

Sniffing out User Experience Smells

Jose Cuevas
jcuevas@ea.com

Abstract

It's a common misconception amongst developers that they know or understand what users want or need. It is equally a misconception that users know or understand what they want or need. However, what users want or need is much easier to identify once they actually use the product or service. So how do you design a product or service without being able to accurately determine the requirements prior to releasing it?

"...when you have eliminated the impossible, whatever remains, however improbable, must be the truth", Sherlock Holmes (Doyle 1890, 111).

Like a great sleuth faced with the daunting task of solving an impossibly complex mystery, you must eliminate the different layers of misconception to reveal what undoubtedly must be the truth. There are many clues, also known as "smells", which can lead you to the truth about your design. However, these smells can only be sensed by those who are willing to evaluate their own product or service with an open mind and a proactive approach.

There are two ways to identify smells:

1. Study and learn material on subjects related to the user experience.
2. Test and observe actual users performing tasks with your product or service.

It has been my experience that testing and observation is not very effective if you haven't spent time gaining an appropriate level of knowledge with user experience design. I have also discovered that explaining poor design is usually met with resistance and disbelief by developers. However, seeing is believing and developers should eat their own dog food once in a while. It's not my goal to discuss a lot of technical jargon, but to instead guide the audience of this presentation on an interactive journey as we explore the user experience through the eyes of the customer and sniff out user experience smells.

Biography

Jose Cuevas is a software engineer at Electronic Arts (EA) and is currently working at the Tiburon studio in Orlando, Florida. Since 2005, he has been involved in QA and Test Engineering for multiple games and currently serves as a production support specialist and developer of tools and services for automated testing of EA Sports games.

Prior to joining EA, Jose spent almost ten years in the Navy as a Nuclear Electrician, Radiological Controls Technician, and Nuclear Instructor. Jose also spent two years at the University of Advancing Technology in the Game Design program prior to leaving the Navy to pursue his dream of working at a major game company.

1 Introduction

The User Experience (UX) is that feeling in the back of your mind that moves to the front when you're either engaged with or discouraged by a given product or service. More importantly, it's based on an individual's emotional response to a given situation and not a collective perception on how the world should operate to meet someone's needs. I'm always fascinated by the developer that has spent long periods of time in a dark hidden corner of the office designing this amazing application to provide some really cool feature only to discover that it's nowhere near what anyone wants or someone else has already created something similar.

The lesson learned is that feedback must be collected as early and often as possible if you want to create something that truly meets the needs of the user and can adapt to changes in requirements (which will undoubtedly happen once the user actually uses your product). In addition, early feedback can avoid wasting time developing something that won't even be used or can be provided by a less expensive alternative through another vendor.

However, feedback is only good if value is obtained from it. It is my primary objective to ensure that I provide a solid set of material that is easy to follow and understand and can be used to improve any team's ability make good use of the feedback they collect.

2 Understanding the User Experience

The foundation of any subject is knowledge. If you do not understand the material that is being presented, you will miss the point. Before testing, it is extremely important to know who the users are and what constitutes a user experience.

2.1 History of the User Experience

Every so often, man creates something that allows us to perform the impossible. In the early phase of that creation's adoption, the design is only good enough to be useful. When computers were first created, it brought with it new possibilities. However, there was no focus on making it the best experience for users, only to provide users with the ability to do something they could never do before.

As the hardware and software evolved, so did the user's expectation of using them. Once users accepted the possibilities of what a computer can accomplish, the next logical step was to make accomplishing things more efficient. Once that was accomplished, the next phase was to make the overall process more pleasant (an emotional response). It is here that the User Experience was discovered.

2.2 Who are the Users

Before we discuss the experience, let's get to know the users. In short, the user is the human consumer of the product or service. So why have an entire section on users? I have two very good reasons:

- The end-user is *not* the only user of a product or service that matters.
- To make a great product, you should focus on your target audience and not try to please all people.

2.2.1 The Other Users

A major product or service is made up many smaller components. To identify the other users, it's important to identify the owners and stakeholders for each of these components. There are usually several internal products or services used within a company to either improve the development process

or be consumed to complete functionality. The end-user may never be exposed to these details, but an internal developer will. There is a cost associated with creating a tool or service that does not provide a good experience for the developer. Yes, developers are very smart people that can craft amazing solutions to very complex problems. However, that shouldn't be an excuse to avoid spending time designing the tool or service to be easier and less painful to use or implement.

2.2.2 The Right Audience

It's also extremely important to identify the right audience for your tests. Everyone is different; however, your target audience will have common goals and requirements that can affect their experience. Too wide an audience can result in feedback that may have little or no relevance to the target audience. If your product has something to do with racing cars, then your target audience should have some interest in competing in a race.

However, it's good to obtain feedback from outside the target audience range when:

- Trying to appeal to a more casual user or new user.
- Having difficulty getting testers within your target audience. Any feedback is better than none and will still reveal issues that your target audience may experience too.
- When trying to identify a new target audience.

2.3 Define the User Experience

Have you ever been asked to perform a task or job that you have very little knowledge in? The first thing you do is look up whatever information you can find on how to perform the task you have been assigned. However, teams are okay with performing tests without ever providing any type of training for the observers in regards to how to identify issues. Then they go into the testing debrief and dispute what problems are actually valid and which ones take priority over the others. Before I begin providing the quick and dirty on how to identify user experience problems, I want to throw out a definition from Wikipedia that will be the foundation for everything else that we will be presented in the following sections.

***User experience (UX)** is the way a person feels about using a product, system or service. User experience highlights the experiential, affective, meaningful and valuable aspects of human-computer interaction and product ownership, but it also includes a person's perceptions of the practical aspects such as utility, ease of use and efficiency of the system. User experience is subjective in nature, because it is about an individual's feelings and thoughts about the system. User experience is dynamic, because it changes over time as the circumstances change.*
(Wikipedia 2012)

I will also add that the user experience is observed at any stage, multiple stages, or across all stages in the lifecycle of a given product or service. To summarize, the user experience is an emotional response by an individual to your product or service. Therefore, identifying issues requires knowing the individual on a personal level.

2.4 Company Experts

Depending on your company, you may have a team of people that are experts in the user experience. However, these experts are usually relegated to handling high level user experience problems (those dealing directly with your end users). Low level, or internal, user experience problems usually have no formal team that is responsible for their design and testing. Therefore, many internal tools and services suffer from very poor user experience problems which can have significant costs to your company because of the amount of time wasted by employees trying to use them. To avoid this pitfall, it is

important to train your developers on how to create a great user experience for your internal customers as well as the external customers and to also perform usability tests in a similar, if not the same, manner.

3 Top Ten User Experience Problems

An easy solution to getting quality feedback is by observing someone actually using the product or service. The best approach I have encountered for obtaining good feedback is to physically be a part of the testing process and get as many developers as possible to participate. There are a couple of things to keep in mind with this process.

Take a step back and discuss with your team on what things to look for (the smells of a bad experience). Some will be obvious because the user relayed the problem directly to you and others will be subtle and require some form of user experience awareness. I found Steve Krug's suggestions for usability testing in the books *Don't Make Me Think! A Common Sense Approach to Web Usability, Second Edition* (Krug 2006) and *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability* (Krug 2010) the most helpful at obtaining user experience feedback. However, the feedback is helpful only if the developers understand why the user experience is poor. Many developers believe that the users think the same as them, which would completely invalidate the need for the term *target audience*. As a consolidated guide of information from multiple sources (Krug 2006. Krug 2012. Microsoft 2010. Saffer 2010. Cooper, James, and Saffer 2010. Garret 2010. Hollis 2012), I present to you my top ten smells of a bad user experience and suggestions on how to fix them:

3.1 Overwhelming

3.1.1 Smell

Users either have a specific reason or general idea about why they want to use your product or service. Research has shown that users typically scan for relevant content as opposed to analyze (Krug 2006, 22). The easier it is to scan, the more likely it will be for them to get what they need. However, if a user is overwhelmed by the amount of choices available to them, research has also shown that users will just choose the first option that seems to make the most sense (Krug 2006, 24). Frustration will settle in if the user has to endure multiple bad decisions because the right decision isn't obvious.

3.1.2 Fix

1. Simplify. Show the most relevant content for the target audience and remove any noise.
2. Size. Research has shown that users tend to gravitate towards the bigger objects as opposed to the smaller object (Hollis 2012). Make the most important decisions bigger than the less important ones.
3. Placement.
 - a. Using conventions helps users feel comfortable knowing where to find things, such as the shopping cart in the top right corner of a website.
 - b. Location is important because users tend to scan a page in a particular pattern. People who normally read left to right will scan from top-left to bottom-right. People who read right to left scan from top right to bottom left. Important content should be within those regions.

3.2 Distracting

3.2.1 Smell

Competing features will prevent users from focusing on relevant information. Imagine you're in a stadium trying to find your friends who are seated in the stands. If everyone in the stadium is jumping up

and down and waving their arms, it becomes very difficult to scan for your friends. However, it becomes much easier to find your friends when there's a break in the action and the only people standing up and waving are them.

3.2.2 Fix

1. Limit moving parts. If there are too many moving parts, the users won't know what to focus their attention to. If every object has an animation for example, it has the same effect as a stadium crowd going wild.
2. Use color wisely. If everyone in a room was wearing gray and one person was wearing red, the person wearing red would stand out. Wise use of color can help you guide the user to certain options and make other options fade to the background.

3.3 Annoying

3.3.1 Smell

Hearing or seeing the same thing over and over again can be annoying. Hearing or seeing the same thing over and over again can be annoying. Hearing or...I think you get the picture. Another annoying trait is constantly disrupting the user with noises or useless information.

3.3.2 Fix

1. Don't ask questions you already know the answer to.
2. Don't repeat warnings if the issue never cleared and provide an option to dismiss the warning.
3. Don't interrupt the user with needless information or sounds.
4. Use sounds sparingly. Also, make them configurable so users can turn it off if desired.

3.4 Confusing

3.4.1 Smell

This is an easy smell. Do the users know what they are doing? If not, assume they have been let down and there must be something we can do as developers to improve their decisions making process. Don't assume they are dumb, though it's quite possible, but instead assume that we are not clear.

3.4.2 Fix

1. Limit the need for training. A well designed product or service can be learned on the fly. Incorporate the learning into the doing. Try to avoid making the learning a separate process because most likely it will be skipped and/or ignored. Remember, most users scan.
2. Obvious signs. Like the yellow brick road in the Wizard of Oz, make the directions to the Emerald City a clear path. Use good default values and communicate clearly where to go and how things work. Lead the user down the right path. As Steve Krug wrote, *Don't Make Me Think* (Krug 2006).
3. Empower users. If there is a problem, show users how it can be resolved without assistance from support.
4. Find support. If support is needed, make sure it is obvious who to contact and how.

3.5 Poor Feedback

3.5.1 Smell

Does the feedback provide any actionable value or is it just unnecessary info that is arbitrarily presented at any given time? The purpose of feedback is communication but it should be relevant information to improve the experience. Hungry customers most likely won't appreciate a long talk with the cashier if they know it will delay them getting their meal. Also, make sure the feedback you provide makes sense to the user and not the developer.

3.5.2 Fix

1. Provide feedback when there is a problem that limits or stops progress and when there is something actionable to be performed. Limit feedback that is nonessential information.
2. How would your frontline support team respond to a user if asked a question? That's how the product or service should respond. Developer information should be captured but not presented to the user. Collect that data to be accessed by developers later.

3.6 Constant Stranger

3.6.1 Smell

When you get to know someone, you eventually stop having to asking them who they are and what they like. Your ability to communicate with those people becomes more efficient over time. Users like to be treated the same way. Don't let your product or service act like a stranger every time it communicates with the user. Also, get a feel for what they like and find ways to improve their experience by suggesting changes or features that may be beneficial.

3.6.2 Fix

1. Adapt to User. Get to know the user to personalize the experience. Don't provide a lot of options that requires the user to configure the experience because it is less personal and can lead to user errors (i.e. bad experiences). Learn about their usage and automatically modify the experience to fit their needs.

3.7 Time Consuming

3.7.1 Smell

I last time I went to the local Department of Motor Vehicles (DMV) office, I scheduled an appointment online. When I got there, they told me I had to wait in line. When I got to the line, they told me it was too long and I had to sit down. This is why I hate going to the DMV. I spent most of my time waiting and not enough time doing. Users want to get or do things, not wait.

3.7.2 Fix

1. Consolidate. If there are a lot of steps for a user to complete, try to find a way to consolidate them.
2. Automate. Complete information for the user that can be calculated by or is a repeat of already provided data.
3. Reminders. If the process is going to take a long time, give the user the option to walk away for a while and remind them when the process is complete.

3.8 Frustrating

3.8.1 Smell

Read the blogs, listen to the feedback, and take them seriously. If you're doing a user test, observe their facial expressions, body language, or even spoken language to identify their level of frustration. You'll know when the user is fed up, however, you'll want to identify all the little things that may have led to the meltdown and not just the last thing the user did.

3.8.2 Fix

1. Gather feedback. Observe, listen, learn, and then correct.
2. Pay attention to details. Small things matter and users will appreciate you taking time to make the subtle changes that improve their experience. Also, be aware of how your product or service is perceived by users with disabilities such as color blindness.
3. Limit negative experiences. A lot of little problems will equal one big problem. Every little mistake is an excuse to leave. Give them enough excuses, they may never come back.

3.9 Unresponsive

3.9.1 Smell

Nobody likes to get a deer-in-the-headlights look from the product or service they are using. They want to know that something, even if it means telling them that nothing is happening.

3.9.2 Fix

1. Feedback. If there is a long running process occurring, then give some type of indication that there is something going on and, if possible, how much longer it will take.
2. Perception. If the loading icon is moving really slow, the obvious assumption is that the loading is moving slowing. Additionally, if it is moving fast, then the loading is moving fast. If it is always moving fast regardless of how fast the actual process is moving, the user will feel much more comfortable. It's like going to a restaurant and waiting a long time for your food and all the waitresses are just standing around. I want to see some action like they're extremely busy and that's why my food is taking so long!
3. Distraction. If you're going to make them wait, this is a great opportunity to throw some advertisements or useful information their way. They do this in the movie theatres now well in advance of the first preview.

3.10 Fails

3.10.1 Smell

Poor functionality is a user experience killer. The product or service needs to work as expected first and foremost. I've seen teams spend way too much time discussing how to improve the aesthetics of the view when the functionality is failing. Provide value then improve usability.

3.10.2 Fix

1. Focus on what the users want most and make them work great. Don't add new features until those important features have been cleaned up and verified to be working great. Users will be forgiving of bugs in your least important features if the main features work great.

2. Expect the unexpected. Be prepared for server downtime, patches, etc. As developers, we need to be pessimistic about our own product or services. It's marketing's job to be optimistic.
3. Test your product or service. This should be obvious, yet we still have to say it because it's still a problem.

4 Conclusion

The user experience is a combination of many different components that drive one thing, how a user feels about your product or service. Testing the user experience is becoming a common practice and we are constantly studying new concepts and experimenting with different techniques to identify problems and create better designs. However, it is important that developers are well educated in the definitions and basic principles that fall under the user experience umbrella to ensure that the improvement process runs smoothly and your team or company achieves the most success with your product or service. I have provided a brief list of what I feel are the top ten issues and how to fix them based on information from multiple sources. Like a good sleuth, you can use this knowledge as a tool to help solve the mysteries behind many user experience problems.

References

Doyle, Arthur Conan. 1890. *The Sign of the Four*. London: Lippincott's Monthly Magazine.

Wikipedia. 2012. "User experience." Last modified July 1. http://en.wikipedia.org/wiki/User_experience.

Microsoft. 2010. *Windows User Experience Interaction Design Guidelines*. Last modified September 29. <http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx>.

Krug, Steve. 2006. *Don't Make Me Think! A Common Sense Approach to Web Usability, Second Edition*. Berkeley: New Riders.

Krug, Steve. 2010. *Rocket Surgery Made Easy. The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Berkeley: New Riders.

Saffer, Dan. 2010. *Designing for Interaction, Second Edition: Creating Innovative Applications and Device*. Berkeley: New Riders.

Cooper, Alan, Reimann, Robert, and David Cronin. 2007. *About Face 3: The Essentials of Interaction Design*. Indianapolis: Wiley.

Garrett, Jesse James. 2010. *The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition)*. Berkeley: New Riders.

Hollis, Billy. 2012. *Creating User Experiences: Fundamental Design Principles*. Pluralsight.com.