The Usability Expert's Handbook

Table of Contents

Chapter 1:
What You Should Expect? *3-4*

Chapter 2:
6 Crucial Usability Standards *5-6*

Chapter 3:
5 Challenges of Application Usability *7-10*

Chapter 4:
11 Best Kept Secrets of Designing Usability *11-12*

Chapter 5:
5 Top Testing Methods for UX Professionals *13-16*

Chapter 6:
6 Guidelines to Improve Your Usability Testing Methods *17-20*

Chapter 1:
What You Should Expect?

Being a usability expert is not as easy as it seems. A lot of time and effort goes into making things appear effortless. This may sound like a contradiction, but it's really not. The internet is a great example of how people struggle to make things as simple as possible. Usability experts aren't born overnight; they have plenty of useful resources that help them understand what is going on and push through the tough times. So, what exactly does a usability expert have in their handbook?

Let's dive in and take a look.

Before we jump straight into the deep end, let's take a while to talk about usability and what it actually entails. The basic definition for usability is simple, it's a term used to gauge how simple something is for a person to use. This could be a tool, software program, or a webpage.

The easiest way to look at usability is to think about if a person walks right off of the street and is given an item or taken to a site, how quickly will they be able to master it? A usability expert is a person who specializes in making things seem simple so that users can adopt it very quickly.

First, let's very quickly buckle down and think really why usability is so crucial to the success of an online business. Think about a person that has two online retailers that they can make a purchase from. The cost of the item is exactly the same, and there is no difference besides how the site is laid out.

Usability isn't specifically about taking out things or dumbing them down, though it can be. Sometimes, it's about taking what you already have and making it easier to digest for the user. People have studied usability for years, and they have come up with a set of rules and guidelines that they follow to ensure the highest success rate for any item. By taking a look into their handbook, you'll be able to understand the hidden secrets that experts have been using for years. Whether you are a seasoned veteran looking to mix things up a bit or are just starting out and need some helpful guidelines, it's good to know what other people in the industry are doing.

There are some "must haves" for anyone that wants to be an usability expert, and the first thing is literally a handbook. It sounds silly to say, but you'd be surprised at the number of people who don't write down useful information in a little notebook or save it in a folder. There is plenty of great information out there, and many great ideas will be lost if you don't keep track of them.

As for everything else?
Well, that's what this handy guide is for.

Chapter 2:
6 Crucial Usability Standards

We just covered how usability refers to web applications that have been modified in a way to be easy and flexible to the end web users. Without any skills or training, users will be able to intuitively associate anything they need to do in the web page with any other interactions as well as performance they see relevant.

But, are there any standards pertaining to the web usability? Let’s take a look.

Vital Web Usability Standards

1. Application

This is one of the most important web usability standards that applies to the departments under the financial administration act that unless they are disregarded by the orders in the council, specified acts or regulations, are accountable for facing the public, web achieve, web content and pages.

2. The Context

The second important web usability standards states that the format and structure of any basic site should make it easier for any users to access and obtain information in the most efficient and convenient way. The Federal Integrity Programme has prescribed a method that identifiers and visual elements can be applied to the design.

An appropriate degree of consistency should be obtained from the method that presents a favorite way to departments to adapt to the changing technologies as well as develop their own for their targeted clients in the most fulfilling ways. For the purposes of this standard, the Policy of Management of Information Technology is supported.

3. Effective Date

Also featured in the most important usability standards is the date within which the usability is to be affected. It is expected to take effect as from 28th of September 2011 and it came to replace the standard on web addresses as well as the standard on the formats of common web pages.

4. Standard Statement

The sole reason for this vital standard is to ensure the Canadian Government websites achieve a high web usability level. The expected results were that the Government websites respects all approaches as well as usability principles. The standard also continued to explain that web content owners, specialists and managers were responsible for ensuring that websites met a number of requirements of web design, addresses and notices as described in the appendices C,D and E retrospectively.

5. Government Wide

The government wide is also included in one of the most important web usability standards which states that the Government secretariat of the treasury board will monitor the usage and application of the standard in a number of ways but not limited to work performance in collaboration to departments, assessment under the MAF (Management of Accountability Framework) and examination of the departmental performance reports.

6. Responsibilities of Reporting and Monitoring

Senior departmental officials are responsible for accessing their deputy heads to make sure they implement on this standard and bringing to their attention any form of difficulties, compliance issues as well as gaps in the performance frame works. As for the deputy heads, they are responsible for the adherence monitor of the standard in their prospective departments and in consistency to the presentation from the treasury policy on evaluation board.

Although the standards are really clear, many companies and consultants have stumbled across some challenges along the way when designing for usability.

Chapter 3:
5 Challenges of Application Usability

Technology is becoming increasingly integral in our daily lives, almost as much as the air we breathe and the house which keeps us warm at night. This is going to continue to be a long-standing trend, barring some unforeseen calamities, and the wheels of progress will not stop turning during that time, either. This means that software will forever evolve and change, as will ideologies behind how it is designed and how we should interacted with it. This makes application usability a constant challenge.

As new designs and trends come and go, new systems and devices rise into novelty and then mediocrity, standards and expectations for how software should look, think and act change constantly, almost in rebellious reaction to standards being founded. As a result, it can make the lives of software designers and software companies rather unpleasant, as users berate and judge them for failed experiments or failing to keep with trends that never hold still.

In the spirit of this, let's take a look at the top five challenges of application usability and why we should all give the programmers a little slack. As users, perhaps if we have an understanding of this, we can also learn to provide proper feedback as to not confuse those who try to forecast our needs into creating another 3DO or another Windows 8. These were our faults, by the way.

#1 - Platform Disparity

It may seem like the Cloud and smart web design are changing how platforms affect software design, but in truth, they're nowhere near negating this issue, and are at least a century from doing so. Platforms are a problem, and no two, be they operating system or physical device, are truly the same. Contrary to what many believe, one Android smart device is not identical to all the others, architecturally, and the disparity grows when it moves to whole other systems such as Windows Phone or iOS.

This increases more when you factor in different devices, all of which have different mindsets and interface themes. PCs require a different approach from smart phones, which require a different approach from tablets, which require a different approach from smart gaming consoles and so on. This means that unless unique redesigns of applications are made for each platform, a unified design and concept has to be created that works moderately well for all at once. This is a compromise, and the definition of a compromise is an agreement where nobody is 100% happy, yet, software must be approached in this manner for the time being.

#2 - Aesthetics

Aesthetics are a problem for application usability, because as a consumer base, we like things to be pretty and shiny. Alas, some interfaces just are not conducive to this, due to screen real estate, processing power or simple practicality.

Overly-aesthetic designs hinder usability, make the programs harder to understand, and result in a bad, overly-engineered design that, while really attractive, slows the machine down or simply isn't workable on a practical level. Again, see Windows 8 as an example of aesthetics getting in the way of functionality. Users did not like the look of Vista or 7, and as a result, an obsession with aesthetics has resulted in the utterly useless Windows 8.

Many mobile applications are beginning to suffer from this as well, and it's worse here with screen real estate already at a premium. Some compromise must be accepted by users where a clean, elegantly simple look with functionality is acceptable, over a gaudy, impossible to use piece of software.

#3 - Productivity

With limitations in screen real estate even on PCs and large displays, there is the challenge of making productivity possible, and making features and tools within the application easy to find and use. The drive for this has created some trends in software that customers love to complain about, but have nonetheless resulted in increased productivity.

Two famous examples of this are the hatred of the ribbon interface that most Microsoft applications utilize now, which while ugly and awkward, does work, and the docking system used by most mobile systems. These make it easier to use the program, but users often balk at them, citing how they inconvenience them, or how they just liked old methods better. Once this mentality is overcome, only then will application usability be properly rated.

#4 - Complexity

This plagues business and artistic software more than anywhere else, but it can be seen in mobile applications too. The problem is that some software is immensely complex and powerful and simplifying the interface just isn't possible for them. This causes the software to be very confusing and intimidating at first glance. Often, the software isn't difficult to use once the user has time to learn the order of approach, but many users don't give this a chance, resulting in them running away and complaining that the software's just impossible. Obviously, this is a misjudgment of application usability. Softwares are starting to integrate additional tools in order to alleviate complexity. One tool that is popular for promoting self-service and ease of use on the interface is WalkMe. WalkMe adds unobtrusive step-by-step guidance on the interface to guide the user through any complex task.

#5 - Feature Centricity

This is partially the fault of the design industry, but as users, we should share the guilt on this. Many programs and applications become bogged down in additional features that can drown out the core functionality the software was intended to serve. This is because of the demand for integration of these features, primarily the case with the demand for social networking functionality being built into everything. Once, everything had a custom browser, now it is social functionality. It's our fault as users for supporting this, but also the fault of programmers for not knowing when to actually disregard what users say. A prime example of this is Skype and

Windows Live Messenger, which are becoming less like communication tools and more like social network desktop and mobile apps as the years go by.

These are just a few reasons application usability is a difficult thing to achieve properly for programmers this day and age, and as users, we must remember these things. Perhaps the next time a change or innovation is made in our software, we should think about these before complaining too vociferously, or before berating the updates too strongly.

When approaching usability, key into the best kept secrets to keep the application or website user experience to par. Next, let’s take a look at these secrets that will keep you ahead of the game.

Chapter 4:
11 Best Kept Secrets of Designing for Usability

Designing for usability in the software and web services industry is an immense challenge. Users often don't realize just what a challenge this provides and they tend to assume that it is in fact less of a difficulty with the diversity and complexity of modern technology, versus the limitations imposed by older technology.

This could not be further from the truth in fact. The problem is, users have so long held this belief that many designers are beginning to pick up this misconception as well, even Microsoft's been guilty of it lately as well as Apple.

Here is a look at the top 11 secrets to designing for usability. Keep in mind that these aren't end-all solutions, as technology changes so quickly, and the public view of how things should work sways with the wind. But, these are generally the best precepts to bear in mind when designing software or interfaces today.

#1 - Using Real Estate Wisely

Using screen real estate wisely is probably one of the most important things that designers don't do in modern times. Wasting screen real estate with large, mostly empty windows will infuriate users, and it greatly breaks the aesthetic of the interface as well. Avoid making a menu fill a screen when it does not need to. This goes the same for not using enough, and scrunching things into small fields where they require too much scrolling. Speaking of scrolling …

#2 - Using Scrolling Wisely

Understand that scrolling isn't something users just take for granted as something that must exist. Infinite scrolling only works in certain types of interface, primarily ones with no looped navigation.
Also understand that nobody likes horizontal scrolling, so designing an interface to avoid this as much as possible is always an important goal to have.

#3 - Menu Centrality

Unless designing for a tablet, and even then to some extent, it's best to avoid the page or "stacked card" interface layout. A common example of this layout is Windows 8, which is unanimously hated by pretty much everyone.
This layout concept is awful and makes the user feel as if they can never see all of their information at a given time. Simply avoid this, really even for tablets if possible.

#4 - Color Schemes

Using color schemes wisely, and picking a base set of about five is always important. Too many colors, or clashing or too-similar colors can result in the interface being hard to read and hard to look at.

#5 - Shortcut Keys

When designing with PCs in mind especially, shortcut keys are important to not only make available, but to make easy to use. After Windows XP, windows shortcut keys became kind of awful, and fast one-handed typing of them became impossible to do. Never underestimate a power user's ability to not only learn but develop an affinity for shortcuts like these.

#6 - Avoid Touch Dependence

Touch technology is prevalent on handheld devices like mobiles and tablets, and for obvious reasons, but designing for usability means not overemphasizing it too much. Requiring elaborate multi-touch actions for tasks will frustrate users in the long run. It is best to treat it like a tap interface, not unlike the point and click of PC interfaces.

#7 - Scalability

Designing an interface to smartly scale to make the best use of screen rations and/or window states is important as well. In a PC environment, users want to be able to maximize windows, or set them as free-floating windows as well. On mobile, users will need it to suit whatever device they have, so that it never looks awkward and designed for just one resolution or dimensional layout.

#8 - Allow Custom Settings

Custom settings can be simple as allowing users to choose fonts and colors, as well as turn off features or components they do not use. This goes a long way to allowing a user to optimize their program, and not feel encumbered. Remember, not everyone's eyes like every color scheme, and not everyone needs every feature software may provide.

#9 - Avoid Excess Docks

Docks are window components that lock into stacked series on the top, left, right or bottom of an application. These are unavoidable in mobile design, and can be useful in PC designs, but it is best to avoid abusing them.

If there is a work space, it is best to not have docks on all four sides of it, nor is it ever a good idea to have two sets of docks anywhere. Adobe is very guilty of this practice.

#10 - Wording and Size Consistency

When there are many windows, pages or menus in an interface, the wording scheme should remain consistent across them, so that if one button says "ok", they all should, if they serve the same local purpose, rather than another saying "okay" and another saying "accept". At the same time, all buttons of similar purpose should retain the same size and rules in code.

#11 - Avoid Tips of the Day

Nobody likes tip of the day notices, which are more prevalent in PC design but are showing up on mobile too. People just regard them as a pop up that they have to close, as they will seek the contained information when they need it, not when the program deigns to show it. Avoid these, but if they must be there, allow users to always disable it if they desire to.

These are the 11 biggest things that designers need to bear in mind when designing for usability in modern software and services, but the list could go on forever if permissible.

Chapter 5:
5 Top User Testing Methods for UX Professionals

Now that you've gotten the secrets down to designing for usability, it is so crucial for the success of your application or website to test for usability. But, have you been wondering what type of user testing technique works best for you? Well, you are not the only one. And thank goodness for LinkedIn, I had the great opportunity to get in touch with a few UX influencers to get a sense of their favorite user testing methods. I asked them what type of user testing method they found most efficient. The discussion did not only yield some intriguing results, but offered a pretty diverse list. I found it in great interest to present the discussion to you to get a sense of what user testing methods would be the best fit for your next project.

Let's take a look at some of the user testing methods that were introduced by my new friends. But first, I want you to meet them – the people whose expertise brought light to this topic.
Please meet…
Thai Dang, Senior User Experience Architext at KIT Digital, Inc
Amy Tandon, Product Manager at RockeTalk
Karin Dames, Test Coordinator at Mobistar
Ian Franklin, Business Psychologist and User Experience Consultant
Genie McDonald, Senior Interaction Designer at Travelport
Behold, the most efficient user testing methods:

1. Focus Groups

When I asked Genie what user testing method she found most effective, she told me that she keeps finding herself wondering what "user testing" truly means. Genie believes that the most effective methods are those that allow you watch and interview real users while they interact with a product or service. Working with focus groups allows you to do just that – focus on a user and their skills. Focus groups allow you to deeply and extensively see a group of users interacting with each other to discuss an idea or concept. This can spark some great inspiration as you will gain more insight from the ideas of the group.

2. Tree Testing

Tree testing is an effective method that provides a reality check for the user experience designer and your business. It allows you to see how well users interact and find items or elements in the website hierarchy. This helps you to understand what points in the hierarchy need work based on where the user stumbled. Thai told me that this testing method allows a partial reality check. This approach reveals whether your information architecture structure is easily understandable. A basic paper-print approach can work for this, but now there are softwares available for tree testing as well.

3. Remote User Testing

Remote user testing allows you to conduct testing from the comfort of your personal space, by computer or telephone. It is much easier testing with this method. Ian shared his thoughts with me, stating that a large consensus feels that remote testing is the most efficient because it can be implemented throughout the whole process of development – from concept to post deployment. “All you need is the user’s web cam to be pointed at the device of application and a web-based tool to record the session,” he says.

A nice aspect of this method is that it is budget friendly. Thai mentioned to me and the rest of the group that remote testing can be carried out even if your budget is minimal. The downside of this method is that the mental model of the user – the user’s thought process when using a product or service – might not come through as much as when using in-person methods.

4. Beta-Testing

During my discussion with the group, Karin pointed out the value of beta-testing. It allows you to roll out a product to individuals who are keen on providing an objective feedback, thus creating a win-win situation. This is because you will not only receive valuable feedback for the product, but you will also be able to effectively market your products before they are shipped.

It is obviously assumed that sufficient in-house testing was carried out to test the product functionality, before releasing this product to the customers. Naturally, you do not want your customers to locate bugs, you simply want their feedback on the product usability, product feature completeness, etc.

Beta testers may not like the idea of any major placement alterations. This is because they are accustomed to having things in specific places. This leads us to the last testing method.

5. User Diaries

Amy believes that It is necessary to involve real users when testing. No amount of tools or test cases can substitute for real life testing with real users. These end-users must not be near the product during the production stage. It might be a good idea to hand it over to some beta testers from an existing user base as well as include a few ‘new’ and potential users for an extended use of 4-5 days. This will get users in a space where they can offer real feedback if provided with beta testing. This will happen in their own course of time, whenever it is convenient for them and needs to be unsupervised. Ask them to simply play around and explore the product.

Remember, for people who pretend to act like real users, their judgement may appear clouded. They have been in close proximity to the product as well as the test cases. Hence, they are unable to provide feedback like real users. Ultimately, they are likely to fail to catch the key usability issues.

Chapter 6:
6 Guidelines to Improve Your Usability Testing Technique

Once you’ve adopted usability testing to your process, you will find that the product will change at time due to the results you will receive. If you are going to add new features, or need to understand your users more deeply, make sure you stay on top of the different usability testing techniques that will provide great results. Your goal should be to discover usability problems, collect quantitative data (e.g. time on task, error rates), and determine the participant's satisfaction with the product.

I've gathered 12 tips to sharpen your Usability testing technique, which is key to discover more errors and areas of improvement in your product.

1. Setting clear criteria for participant recruitment

Recruiting the right participants is key for effective user research, because your research results are only as good as the participants involved.

You should deny participants who have conflicts of interest (working for your client or competitor), who have inappropriate computer and Web experience (too little or too much experience unless it is appropriate for the project) and those who are not very expressive.

2. Amount of participants

A long time ago (2000) Jacob Nielsen wrote that only 5 participants are necessary for a valuable usability test and that the gained insight diminishes rapidly after the fifth.

Usability.gov determed this number with the help of a formula, which is not that different from Jacob Nielsen's number.

Usually 3 to 5 respondents per round are enough to encounter many of the most significant problems related to the tasks you're testing. It's pretty much a certainty that you won't uncover some of the serious problems in a given round of testing. That is why you'll be doing more than one round.

3. Mention your objectives clearly to the user

Put the candidates at ease and run them through the software tools and equipment. Explain the objectives of the test, how long it will take and how the gathered data will be used.

Inform the participant that you are testing the product, not the participant's skills. Respondents have a tendency to attribute failure in the task to their own incapability, rather than a flaw in the design. Tell them they can't do anything wrong. In fact, the more mistakes they'll find while testing, the better. Stress this point more than once so test participants understand it clearly.

4. Choosing tasks carefully

Set tasks that are essential to the success of the new website or application, such as buying products, paying bills or contacting the client. If these 'top-tasks' are not clear to you, you could always ask the client which questions your research will need to answer.

People also tend to perform more naturally if you provide them with scenarios rather than instructions. Instead of asking them to find the contact section of your application, you could phrase it like a scenario. For example: "You fell down from the stairs and had to call the ambulance. You're wondering if your medical insurance is covering this and would like to contact them - Find the telephone number".

A scenario provides some context and supplies information the user needs to know, but doesn't (e.g. username and a password for a test account). It's important not giving away any clues in the scenario.

5. Ask your respondents to think aloud during the test

Think-aloud protocols, or TAP, involve participants thinking aloud as they are performing a set of specified tasks. Ask them to say whatever they are looking at, doing and feeling as they move through the user interface.

This method has several advantages. You'll know what your users really think about the design which could turn into actionable redesign recommendations.

6. Do not interrupt the flow of the participant's thought process

Shut up and let the participants do the talking. This is not the time to interpret their actions and words. As an observer or moderator, you should listen and take notes.

7. Don't lead the user

As a facilitator you should stay neutral, meaning you shouldn't influence your respondents or lead them to a desired result (consciously or unconsciously). If you do, your testing will lose its credibility. For example, when a test user is testing a sequence of screens and should click a button to continue, you shouldn't point out to the button or even mention the message on the button ("Continue").

Although a very difficult point for the moderator, allowing the tester to struggle is important and brings massive benefits. If you're being asked what they should do, respond with "What do you think?". The answer is very valuable.

8. Have the confidence to stop a user and refocus them on the task

Some respondents have the tendancy to loose track of what they were doing. Repeat the initial question or task to get them back on track. Do not lead the user.

9. Something about note taking

If you are the facilitator of the session, you shouldn't be the one who is taking notes. Instead, get the observer(s) to take notes. Give them specific things to look for. If you're both the moderator and note-taker, stop taking notes about things your not going to report on (either because of time or scope issues).

You could also try Morae's data logging tool. Techsmith made an excellent video about data logging with Morae and David Travis wrote about exporting this data to Excel.

10. Plan to quantify your results

When gathering data, it's easy to ask questions like "Did you think the navigation was clear?". You'll probably get a 'yes' or a 'no', but how will you quantify these responses?

To help you out, psychologist Rensis Likert came up with the 'Five-Point Likert Scale', in which the respondents specify their level of agreement or disagreement.

The format of a typical five-level Likert item could be:

- Strongly agree
- Agree
- Neiter agree nor disagree
- Disagree
- Strongly disagree

11. Prioritize, rank and list

Ask your subjects to prioritize, rank or list their answers, this instead of asking questions which will be answered with a "yes" or a "no". For example: "What are the three things you noticed on the homepage?" If none of them point out to the section which is important to you and the product, you should think about reorganizing the homepage.

12. Keep the tests short

The length of the test depends on many factors such as scope, amount of participants, the number of tasks, the duration of each task ... which is why a test can range from 15 minutes (for a single page design) to over one hour (full Website design). Studies exceeding 30 minutes have a higher participant drop-off because you are likely to lose their attention.

Do you have some useful tips on Usability Testing?
Some useful resources

Books:

- Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems - by Steve Krug
- Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests - by Jeffrey Rubin
- Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics - by Tom Tullis
- Other related books

Internet:

- Five Second Test - Landing page optimization for your mocks and wireframes
- User Testing - "The fastest, cheapest way to find out why users leave your website"
- Chalkmark - Online Screenshot Testing Software
- Silverback - Guerrilla usability testing software for designers and developers
- Facilitating a Usability Test - Christine Perfetti has several video tutorials where she addresses the role of the facilitator in a test.

About WalkMe

WalkMe™ helps customer support managers to increase self-service adoption, reduce incoming support requests, and lower service costs. Leveraging the *WalkMe*™ interactive self-guidance technology, support managers can insure their customers have a simple, smooth & burden-free online experience, eliminating customer confusion and frustration.

Through a series of interactive tip balloons overlaid on the screen, tasks are broken down into short, step-by-step guided instructions, which help customers act, react and progress during their online experience. As a result, customer support managers can empower their customers to self-task successfully even through the most complex processes. Moreover, *WalkMe*™ reduces your customers' frustration of waiting for assistance, shortens the time it takes for support personnel to handle an incoming request and strengthens your company's support reputation.

About the Author

Jessica Miller is a usability specialist & editor of Usability Lab, a blog focused on the issues related to web usability.