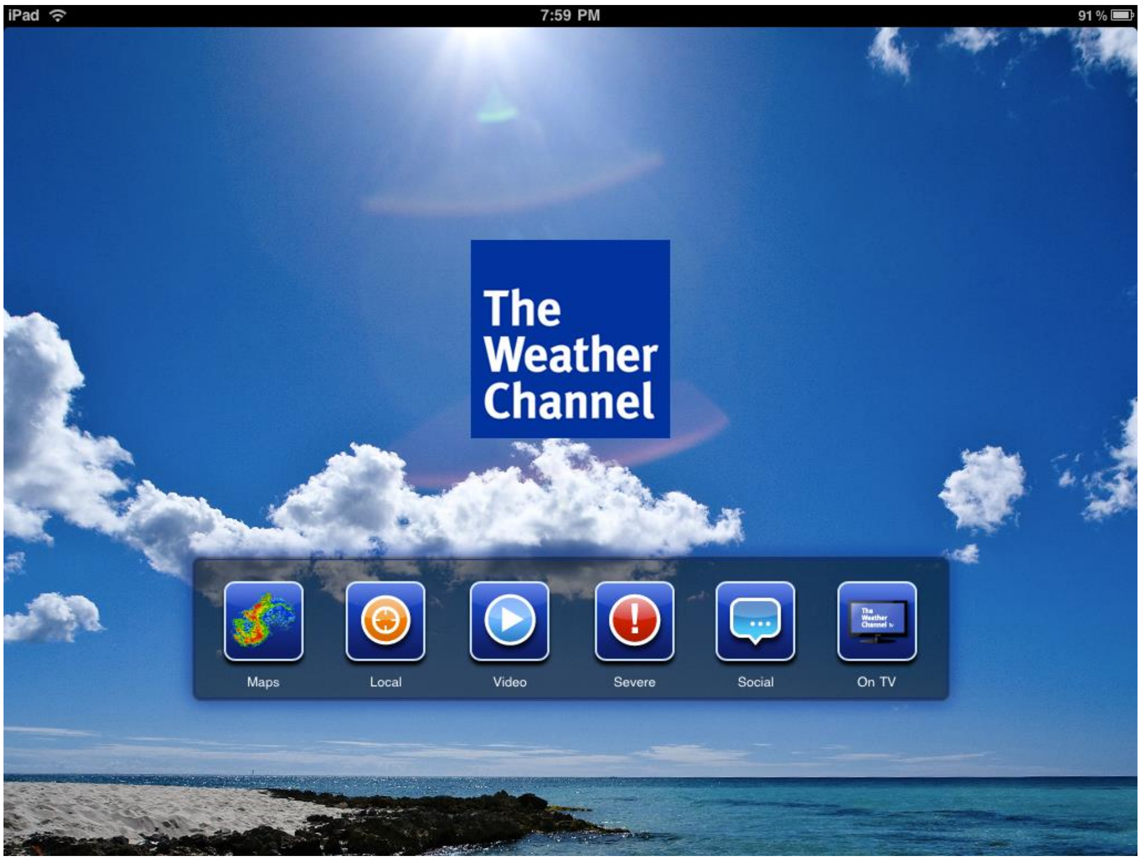


Figure 6: The Weather Channel app for iPad.

Being an app for weather information, the user will expect to see the local weather. Thus, the application should start there, with the information clear at the front and center.

Weather is something that changes continuously, and therefore, we often check it several times a day. Therefore, the application should bring the user to the last screen he or she was before closing, without requiring an extra step. This is, effectively, what Apple's iPad Human Interface Guidelines reasonably recommend. The branded image can be shown while data is loading.

There are other problems:

- It is difficult to know what differs from *On TV* to *Video*.
- Labels are not semantically equivalent; they mix nouns with adjectives, prepositions…
- *Severe* might be better renamed *Alerts*.

2. A list of 10+ common design mistakes

2.1 Introduction

The usability of applications improves when the users know how to operate the interface and it guides them through the working flow. Failing to accomplish the common rules usually prevents from both objectives.

Although the worst application design mistakes may be domain-specific, there are some common mistakes that plague lots of applications. The three main reasons for an application to fail are:

1. They do not solve the right problem.
2. They have the wrong features for the right problem.
3. They make the right features too complicated for users to understand.

Any of these mistakes may be critical for your application. In order to evaluate the application, one safe advice is to conduct user studies with representative or real users. Early prototyping and testing in order to get user feedback may avoid resources waste. Designing should be an iterative task, with several rounds of user testing in between.

The most usual way to get a wrong design is to listen to users. The best advice is to watch them rather than listening to them. Analysing how they work may be a better tool for a successful application design.

Despite this, there are a lot of general guidelines for user interface design that should be accomplished, or, at least, taken into account. Some guidelines are important enough that breaking them may doom your application or web site.

2.2 The 10+ common design errors

Next we review a list of common design mistakes.

2.2.1 Non-Standard GUI Controls

We have to take into account that the user will spend more time on other applications than our application of interest. Therefore, the user has a higher amount of *exposure* to a certain type of GUI controls. If those are common to many applications, we should not change them. Changing the appearance or behaviour of those controls will confuse readers. The best interaction designers have refined the standard look-and-feel of GUI controls over 30 years, supported by thousands of user-testing hours. It's unlikely that you'll invent a better button over the weekend. Even if your design is better, it will be only present in a single application. When seen in context, it will be an anomalous example: Users will always have thousand times more *exposure* to a different design. Users' cognitive resources are better spent understanding how your application's features can help them achieve their goals.

2.2.2 Elements that look a GUI Control without being one

It is the opposite problem to the previous one. Seeing text that looks like links or buttons will induce mistakes and confusion to the users. This has been present in several tools or webpages. See Figure 7. The “Please select a product” control looks like a link but it does not start any action.

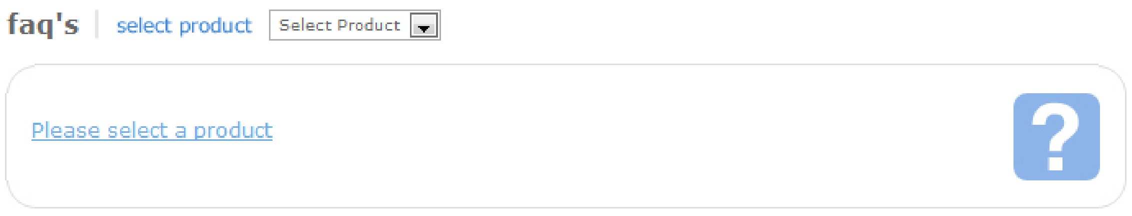


Figure 7: Text that looks like a link and it is not.

2.2.3 Inconsistency

The same words, buttons, or commands should be used throughout the same application for the same task. For instance, using “Save Model”, “Save File”, “Save Design” on different parts of the application for the same task will confuse the user and make him or her to feel insecure. Not only commands should be consistent, but designers should also make sure they the conventional rules of colours, shapes and forms to maintain identity of the interface or product. Striving for consistency will reduce errors, avoid confusion and improve efficiency.

There is an interesting experiment you can test by yourselves designed by Sakshat Virtual Labs in http://iitg.vlab.co.in/index.php?sub=72&brch=170&sim=862&cnt=1604. The objective of the experiment is to understand use of consistency of presentation in user interface design. You can see it in Figure 8. There is a car that starts moving when we press the “On” button, and stops when we press the “Off” button. We have to follow the instructions that appear on top of the screen, sometimes with inconsistency of colours, for example. The results of the experiment show that, even if you do not make any error, the reaction time is higher during inconsistency notifications.

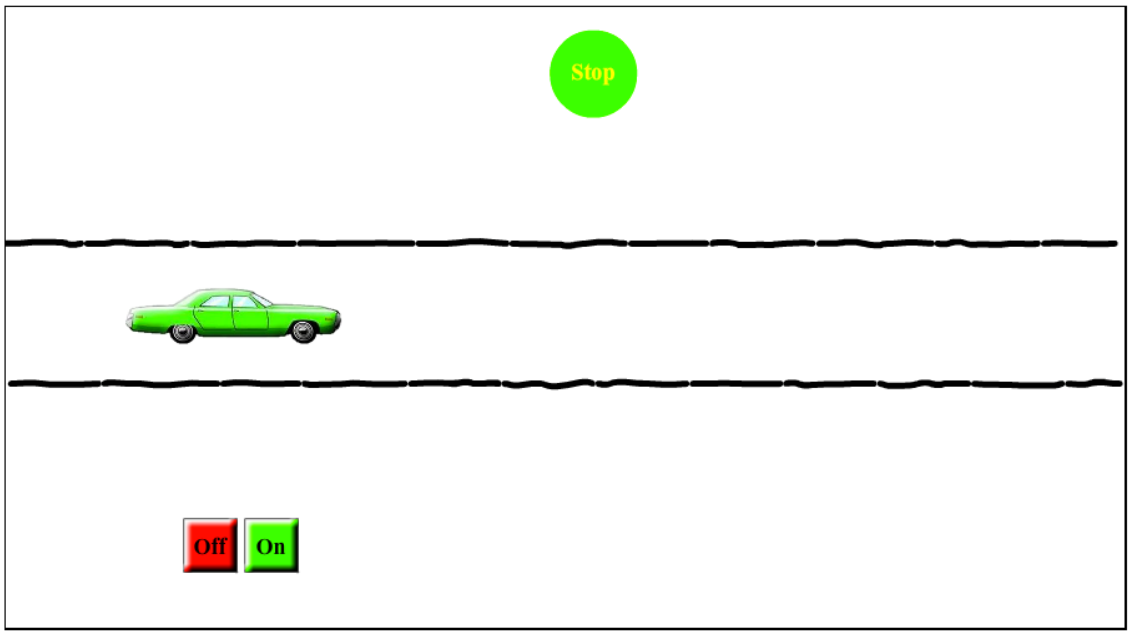


Figure 8. Experiment that demonstrates that inconsistencies in the interface leads to a higher number of errors or longer reaction times.

2.2.4 No Perceived Affordance

"Affordance" means what you can do to an object. Sometimes you visit a website and you do not know what to do there or where to go from there. The lack of familiar UI elements may produce this sensation. Perceived affordances are especially important in UI design, because all screen pixels afford clicking — even though nothing usually happens if you click. Therefore, one must avoid that the user requires clicking here and there in order to get an action. Action starters must be clear and comprehensible.

One example of element that may conflict with perceived affordance is drag-and-drop. Although in some applications (such as graphic design) the users may be familiar with this interaction mode, drag-and-drop elements in other UIs often do not provide any clue to the user that something can be done there, or what the final effect will be. In contrast, buttons and checkboxes are UI elements that are obvious to use. Sometimes this problem is solved with text, but, if the amount of required text is high, or it disappears after the first use, we should reconsider redesigning.

This was a common problem in old computers where the resolution of screens induced the programmers to create drop-down menus that were invisible unless pulled. As a consequence, sometimes the user felt in this situation of "not knowing what to do". Note that this is something that is appearing again, especially in mobile devices applications. Again, the reason is the same, the lack of space. Instead of solving the problem smartly, such as with a small semi-transparent button that affords to click but, at the same time, allows to see through, some programmers make the access to these features difficult such a certain key click or gesture is necessary.

2.2.5 Too small click targets

Click targets may be big enough for users to click properly and not be missed out. Sometimes buttons or checkboxes are clearly perceived, but they are too small or do not appear in the visible area and therefore are difficult to reach.

Like in the previous case, these sorts of unreachable UI elements appear again and again in mobile devices. They usually come in two different flavours:

- User interfaces not properly designed for the screen resolution or size: Designers of applications for mobile devices may take into account the resolution and size of the display been used. Sometimes the application has been optimized for a certain device and does not work adequately on a different, smaller one.
- Web pages that contain UI elements that are displayed very small. With the new browsers present in smartphones, users do want to visit the desktop versions of the webpages instead of the versions designed for mobile devices, which are often poorly designed, or are updated infrequently. Webpages that contain buttons to be clicked, often present those too small because the mobile browser does not automatically resize them, or facilitates their use.

Of course, these elements are particularly problematic for elder people or for people with motor skill disabilities.

2.2.6 Lack of feedback

One of the most basic guidelines for improving a dialog's usability is to provide feedback. The user must always know the system's current state, what response has been given to their commands, and what is happening.

If you do not provide visual or audible feedback on what is happening, the user will guess, and maybe he will guess wrong.

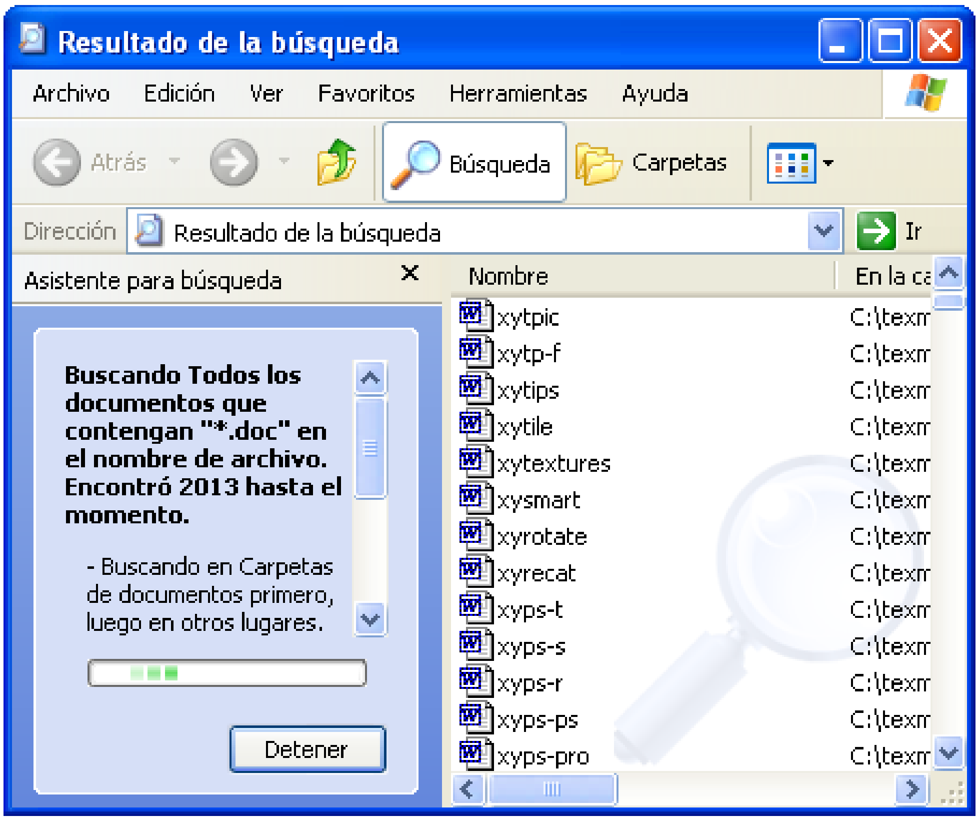
2.2.7 Lack of progress indicator

The lack of progress indicator is a variant of the previous one. If no progress indicator is provided, or the progress indicator does not work as expected (see Figure 25), then the user does not know if the system is effectively doing what he or she asked it to do. When a system fails to notify users that it's taking a long time to complete an action, users often think that the application is broken, or they start clicking on new actions.

If you can't meet the recommended response time limits, say so, and keep users informed about what's going on:

- If a command takes more than 1 second, show the "busy" cursor. This tells users to hold their horses and not click on anything else until the normal cursor returns.
- If a command takes more than 10 seconds, put up an explicit progress bar, preferably as a percent-done indicator (unless you truly can't predict how much work is left until the operation is done).

We will describe this issue later.



Buscando Todos los documentos que contengan "*.doc" en el nombre de archivo. Encontró 2013 hasta el momento.

- Buscando en Carpetas de documentos primero, luego en otros lugares.

Nombre	En la ca
xytpic	C:\texm
xytp-f	C:\texm
xytips	C:\texm
xytile	C:\texm
xytextures	C:\texm
xysmart	C:\texm
xyrotate	C:\texm
xyrecat	C:\texm
xyps-t	C:\texm
xyps-s	C:\texm
xyps-r	C:\texm
xyps-ps	C:\texm
xyps-pro	C:\texm

Figure 9: Searching window with a progress indicator that does not indicate progress.

The search window in Windows XP (see Figure 9), like the one in Windows 7, shows a progress indicator that does not provide any clue on the remaining time for the search to finish. In Mac OSX, there is only an animation that indicates that something is running, but, since the search can last several seconds, it should be replaced by a progress indicator.

Concerning the state of the application, you must never forget:

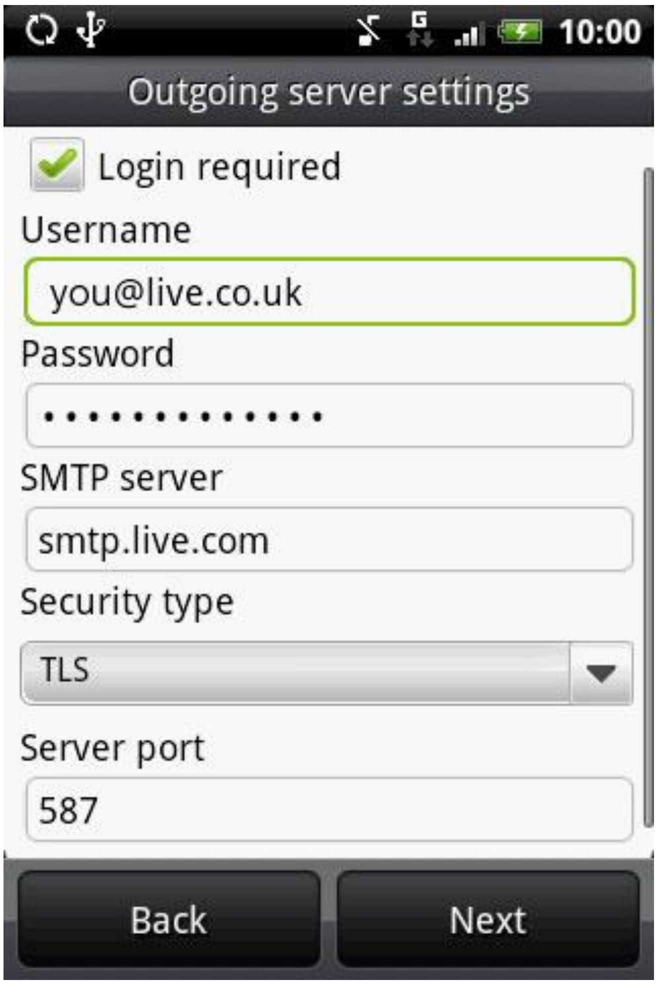
- Show users the system's current state.
- Tell users how their commands have been interpreted.
- Tell users what's happening.

2.2.8 Bad Error Messages

Error messages are a special form of feedback: they tell users that something has gone wrong. The most common guideline violation is when an error message simply says something is wrong, without explaining why and how the user can fix the problem. Such messages leave users stranded.

Informative error messages not only help users fix their current problems, they can also serve as a teachable moment. Typically, users won't invest time in reading and learning about features, but they will spend the time to understand an error situation if you explain it clearly, because they want to overcome the error.

An important error has to remain on screen for one or more seconds: If the message is too fast, and comes after an action that required several seconds, you might be looking somewhere else and miss the message. Take for instance the configuration process of mail accounts in a mobile device, shown in Figure 10: After you have entered the information on the account, it usually takes some seconds to verify the parameters of the mail servers and account name and password.


10:00

Outgoing server settings

✓ Login required

Username

you@live.co.uk

Password

••••••••••••••

SMTP server

smtp.live.com

Security type

TLS

Server port

587

Figure 10: Mail account configuration on an Android device.

If the process fails to verify the account, it will issue a message for a short time, and bring us back to the configuration view. If we distract a couple of seconds, we may miss the message and become confused on which was the reason why the process did not work (inability to reach the server, username/password error, bad server port/server security configuration…).

On the Web, there's a second common problem with error messages: people overlook them on most Web pages because they're buried in masses of junk. When we try to register for a service such as a blog, a shop or so on, we eventually have to fill a large form that does not fit in a single screen. Since the registration process may require some information as compulsory, and some other fields might be checked for correction, we may end up with one or more errors that may come in different flavours. Sometimes, we see a new red coloured label accompanying the wrong field. If we have multiple mistakes, we often change a single file and try again, and fail again. Having simpler pages may alleviate this problem, but providing a better error presentation (often discarded by the developers because it is more difficult to implement) may improve our experience.

2.2.9 Asking for the Same Info Twice

Most of us have suffered this on hot line services: typically, automated response machines from communications companies ask us several things such as the ID number, our telephone number, our name… And then, a human also ask us for the same data… And he eventually passes us with another person that asks us the same information!!!

This is a bothering practice that sometimes applications reproduce. Users shouldn't have to enter the same information more than once. If there is something computer excel at is at managing data. Therefore, the application should gather the information and make it available for the different parts that might require it.

Some booking applications (theatre or other events that last for several days) do recurrently commit this mistake. An example of booking flow that commits this mistake in a real application was:

1 Fill in the number of people
2 Select the day of the booking
3 Select the timing
4 Try to book seats

When the user was not able to book correlative seats for all the family, he wanted to cancel and select a different timing. However, the application turned back to first step making the process cumbersome and prone to mistakes. As a consequence, unless you found the right seats the first time, more and more time was consumed, and more and more frustration was generated.

2.2.10 Lack of default values

Default values may help users in various ways. They are especially important in form filling tasks. Several uses of default values are:

- Accelerate interaction: Several tasks may have acceptable default values, such as the length of tasks to be added to a calendar. Having default values in this case may free the user from specifying a value and therefore its task is achieved faster.
- Teaching how the formatting of a certain input is or a certain answer that can be appropriate for a question. Non-American users have had a date formatting problem many times. When entering dates in apps, having an example date may reduce wrong entries in a simple manner.
- Sometimes a default value may be perfectly safe or a common value, for instance in software installation processes. This may help novice users if they do not know about the details of the task. For example, port identifiers for different security protocols in mail servers are usually the same, and therefore, the configuration utility for mail may put them as default.

2.2.11 Fail to provide any usage notion on the application

Today, with the proliferation web-based applications and app stores, it is very common to encounter many new applications even without a deliberate search for an application to solve a problem. Application recommendations, for instance, that rely on groups of applications downloaded by users, that might by unlinked by theme or content is one of the sources of what could be called *arbitrary* app testing. As a consequence, users often approach the app without any conceptual model of how it works or which problem it solves. They don't know the expected outcome, and they don't know the basic concepts that they'll be manipulating. For well-known applications, such as word processors, this is less of a problem, but for new applications this may be a problem.

When you throw a user into an app and do not have any tutorial introduction or properly guide him through the steps he has to perform, the user will probably feel frustrated. For recommendation sites, such as Apple's App Store, this might turn into a bad evaluation that might throw the app into the back seat of low quality applications and never be able to go back. Note that most downloaded applications are in the top 25 or top 50 because the model of search of applications implemented in iPhones and iPads was built this way (i. e. easily see the Top 25 or Top 50 tiers but a nightmare to dig into the remaining apps). This is a problem for the newcomers: A couple of bad evaluations may sentence the app to lie into the almost unreachable jungle of mediocre applications.

You may see this from time to time when you read evaluations of apps you have used: You may find a user complaining about a missing feature you know it exists, though it is difficult to find because it appears in the custom Preferences of the operating system, instead of the application itself.

Applications should guide the users into the tasks by taking into account both the type of application and the type of user. When designing mission-critical applications may often assume that the users will have been trained and/or tried the app many times before. For other cases, we cannot assume the user will read the upfront instructions.

2.2.12 Not Indicating How Information Will Be Used

The information that an application asks to the user should be clear. The user needs to know how the application will use each bit of information. A classical mistake by inexperienced users arises when they are asked for a nickname in a blog or forum: Often, the user does not know that the nickname will be used for identifying them in their postings. As a result, many times users end up typing something inappropriate.

Other pages may require entering a postcode prior to showing any product in order to show the users either the shipping costs or the availability of a certain product in the shop closer to the user. If the application fails to communicate the reason, many people may enter a fake code or simply leave the page if they are worried about privacy concerns. A simple solution is to tell the reason of this question, or to delay the calculation of the shipping costs until a certain product has been selected, provided that the user is informed that the shipping costs may vary depending on the location.

2.2.13 Organising the data according to internal application design

Any application should not reflect the system's internal view of data rather than users' understanding of the problem space.

For instance a Research Tracking system that stores the information relative to the publications of the researchers will require an option to enter a new research track (i. e. a new published paper). Obviously, this will be the most often used option by a researcher. Probably, the options to generate the users' CVs will be second to those, and the rest of features will be of infrequent use. A normal user would expect from the system, after selecting a button for entering a new research track, to enter the title of the research paper. What the user found was a view **to search** by title the supposed research element. Obviously, this was an option added to avoid double copies of the elements, which can be entered by any of the authors. Clearly, this means moving the programmers' effort to detect duplicates to the user! This is wrong because it breaks some of the golden rules of design that state that the system should pardon the user if he errs, and should be able to recover from users' errors. Moreover, it incorrectly exposes the inability of the system to detect duplicates. A simpler way should be to allow the user to introduce the title of the work, and then check in background whether there is a similar title in the database, and then ask the user if it is the same.

2.2.14 Reset buttons on Web Forms

It is almost always wrong to have a Reset button on a Web form. Reset buttons clear all the fields, thus forgetting the entire user's input. Unless you are working on a system tailored to entering data, which almost never happens on a Web Form, this is an undesirable behaviour. It violates one of the most basic usability principles: to respect and protect the user's work at almost any cost. Therefore, destructive actions should have a confirmation dialog, the most destructive the most prominent.

However, designers should avoid falling into the opposite problem: too many confirmation dialogs. Asking the user to act too many times will lead to users that do not read the warnings, and therefore answering wrongly some times.

3 Typical UX problems

As we have commented elsewhere, designing and implementing an application is something complex, and user acceptance of the application may be a key issue in the future results. UX is often neglected during the design of an application, but it is of key importance for consumers, since it will determine its technical and emotional connection with the applications or web pages. Moreover, studies say that every dollar invested in UX yields a return between 2 and 100 dollars.

Therefore, it may become very important to be aware of the users' needs and expectations. Throughout the whole development process, many actions can be taken to make sure that the user experience does not suffer or that we detect UX problems early enough to have time to correct them.

Despite that, it is quite common to *forget* the user in early stages of the project, or to deal with UX as if it was a separate part of the project. Some of the common problems are listed below:

3.1 Leave the UX for too late

As we have said, a very common problem is leaving the UX research for the final stages of a project. Since usability testing uncover important problems that might require changes in the architecture of the application, letting the analysis for the latter stages (which may be commonly pressed by the schedule), may be a big mistake.

Significant changes in the architecture of the application or the overall design may be easily addressed when the application is in its early stages, but very difficult the more time has been invested into it. Several studies have shown that the maximum benefits are achieved when UX introduced early in the development process.

In many cases user feedback can be obtained with simple, unfinished versions of the applications. Even with prototypes, a lot of information can be gathered on the user interface design. Therefore, it is highly advisable to evaluate the design in its early stages, leaving more room for changes or improvements in case the results guide you to do so.

3.2 Improper distribution of UX efforts

Usability testing may discover many elements that can be improved in an application. Commonly, the result of a user study will be a list of problems with some level of priority that rate the importance of the problems. When dealing with problems, it is important to keep the focus. Since the resources are limited, it is always important to address the most important problems first, and the less important later.

In order to do so, it is useful to keep in mind the 80/20 principle: Not all the features will be accessed with the same frequency. Devoting more efforts to the 20% of the features that are more commonly used or are more important to our application may be a good way to invest efforts.

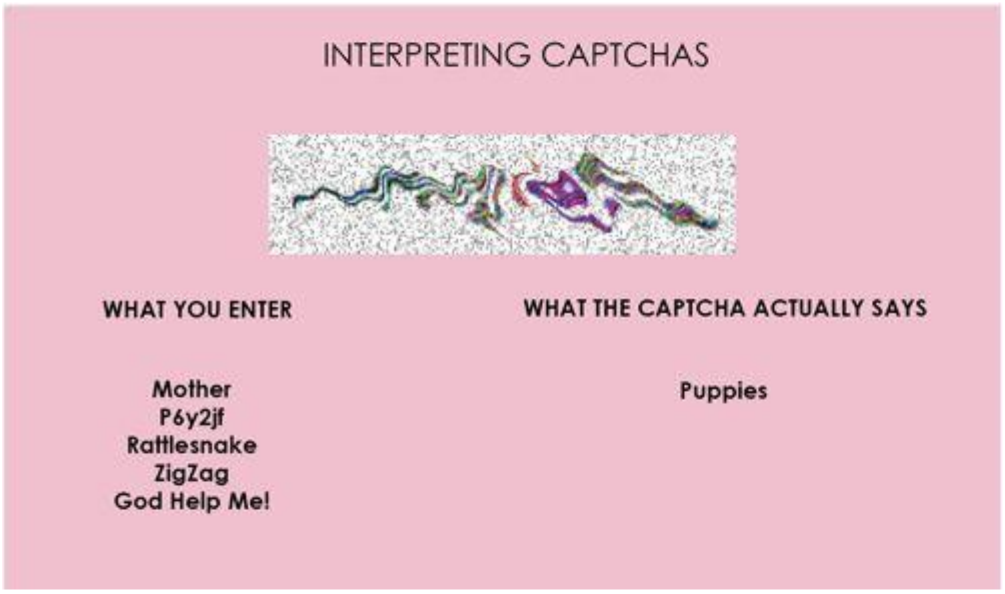
3.3 Lack of communication

In big companies/projects, the people working in UX may form a medium-sized team. Since its work may be quite different from the rest of engineers, this might result into an isolated group of people that seems to be working in a very different project than the others, with just a finite number of common points of interest.

Working together with developers and analysts, and spreading the information throughout the company/project members may make the work carried out less mysterious. Note that UX team are not there to *correct* or find the mistakes of anybody else in the project, but to help the other project members to make a better product for the user.

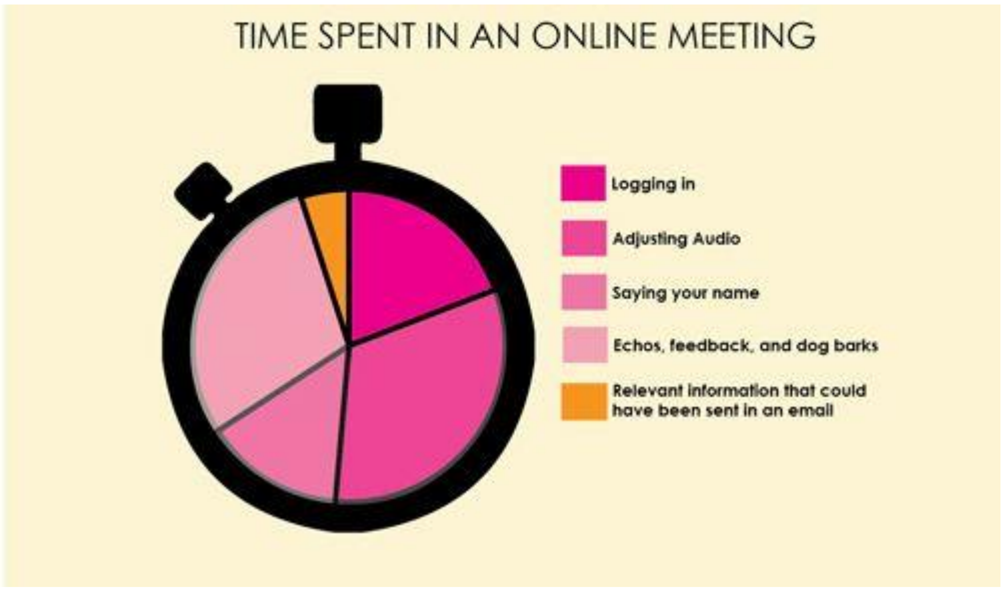
Thus, a good policy is to communicate frequently with other groups, both formally and informally, since this will facilitate collaboration.

3.4 Poor UX examples

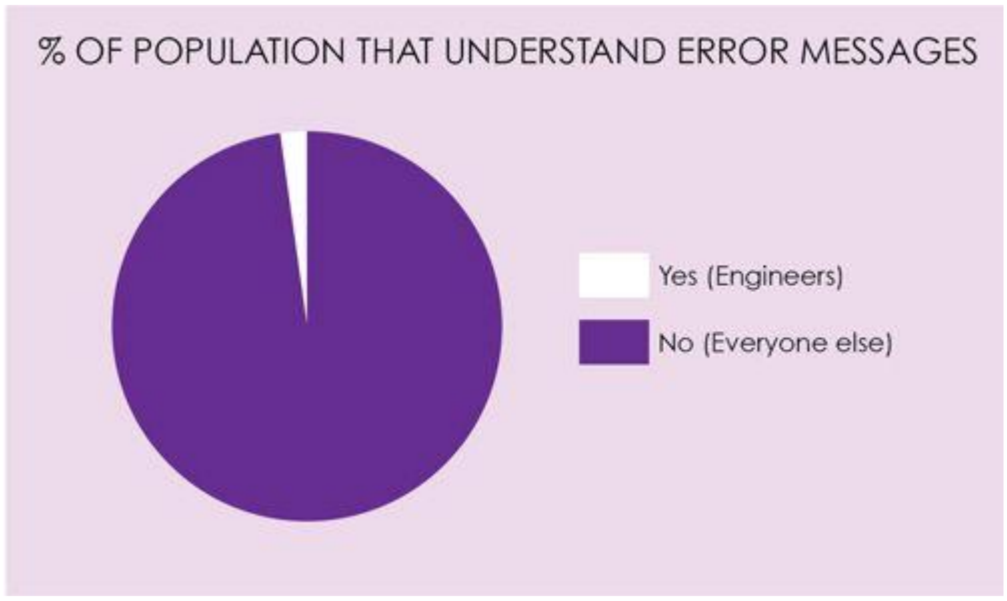


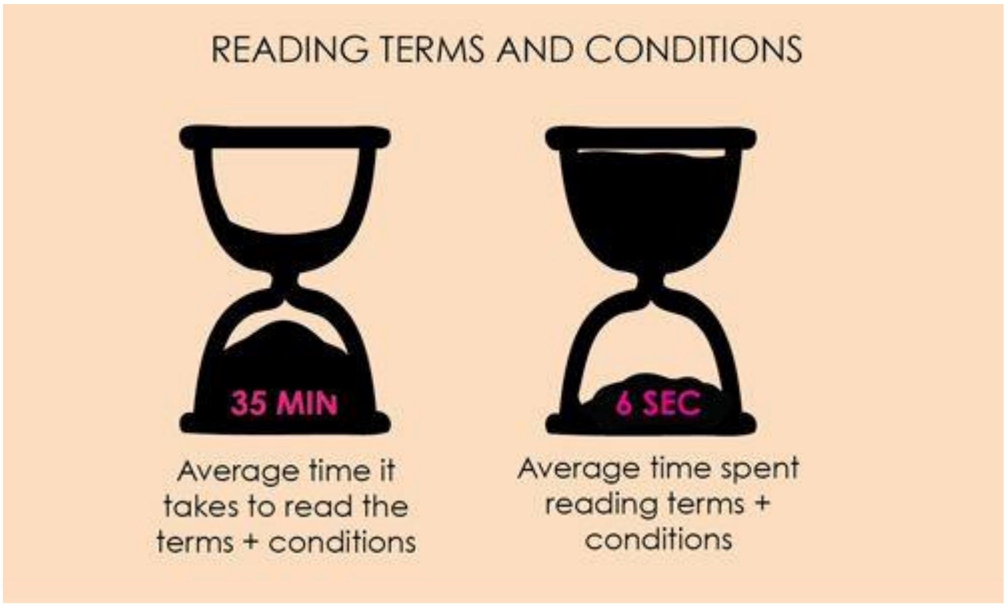
INTERPRETING CAPTCHAS

WHAT YOU ENTER	WHAT THE CAPTCHA ACTUALLY SAYS
Mother P6y2jf Rattlesnake ZigZag God Help Me!	Puppies



TIME SPENT IN AN ONLINE MEETING




READING TERMS AND CONDITIONS
35 MIN
Average time it takes to read the terms + conditions
6 SEC
Average time spent reading terms + conditions

4 Rating the severity of usability problems

The first step to solve the problems of an application consists of detecting them. Once detected, and as it has been commented previously, maybe not all the problems can be properly solved because the resources may be scarce. To make the efforts more useful, it is important to focus on the important problems to solve, and this requires an appropriate rating of their severity.

Not all the problems will have the same importance, and not everybody will probably agree on the exact degree of severity of a certain problem. Several researchers have developed some guides that may help you to properly rate the importance of problems.

Usability testing may discover many elements that can be improved in an application. There are two main parameters that are used to classify the severity of the problems: frequency and severity.

4.1 Frequency

Measuring the frequency of a problem is generally straightforward. We can count for instance the number of users that encounter a problem and divide it by the total number of users.

We can then classify the found problems according to the frequency (the more frequent the more severe). It can also be used to estimate the sample size needed to discover a certain percentage of the problems.

On the bad part, simply counting the problem frequency does not directly reflect the importance of the problem. A rare problem might completely destroy the database and therefore be very severe.

4.2 Severity

It is a less objective measure, as it will partially depend on the evaluator. The typical method is to define a simple set of categories and try to fit each problem in any of those. There have been several approaches to define those categories.

Jeff Rubin proposes a local evaluation of the problems. The different categories proposed are:

- **Level 4: Unusable:** The user is not able to or will not want to use a particular part of the product because of the way that the product has been designed and implemented.
- **Level 3: Severe:** The user will probably use or attempt to use the product here, but will be severely limited in his or her ability to do so.
- **Level 2: Moderate:** The user will be able to use the product in most cases, but will have to undertake some moderate effort in getting around the problem.
- **Level 1: Irritant:** The problem occurs only intermittently, can be circumvented easily, or is dependent on a standard that is outside the product's boundaries. This could also be a cosmetic problem.

Jakob Nielsen proposed the following four-step scale a few decades ago:

- **Level 0:** Not a usability problem at all
- **Level 1:** Cosmetic problem only: need not be fixed unless extra time is available on project
- **Level 2:** Minor usability problem: fixing this should be given low priority
- **Level 3:** Major usability problem: important to fix, so should be given high priority
- **Level 4:** Usability catastrophe: imperative to fix this before product can be released

Dumas and Redish use a similar categorization but thinking in global versus a local dimension of the problem. The four levels they propose are:

- **Level 1**: Prevents Task Completion
- **Level 2**: Creates significant delay and frustration
- **Level 3**: Problems have a minor effect on usability
- **Level 4**: Subtle and possible enhancements/suggestions

4.3 Conclusions

The general advice is not to use a large number of categories, since they will not give that much extra information and will on the other hand put problems to the evaluators to properly classify problems. It is also important not to get obsessed on the number of categories, 4 or 5 may do the work. Different evaluators may disagree on some problems' severity. It is also very important to deal with frequency separately from severity.

Moreover, since we are making a report on somebody's code, it is also important to point out positive findings.

5. Mobile usability problems

Smartphones have become a much extended device. Many people have one and they use it not only for phone calls, but also for entertainment, mail, and web browsing. With the term mobile device we refer mainly to smartphones (or feature phones), although some of the problems also apply to tablets.

Although the mobile phones have improved more and more, some authors believe that "mobile usability" is pretty much an oxymoron. In most cases, it is neither easy nor pleasant to use the Web on mobile devices.

There are some factors that have noticeably improved the last years:

- Screen size and resolution
- Web browsers in mobile devices

When attempting to perform a task using a mobile device, many problems are found. One of the most frustrating problems is the poor design of web sites, not adapted to mobile devices own characteristics, such as the size of the screen or the lack of Flash in some gadgets.

5.1 Problems using mobile devices

The most important features of mobile devices that may challenge our work are:

- **Screen size and resolution:** Although we may currently find mobiles with screens of up to 4.3 inches, and resolutions up to 640x960, we interact with a finger, which effectively reduces the size of the screen several times. Moreover, larger resolutions improve the quality of the text or pictures, but hardly influence the amount of information that must be presented at a time. The size of the screen makes the use of large fonts and sizes of widgets compulsory. As a consequence, less information is visible, and therefore the users must rely on their short-term memory to build an understanding of an online information space. Even high-resolution devices such as tablets require to zoom-in if we want to properly read a page of a newspaper.

- **Awkward input:** Besides of our familiarity with regular-sized keyboards and mouse, GUI widgets are difficult to operate without a mouse. Even if we can simply click buttons, text introduction without physical feedback using small keys, and with a keyboard that occupies part of the screen is more prone to errors and takes longer than with a physical keyboard. Physical keyboards, even if they are small such as BlackBerry's improve data creation a great deal. For larger devices it is still an issue. Large virtual keyboards cannot be compared to physical keyboards because we cannot have a comfortable typing posture (we cannot lie our fingers on the keys as we do in regular keyboards without typing).
- **Download delays:** Even with 3G connections if the webpage is not designed for a mobile device and thus has less text and smaller resolution images, loading pages requires some time.
- **Web sites designed for desktop:** With a large user base, many web sites did optimize their contents for displaying in an iPhone. Sometimes this works, but competing products (including iPads) are now featuring *full size web viewing* and being *full* a synonym of desktop quality. However, this induces some notable mistakes and misperceptions. First some devices include Flash, but others do not. Thus, some pages will not render properly on iPads or iPhones, and sometimes we are not able to know why the page did not load correctly. Second, widgets designed for a mouse access may be difficult to reach with a mobile device, and this would require a high level of zooming. And finally, even tablets have larger screen sizes, there are not big enough, and therefore require zooming-in for a proper reading experience.
- **Sites designed for mobile devices that show up in tablets:** Some websites have been designed for mobile devices, ideally, with small screens. Despite that, you may end up loading the smartphone version in a 10 inches tablet because the web detects the operating system rather than the screen size.

Even though connectivity problems will hopefully diminish in the future, but it will take many years until we have mobile connections as fast as even our office connection, without interruptions.

The studies prove that sites designed for mobile viewing improve the outcomes by the users and their satisfaction. Moreover, although some smartphones provide a mouse input-like element, large screen phones (touch phones) generate a better web site navigation experience, mainly due to the larger screen.

Devices with a full QWERTY keyboard improve over touch phones in tasks that require typing (curious enough?), such as e-mail, blog management, or Facebook.

5.2 The WAP paradox

Curiously enough, researchers at the Nielsen Group found that, although current phones are better in several aspects (larger screens, better processors, improved browsers…), the results of several tasks took more time than previous studies using sites designed for WAP.

They asked the users to perform two simple tasks:

- Find the local weather for tonight.
- Find what's on BBC TV 1 tonight at 8 pm.

These two simple tasks were also analysed several years ago when mobile phones were not so sophisticated than today. Paradoxically, the users attempting to perform those tasks several years ago required less time to solve them than today's users. They required an average of 38% time more with new phones.

The main reason is that when the first experiment was carried out, the mobile phones usually came with a set of applications from the carrier that provided access to a set of services such as the weather forecast. As a result, even if the phones were slower, the users were able to gather the required information in less time. Currently, the user has usually access to a broader amount of information and, without dedicated applications, he or she has to dig a little bit more to obtain the same results.

Today's approach is mainly search-based. This implies a lot of typing, with the typical errors easy to produce in a virtual keyboard. Thus, the timings increase. Previously, users were using a restricted number of applications, but with few typing, the results were fast available.

What really may make the difference are the applications. If you have the right app for the right moment, you may obtain the results faster, because the application will be designed for mobile use, and it only solves a certain problem. Current application marketplaces are the equivalent of a huge deck of WAP sites preloaded in your phone.

In any case, we are still very far from getting an acceptable user experience in mobile phones. There is still a great distance from what touch phones offer, to "full-featured" browsers on PCs.

When designing for mobile, we have to balance between making content and navigation salient so that people do not work too hard to get there, and designing for a small screen and for slow downloading speeds. That's why almost every design decision must be made in the context of the site being designed, and what works for one site may not work for another.