

## General usability issues

---

Usability comes from the word 'user' which simply put would mean 'the ease of **use**' for the end **user**, or in another words **ability of the user** to employ/interact with the given system/product at hand. See the obvious emphasis on the word user in the earlier sentence to understand how important s/he is to the end result of the product. The offering is as good as the user perceives it to be and not as good as the development/marketing team emphasise it to be. Only if the buyer of the product is able to take full advantage of the application without much hassle and with relative ease and comfort does a product pass the ultimate marketability and usability test.

With technology evolving everyday allowing for more complex systems to be created and the market space for every product being saturated quickly faster, usability becomes even more important. The user is neither bothered about the implementation complexities nor does s/he have the patience to learn how to use the product. They want to be able to use the product immediately and they also want it to respond to their requests immediately, a slow/complex system is often deemed as an unusable/unreliable solution. The essence of a user-friendly application is the efficiency of the system, ease of use from an end user's perspective and the final satisfaction that the user derives from using the system/product.

Here is a list of general usability issues to keep in mind.

- **Use the right color combination and be careful with your choices**

Colors have been an integral part of our lives and over the years we have formed strong opinions about different colors where each of them immediately evokes a certain kind of response/feeling.

Some common colors and their connotations are as follows:-

1. **Black** has been regarded as a powerful color with connotations of power, authority and elegance, which explains the reason why black remains a favorite with fashionistas. It also stands for mystery and empty ness along with grief and loss at times, which explains the use of the black veil/tuxedo for mourning etc. Be very particular with the use of this color in your UI as you certainly don't want the user to take away the negative connotations associated with it. Some place where the use of black could be perfect is games where you want to depict the unknown like a map/road/course of action etc.
2. **White** has always been associated with purity and freshness. It also stands for freshness of ideas or a new beginning, hence the reason for it being used as the main color for most search engines like google etc.
3. **Red** is a stimulating color which is generally used to draw the user's immediate attention, hence should be used with extreme care because you don't want to alarm the user too often. Red is also perceived as a violent/angry color at times which explains why most websites don't seem to use red that often unless the intent is to depict something gross/violent/gory.
4. **Blue** is a soothing natural color, one which symbolizes tranquility and peace, this could probably explain the reason why most Nokia phones comes with a default theme that has shades of blue. One reason for blue to be considered a soothing color is that we see that color in most abundance around us with both the sky and ocean being blue in color.
5. **Green** is another of nature's color as they say because the plants and ecosystem around us is green, and it is associated with freshness. Green is also associated with power and masculinity at times which could explain the reason of it being used in lot of gaming applications which emphasise on strategy/war etc.
6. **Yellow** as a color attracts the attention of the user almost instantly because of its bright hues, but at the same time it could be intimidating and sometimes irritating coz it is not that easy on the eyes as a color. One should be careful with using the right shades of yellow, a lime yellow could be perceived as fresh and refreshing whereas a bright canary could be very jarring on the eyes.

Most of the colors have their own connotations and meanings given the cultural backdrop and hence you should do a careful analysis of the intended market before using a particular color.

- **Study your target users**

This has to be easily the most important component which drives the usability design process. Without understanding the target users no user friendly product can be developed. The skill sets/capabilities/expectations/usage pattern of the target audiences goes a long way in designing a user interface which is both efficient and easy to use. The first two would help you in understanding how much input/information you should seek from the user in your application and how much of it should be programmed to avoid inconvenience to the user. For instance lot of times the user might not be conversant with the complexities of the internal implementation details like VPN gateway settings, enable/disable of settings etc. While the latter two would help in understanding how to implement the outputs of a particular request and in also writing the help manual/documents.

A nice way to capture the target user's expectations would be to have one on one with some prospective users, people not in the know how of the development/design process. Sometimes a survey/questionnaire mode to gather datapoints could also be a good idea.

- **Focus the user tests on the most relevant product versions/features**

While writing usability test cases, you should always consider the product offerings and the features that are going to go out in the current release version. More emphasis should be placed on testing the features that you perceive the user to be using the most, as those are the bread and butter of the application so to speak. Also the user would want at least those basic features to be easy and reliable to use. For instance in a dialer application the user would at best expect to dial/receive/hangup calls, even if at times it doesn't let you edit/delete a number from the dialpad etc. So while writing the test cases you should drill down those functionalities more with more usual/alternate/exceptional flow cases.

- **Don't be blinded by your own expertise.**

Ground rules to make your application usable, do not be fooled/blinded by your own expertise in the development/implementation phase. The end user does/need not know the internal details of the product, all they care for is the ease with which they can use the features developed by you. If the user interface is cumbersome and complicated to understand likely chances are that the user won't use it too often. As they always say "**first impression is the last impression**" if the user sees a cluttered home view, with complicated navigation/menu layout etc they would be very irritated with the product and more often than not they won't even bother to check the durability and robustness of the solution.

Another golden adage that holds true is "**looks can be deceptive**" and sometimes a product with good look and feel ends up attracting more sales and user satisfaction as compared to another application with lousy UI but strong features. So always ensure that you design keeping in mind the end user sensibilities and requirements. While testing the user experience don't use your friends for testing, because their opinions/feedback could be colored as they might not want to offend you by passing negative comments about your own. Always ask the end user's input, they are the best judge of your application's layout and usability.

- **Know the users and design for them**

Always design for the users and not for yourself. The user is the one who is going to spend his precious dollars in buying your product and they always want the best value for money. How often have you come across a product with bad interface selling well, a well packaged product sells better than a lousy one.

- **Take users into account in every phase of the application development process**

Users are the single most important component of the entire value system from your point of view. If the product doesn't sell and the company doesn't make money they wouldn't be in business which in turn could mean not so good news for you. So always involve and keep in mind the sensibilities and expectations of the user. It would be better if not best to involve an end user in every phase of the development process as they would give you constructive comments which could prove beneficial in designing an interface that is usable from their stand point.

- **Make porting easier, don't hardcode the user interface**

While designing/architecting the solution keep in mind the various form factors/specifications and other visual parameters of the devices you are targeting the solution for. This would help in ensuring that while porting from one device family to the other the efforts are minimal. Use a disjointed architecture like MVC which separates the business logic from the UI as much as possible so that the back end code can remain the same while porting and only the UI needs to be customized. It is never a good idea to hardcode the co-ordinates/location etc which drawing/creating the UI as the same co-ordinates won't fit in for all devices.

- **When porting an application, do it for the targeted users**

- **Porting an application successfully means porting the user experience too**

Porting is nothing but customizing the solution so that it works on the targeted platform/device/language etc. The end user does not know what/how you have created the solution i.e. whether you had first written it for S60 and now you have just changed it for S40 or whatever, for them it's a new solution for which they have paid money and they want it to work reliably and have a relative ease of use. So don't think that since you are just porting a solution you can not think about the end user experience, the basics still remain the same even in this case.

- **Choose the right navigation**

From the main view only the core functionality of the application should be available. The navigation model should put more focus

on the main tasks hiding the complex and advanced functionality from the beginners.

- **Ease of input for the user**

Make sure that providing input is easy for the users. Application should not be expecting large input from the users, instead of text prefer the other forms of entry like selecting from a list. Also minimize the number of clicks, use default values while taking inputs etc.

- **Help please**

At any point of time users should not have a confusion on *how to proceed, where to click now*. Provide the context sensitive help, and a user guide.

---- Edited by Mayank on 11/06/2009 ----