



Applications/products/solution are becoming more advanced and complex in nature with rapid advances made in the hardware capabilities of the device and the ensuing software improvements to support the increased processing prowess at hand. With the ever increasing complexity of the solution comes the perennial problem of either educating the end user the nuances of being able to use the system or ensure that despite the complexities behind the interface it remains still as easy and convenient to use the product. The former is time consuming/costly and sometimes impossible given the market dynamics and the geographical reach of a product. Also higher the time taken to learn how to use the product lowers the chances of a satisfied user.

Time is of essence in today's world and more often than not the user doesn't have the time or the patience to go through a training/manual to understand what's and how's of the system before being able to use it. They rather things be upstart and ready to use if as little effort and time taken as possible.

This brings us to the second aspect of a software development, ensuring a higher level of usability at every step of the development life cycle. **Wikipedia** describes usability as:

The primary notion of usability is that an object designed with a generalized users' psychology and physiology in mind is, for example:

- ◆ More efficient to use?it takes less time to accomplish a particular task
- ◆ Easier to learn?operation can be learned by observing the object
- ◆ More satisfying to use

For more on that check [this link](#).

Usability becomes all the more important when designing for the real estate constraints of the mobile device. A user interface which might look very nice and chic on a desktop might look very cluttered and ugly on the mobile device. Even before getting down to the development of the product, a careful analysis needs to be made about the overall look and feel of the interfaces and how to compress the content to the device specifications without missing out on any key components.

Key components of usability design are:

* **Understanding/respecting the user**

Since the idea of usability is to ensure a high user satisfaction which in turn is driven by the ease of use and efficiency of the product being offered to the customer, it is imperative to understand who the intended user of the solution is likely to be. The comfort level, usage patterns, technical skill sets, expectations/requirements, cultural/linguistic identity mapping of the target user should ideally be the first step of usability designs. Some ways to achieve this could be 'one to one interviews?', 'questionnaires and surveys?', 'observation of the user?', each of which would throw up data points and information which would be helpful in usability design. Conducting face to face meetings with users to understand what they are looking at from the intended product, what are their usage patterns for a similar competitor product? Questionnaires and surveys would give insights into what the key components of the product should be from a user's desirability aspect, what features/interfaces works well for them and what doesn't. Sometimes spending time with the end user to just observe their usage patterns, behavioral patterns gives us an insight into what the resultant product should look/work like.

A nice idea would be to have a diverse set of potential user group as this would throw up ideas/insights with different perspectives. This would help in bringing in new/fresher ideas, as a closed user group interaction

Designing_usability

sometimes can't think outside the box as they say. It might also help sharing with the end users, some of the ideas you might already have about the usability of the system. This would help getting a buy-in from them in case they like it, or also assimilate improvement ideas if there are any.

It is imperative to never under-estimate the intelligence of the user, as they generally know what they want even if they don't know/need to know how the same can be achieved. The latter is anyways not their job; it's your area of expertise. During your interactions with the user never try to intimidate/overwhelm them with technical jargons and complexities. Always ensure that you bring forth easy to understand and follow discussion points and outlays, try to present it in easy to relate and follow manner, as the primary idea is to have them at comfort for them to share their ideas with you.

Another important aspect of the user interfacing and interaction is to get a final buy-in/sign off of sorts from them. During your interactions capture all their ideas, collate them into some sort of a template which can be used as a reference point and once you have done that get it signed off by the group you are interacting with. This would help you in your usability testing/validation once you are ready with a blueprint of the product or a usable/demo able version, depending upon how you are approaching the entire usability engineering.

The users communicate with each other! That shouldn't be a surprise for anyone. And if good news travel fast, the bad news travel even faster. It is very important to remember the basic fact that you won't be in business if not for the user who spends their precious money to buy the product/services that you have to offer. Always ensure that the adage "consumer is king" is respected, just having a killer application with jazzy technical stuff and not being all that usable won't do the job. The end user has to be satisfied and happy enough to recommend your product to others, always strive for positive word of mouth publicity and a product which scores high on usability generally does end up being talked about in good reviews.

*** Testing and validating the system**

Once you have captured the user requirements and done a careful analysis of the user psyche and usage patterns, you get down to the actual implementation of the same into the product. You should ensure you have a budget/buffer for testing and validating your system at every step of the design with the end user so that you don't end up with a half baked/not usable system at the end of the life cycle.

An iterative approach so to speak would allow you to quickly remedy any glitches/mistakes early on with lesser costs and time impact on the overall deliverables. For instance if you were to follow the waterfall model of development this kind of validation would become very difficult as you would have a structured approach/timelines to the overall development. Consider carefully the impact of this on the final outcome of the process before even getting to the individual phases per se.

*** Design to make the constraints your strength**

Even though this may sound a little philosophical but if actually put into practice it can turn out to be that ways. Most mobile devices have a limited screen space, but are rich in color support, UI controls, layout and other tools which can be used effectively to create an enriching user experience.

Some key points to remember while doing usability design for mobile applications:-

- Ensure that the application follows a neatly organized navigational layout; the idea is not to confuse the user but rather to help them at every step.
- Maintain consistency of fonts/colors and text sizes on all views to ensure an easy on the eyes user experience. Localize the content where ever possible.

Designing_usability

- Provide a help manual/online help/context sensitive help to guide the user with the workings of the system
- Always follow the Style Guides while implementing the options/commands layout, you don't want to place a negative action on a soft key associated with a positive action and vice versa.

A thorough usability design and implementing it properly during the coding/testing phase would ensure that you have an application which is both ease to use and is well appreciated by the customer.

For more on usability see:

- Tim Fidgeon's article: [Design vs Usability](#).
- John D. Gould and Clayton Lewis's article: [Design for Usability: Key principles and what designers think](#).

---- Edited by Mayank on 12/06/2009 ----