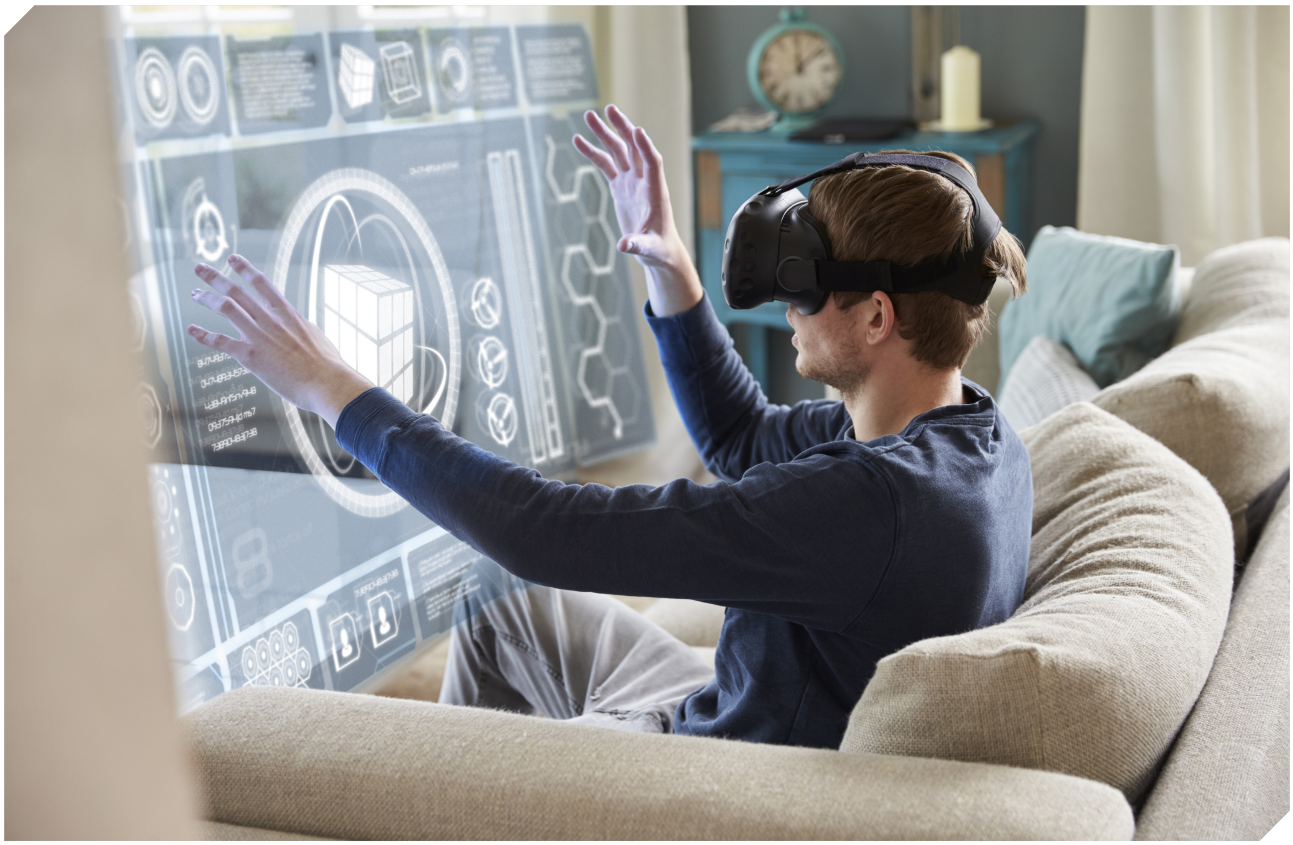
The Qt Company

WHITEPAPER

The Insider's Guide to an Excellent UX for Non-UX People

User experience (UX) is a term you can't avoid these days. That's because an excellent UX can more easily provide something that most other facets of your business cannot – a direct path to the customer's heart. If that sounds a bit too grandiose for a two-letter acronym, remember that more than anything else, the success of powerhouse companies like Apple and Tesla is dependent on their UX. An overwhelmingly positive user experience creates legions of dedicated customers and has the power to elevate your company into a cult-like brand.
This white paper will look at all things UX as it relates to product development – what it is, why it matters, and what you can do about it, including some basic best practices for achieving an excellent UX with your products.



Contents

What is a UX?

If you're not sure exactly what a UX is, you're not alone. The precise meaning of UX can still generate discussion and debate among UX experts, however the ISO organization has finally created a standard definition as a person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service.

The complete ISO definition of UX

UX is a person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service.

User experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours, and accomplishments that occur before, during, and after use.

User experience is a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use.

Usability, when interpreted from the perspective of the users' personal goals, can include the kind of perceptual and emotional aspects typically associated with user experience. Usability criteria can be used to assess aspects of user experience.

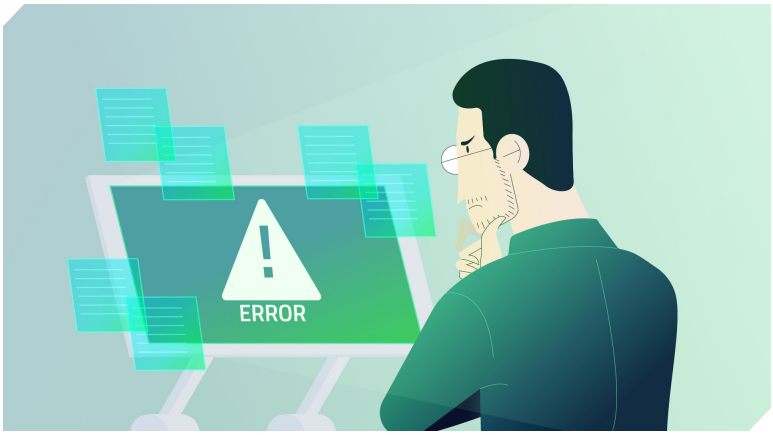
UX designers try to fulfill the user's needs by creating products that are delightful to use. This can be challenging to deliver since experience is a UX designers try to fulfill the user's needs by creating products that are delightful to use. This can be challenging to deliver since experience is a multi-faceted concept. Everyone's experience is different as it is formed by individual perceptions, actions, motivations, and understanding, so the resulting impression taken away by multiple users from a single interaction can be significantly different. An experience is also strongly affected by a user's expectations prior to the interaction. As an example, a streaming music service with a beautiful, elegant, and intuitive user interface won't always deliver the goods. If you can easily find what the music you're looking for, you'll likely consider the UX to be delightful, however if you repeatedly can't find the music you like, a great UI can't salvage the resulting poor user experience.

The User Interface (UI) is the means by which a person controls a software application or hardware device (graphical user interface, keyboard, mouse, buttons, switches, etc). As the means for the user to interact with the product, the UI is an essential part of a product's UX.

Another significant challenge is that a UX is defined by the user's feelings and emotions, which is not an easily measured characteristic. A great user experience should inspire passion, excitement, or intrigue. Product designers that focus only on improving aesthetics and simplicity may in fact have no effect on a UX or even negatively impact it. That's because a good UX is not merely defined by good usability. While usability – the ease of use and simplicity of learning for a product – is an important part of any UX, it's not the only factor.

A product with a more complex user interface may in fact provide a better UX if it contributes to the user's feeling of mastery, provides opportunity for discovery, or underscores the value of an expensive purchase. Whether a UX is good or not, is actually in the eye of the beholder.
It's also important to note that user perceptions, motivations, and expectations are not static. They change with the user's awareness of their environment, exposure to different technologies, competitive and market forces, and shifting needs.

> *Whether a UX is good or not, is actually in the eye of the beholder.*



Invisible Design

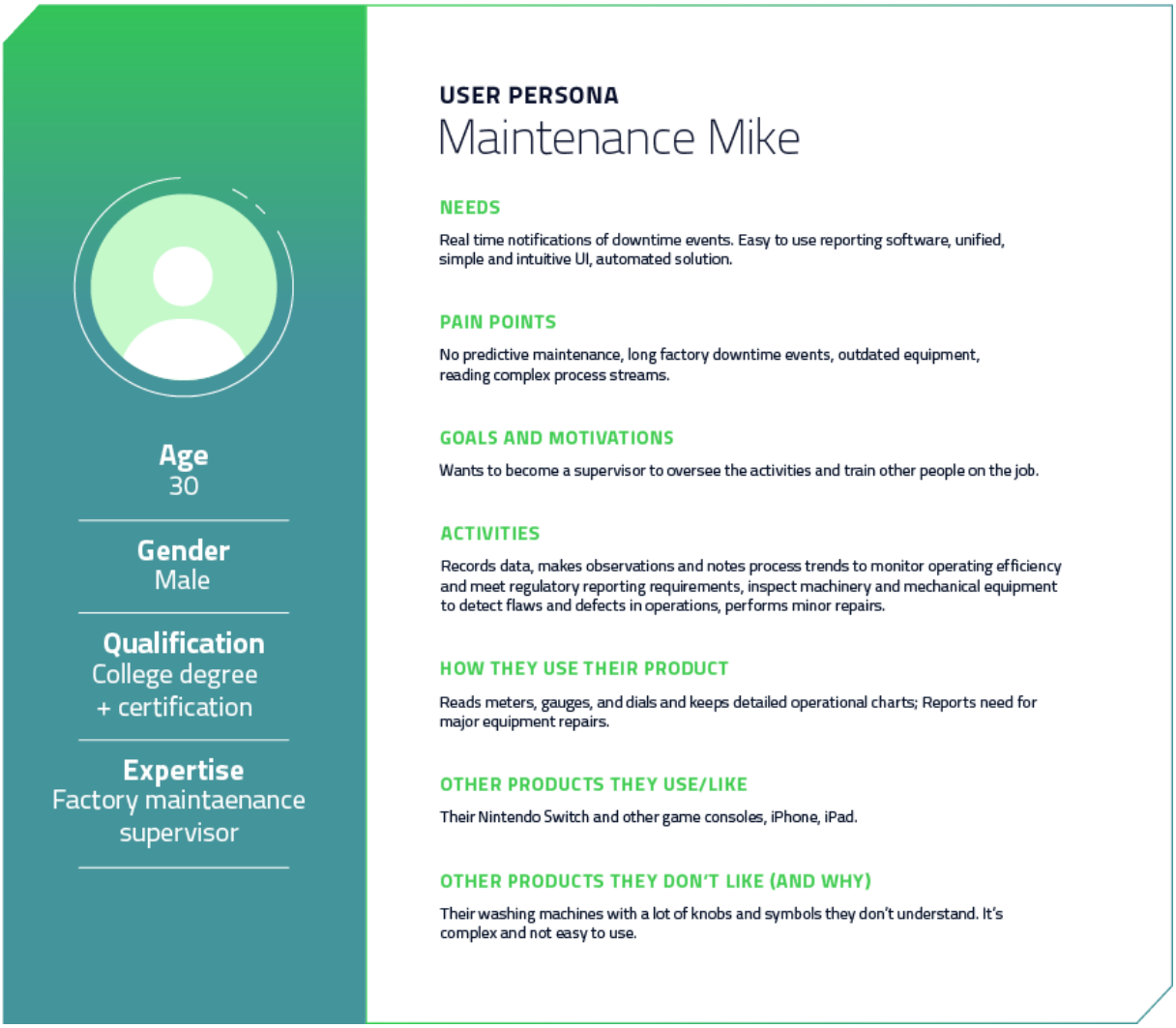
People notice design when it is bad, yet the best designs seem to evade our attention. While many design disciplines focus on making things of beauty, in UX design the core value is one of invisibility. As noted by UX guru Don Norman in his book *The Design of Everyday Things*, "good designs fit our needs so well that the design is invisible, serving us without drawing attention to itself." Sometimes invisibility takes the form of the visual – a design is simplified to reduce the sensory overload, allowing users to clearly focus on a task.

Other times, it reduces the interaction required by automatically taking care of the user's anticipated needs or by introducing options exactly when they are useful. Any way that it occurs, design invisibility helps products become a natural extension of users. UX designers achieve this by skilfully combining design education with product knowledge, human nature, and the world. The tantalizing promise of invisible design makes a great UX designer indivisible from a great product.

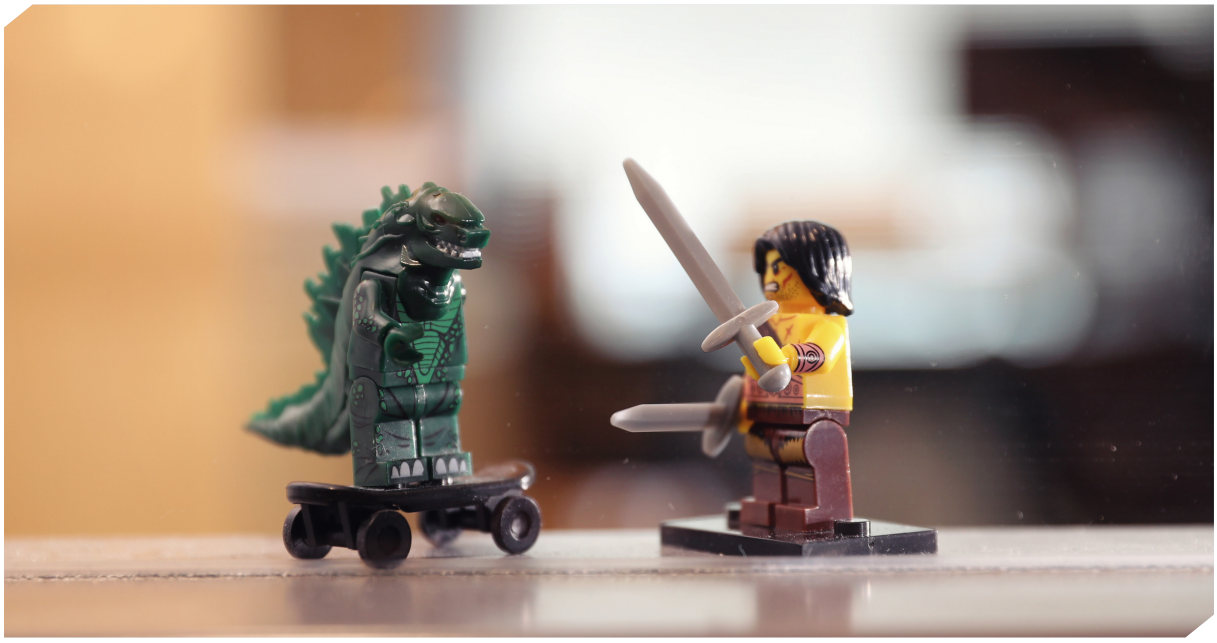
The User

Everyone can agree that a user experience needs to revolve around the user. However, it's essential to know who that user is and how they use a product. Even if you think you know your target audience well, building a profile that describes your user can help you better anticipate and focus on their needs. This user profile (also called a persona) will help guide you in answering questions about how they act.

There are lots of places you can find example templates for creating a UX user persona. However, the most accurate and convincing ones will come from researching your potential customers and conducting surveys, field interviews, and focus groups.



Creating personas requires quality research combined with astute insights. Asking users hundreds of questions about what they need isn't productive – users don't always know what they need, may not be able to express it, or may have conflicting ideas of what they want. Instead, ask users their goals, how they currently achieve those goals, and what their pain points are. If possible, watch them interacting with your product or a prototype in a real-life situation to develop insights on how the product gets used and what other things are important in the context. As an example, if you discover your product is mostly used in a noisy environment, you would add "works in loud surroundings" to your persona. With a contextual clue like that in your user profile, you'll know that building a design based around audible feedback may not be the best approach.



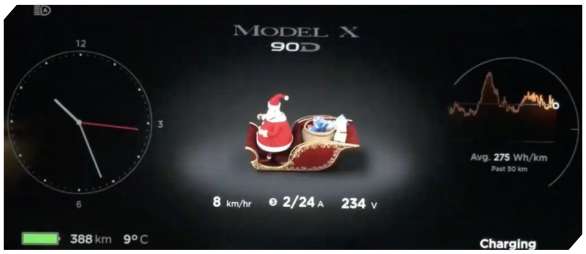
UX and Culture

A person's culture affects more than just localization preferences. User motivations tend to be universal and unrelated to gender or culture, however, user perceptions differ. In fact, people from different cultures will perceive the usability of a UX differently. Factors that contribute to a cultural viewpoint include perceptions of social equality, expressions of individuality, comfort with uncertainty, ease of adopting change, and attitudes towards indulgence. An application that rewards individuality by allowing completely custom user logins might be desirable in one culture and distasteful in another. Because cultural bias informs a user's world-view, an understanding of your user's predominant culture is crucial for creating a UX that best serves its audience. This will guide a UX designer in fundamental attributes such as whether users find more options desirable or excessive, whether they like to be guided or figure things out themselves, or whether the interface should be cutting-edge or more traditional in nature.

UX and Emotion

Emotional design is an important consideration when creating a product. A concept first introduced by Don Norman, emotional design strives to create products that elicit positive emotions in order to create a good experience for the user. According to Norman, the user forms an emotional connection with an object because it is appealing to the senses, which in turn makes them value it more. It may be far easier to imagine building a physically pleasing remote control than a more practical object like a CNC milling machine or a medical ventilator. However, understanding the idea of emotional-based bonds can help create an object that's worthy of admiration – regardless of what it is. It's important to note that emotions occur not just from the sensory experience of an object; a user feels different emotions throughout the entire user experience. For example, in a game there are definitive steps and a known progression that allows a designer to plan emotions that emphasize the story arc with feelings of struggle, calm, apprehension, or triumph. Productivity applications however don't incorporate such negative elements as struggle, fear, or disgust – at least not intentionally. And positive emotions in a non-gaming application aren't quite as easy to elicit or even define. What is joy in using an industrial machine control application? Do they feel safe, knowing the machine will alert them if something dangerous is about to happen? What about medical equipment? Do users feel good knowing that their equipment produces reliable results? And how about in the car? Do users feel joy when seamlessly connecting their devices to an infotainment system? It's important to know your market, your user, and how you want them to feel when using your product. When you know the app has unavoidable pain points, think about how to minimize them but also how to recover from negative feelings. As an example, your app may require the user to enter a number of complicated parameters. If so, consider rewarding the user with a visually attractive and interesting display showing the parameters, making it rewarding to overcome the pain of the input process.

Another challenge of emotional design is that users can become accustomed to product experiences. This is called hedonic adaptation and it happens when the product no longer captures the user's attention. We all seek novelty and product users are no different. Once users have adapted to a product, they will stop experiencing happiness when they use it, and any positive experiences they've had will fade over time. (There is a good side to this – negative feelings also fade, so poor experiences don't last forever.) To counteract the hedonic adaptation process, you can periodically introduce new features and new content every couple of months to keep your product fresh and appealing.



Tesla overcomes hedonic adaptation by adding Christmas mode to the digital instrument cluster.

First Time User Experience

The first time a user interacts with a product is different than all preceding interactions, enough so that first time user experience (FTUE) merits its own acronym. FTUE design is important because it guides the user in how to use the product and helps them understand what it does and how it works. A bad FTUE can turn off new users and extend the learning curve, or worst of all make users abandon the product altogether. Remember that your FTUE is laying the foundation for the user's understanding of the product, and the lessons they learn from first use will shape their understanding and their experience going forward. Some attributes to pay attention to in your FTUE:

Intuitiveness – design things that make intuitive sense from a newbie's perspective. An example is drawing from our understanding of the natural world and environment to have items fall, bounce, or similarly react like physical objects would

Familiarity – leverage the user's understanding of existing UI patterns that your personas have been exposed to. If you expect that all of your users will have an iPhone, having iPhone-like controls and gestures will feel comfortable, even for new users

Consistency – when you incorporate unique features or design metaphors, use them consistently throughout your product. Once the user has learned how your UI works, they'll always know what to expect and how to use it

Low barrier to entry (otherwise known as approachability) – make it easy for the user to start using your product; the fewer things required during the login/registration/setup phase, the better. That advice also applies to the initial learning curve. While some research indicates a higher barrier to entry can build more dedicated users, this comes at the cost of alienating new users

UX and Flow

Very closely related to emotion is the concept of *flow* – also called being "in the zone" or "in the groove". First defined by Mihály Csíkszentmihályi, a Hungarian-American psychologist, flow is an immersive experience where you are so engrossed with your task that the world around you vanishes. We've all had times when we're so in tune with what we're doing – experiences such as playing a video game, skiing down a mountain, playing a guitar, or cooking dinner – that we're 100 percent immersed in the task. When we're in a flow state, we don't notice the passage of time, our physical body, or any self-consciousness, and we feel attentive, content, and in-control.

> *"Flow is being completely involved in an activity for its own sake. The ego falls away. Time flies. Every action, movement, and thought follows inevitably from the previous one, like playing jazz. Your whole being is involved, and you're using your skills to the utmost."*
> *- Mihály Csíkszentmihályi*

Creating a flow state is the ultimate goal of the UX designer. While this concept is widely understood in game design, it is also very applicable to non-game applications. We want our users to not notice the application is in-between them and their goal. Flow requires that the difficulty of the task isn't too easy (or the user becomes bored) and not too hard (or the user becomes frustrated) – but just right. Reading references about how game designers achieve this goal may give those of us who are building more prosaic applications some ideas on how to achieve flow with our products.

The Experience

Now that we've covered the user, let's talk about the experience. The UI usually forms a significant part of a product's user experience. However, it's critical to remember that UX is not UI. An experience is much more than the appearance of an application or the look of a product. You should build an experience where the user feels in control during the entire customer journey from beginning to end. That means your product, your company, your service, your UI are all part of the UX, and all of them should work together to help build a positive relationship with the user.

> *UX is not UI – the experience is much more than just the interface.*

We'll be looking at aspects of product design that cover UI and UX in the following sections, with an understanding that the UI should always be considered as just a portion of the greater whole of the user's experience.

Configuration Defaults

Software engineers have a perennial battle in application development between adding more features versus simplifying for usability. Creating configuration options is often seen as the solution to this problem. This way, the majority of users can use the default settings, while advanced users can configure the application to their wishes and expand the application's capability. Before you fall into the trap of exploding configuration options, remember that studies show less than five percent of users actually change their default settings. It's far better to try to recognize what the user is doing and provide contextual tips, or assess the user's skill level and reveal new options. Adobe Photoshop and Illustrator are extremely sophisticated programs for graphic design and illustration. Due to their vast number of features they have unavoidably steep learning curves. However, Adobe has been able to hide a good deal of their applications' complexity behind consistently laid out toolbars, tips and tricks that are visible on demand, and intelligently selected presets.

As we stated before, application complexity isn't just a UI problem. To make the best user experience, you should anticipate a user's skill level before they run your app so that you can automatically configure it for expert, advanced, or novice use. How you achieve this depends on what data you have available – how have users interacted with similar products in the same portfolio? Do they have an existing login on your system (with historical data) or an upgrade? Allowing your product to become individually tuned to the user's expectations makes for a much happier user.

Since less than 5% of users change their default settings, don't bury options in a configuration setting. Instead, provide contextual tips or reveal options based on a user's skill level.

Handling Errors

When everything goes smoothly – no hardware failures, bad data, or improper inputs – it's much easier to design a good application task flow. The real difference between good and great UI design is how hiccups are handled. We've got a few tips on how to turn error handling from intolerable to acceptable.

Allow the user to recover.

Error messages should tell the user how to correct the problem or let them know there isn't anything they can do to correct it. For example, if it's an online service and there are connectivity issues, telling that to the user allows them to fix their WiFi connection. But if the back-end service is down, you should tell them so they don't waste time trying to fix their Internet connection.

Know your user.

If your user is an IT specialist a message like "/dev/hd3 seek fail, swapping to RAID" may mean something valuable and important. But if they're using a home security system, a message like that may be pretty disconcerting and could lead them to distrust your system. Knowing your user's level of knowledge and their ability to recover is important to understanding how much detail and what warnings or errors should be shown. You should always attempt to speak the user's language – the terminology you use should come from the user's vocabulary and be based on their knowledge and expertise

Don't burden users about problems they can't fix.

If your product fails in a catastrophic way, giving users a long string of hexadecimal digits or technobabble to read to a support technician isn't a great solution. Better that your techs remotely query the product without inconveniencing the user and better still if your product fails gracefully by degrading performance or is fixed via over-the-air software updates. An example of a graceful failure would be a product with an always-on active Internet connection that can handle a broken connection by continuing to work without real-time updates and re-synchronizing when the connection is restored. The same advice goes for reporting errors users can do nothing about. If there's no way the user can fix a problem, you're better off to display a red "trouble" light rather than give them a technical report they can't use. When your car shows a red engine light, that means more to most users than seeing that an O2 sensor isn't working properly – the red light tells them to bring it to a service shop, while a malfunctioning O2 sensor isn't something they're likely able to understand or repair themselves.

Give the user control.

When the user loses a sense of control it creates a negative reaction. If a product encounters an unforeseen circumstance and there are multiple paths to work around the error, the best course of action is to let the user decide. As an example, if a document is missing a font, asking the user if they'd like to download and install the font or substitute an existing lower quality font lets the user choose which option is better for them. They may choose a different option depending on what they're trying to do or the time they have to do it.

Don't lose information.

Losing the user's information is one of the worst things you can do and you should only do it when there's no other option. This type of problem often shows up when handling errors in user forms. If the user fills in a form and then encounters a database, permissions, or connectivity error, there's nothing more frustrating or irritating than to have to retype all the information once you've rectified the error. Don't make your users repeat themselves – nobody likes to be ignored. For a great example of how to keep the user's information intact, look at Microsoft Office. The Office suite's auto-recovery feature is a great example of reducing user frustration in the face of adversity. If applications were perfect, they would never crash – and you could argue that implementing an auto-save and auto-recover feature is a waste. The reality is, anyone whose file has been saved by an auto-save or auto-recover feature is exceedingly grateful that the app hasn't lost their work. Protecting users from application bugs may have taken Microsoft more work to implement, but it's a way to turn what could be a hugely negative experience into an overall positive one.

Don't interrupt the task flow.

If the user is productively working, try not to interrupt them unless absolutely necessary. The user's time is valuable, so if you can handle errors outside of a workflow, do it. And try to keep the user's interruptions to a minimum by batching up errors if there are more than one happening for the same reason.

If the user is productively working, try not to interrupt them unless absolutely necessary.

Here are a couple of examples of these error-handling principles in action.

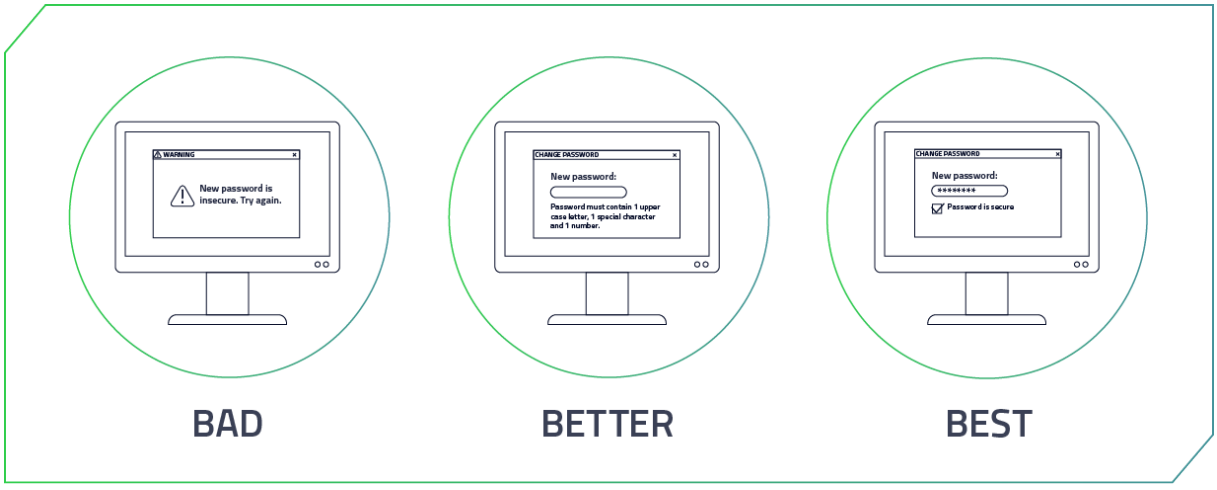
A bulk file copy that encounters existing files in the destination directory:

Prompts the user with, "Overwrite: Yes or No" for every single file.

Encounters the first file to be overwritten and asks the user if they would like the same answer to apply to all subsequent files.

Brings up a list of all file conflicts and allows the user to check or uncheck which files they'd like to overwrite, then allows the remainder of the files to be copied to a new destination.

A password reset that asks the user for a new password:



Sees that the password matches an older version, or doesn't contain enough special characters, or isn't long enough, and fails telling the user, "New password is insecure. Try again."

Shows the user in advance the password criteria and if the user's password fails any of the criteria it tells the user what is wrong and asks them to try again.

Shows the user in advance the password criteria, indicates when each criteria is met using green checkmarks as the user types, and only allows the user to proceed when the password is fully acceptable.

Instant Feedback

It's best to provide the user with feedback to their actions as soon as possible. Feedback allows the user to understand what actions they've done that led to any particular situation, and lets them correct or adjust as needed without undoing or redoing too much work. To revisit the password example, showing a bar with the password strength that grows longer or changes from red to green lets the user know exactly when they have created a sufficiently strong password. Instantaneous feedback also gives the user the sense that they are working with a reactive and responsive product, adding to an overall positive experience.

Studies into the psychology of learning show that a small delay between an action and a related event creates the strongest mental association when the feedback event occurs within 150-500 milliseconds of the action. If your software is trying to condition users to react to certain stimulation – such as avoiding an error or providing a reward – consider inserting a small delay before providing your feedback to naturally tap into our innate learning circuitry.

UX and the Funnel

The entire product acquisition journey is part of the UX. Marketing is what allows your potential users to find your product – web, online, advertising, emails, tradeshows, social media, etc. Once your prospective users are aware of your product, they'll need to evaluate it. How your product is portrayed, reviewed, and discussed forms part of the user's expectations, whether your sales cycle requires a visit from a sales team, a downloaded product brochure, or online research and comparisons. The expectations formed during the product acquisition phase are realized once users start using their new purchase. If the new user's expectations are met, they may have a wonderful UX. However, if they're off the mark – even if the product is otherwise fantastic – you'll have disappointed users.

Have you ever seen a trailer that looks like a comedy, but the actual film turns out to be a serious drama? Even if the film is good, it leaves you with a negative UX because of mismatched expectations. To avoid disappointment, the marketing and the product need to be in alignment. One way to help align user expectations constructed during the marketing phase and those encountered in using the product is to do task analysis to be certain you're focusing on salient points that help your users. Then, be certain that the way your marketing expresses those points are faithfully encountered in the use of the product. If you have a portable speaker that advertises, "works with all mobiles" but Bluetooth can't be made to work from an Android phone, the product won't fulfill its user's expectations.

It's best to figure out the task flows that solve your user's pain points, and build your marketing campaign around these while making sure you still weave in your product's unique selling proposition. If the user knows that their hardest problems will be solved with your product, they're already going to be in a good frame of mind before they start using it. If you've created a streamlined and simple way to get your users quickly solving their problems – with a minimum of installation, configuration, or troubleshooting – you're going to deliver on the expectations you've built.

Consistency is King

A consistent UX is a good UX. Similar to a product's need for reliable execution, the user should be able to rely on getting the same experience every time. This need for consistency certainly applies to the UI as a portion of the UX. If your UI uses a consistent design language and metaphors, the user knows what to do even if they're uncovering a new portion of the interface. Once a user has learned something, that knowledge should apply everywhere within your application.

Most users can be presumed to have an existing knowledge base about design elements and how they're expected to behave. Don't squander that knowledge; use it to your advantage. Companies like Apple, Microsoft, and Google have made their design patterns well known and commonplace. Don't re-invent the wheel when it comes to platform conventions, design patterns, or display styles. If you are creating a Mac application, use OSX style widgets, gestures, and paradigms, and the same goes for applications running on Windows, Android, or iOS – use the native design language whenever possible.

If your product is pervasive enough, you may want to build your own UI/UX that remains consistent across platforms. An example of this is Netflix – whether you're using it from your Android phone, your Smart TV, or your XBox, you can expect a similar experience. Consistency is still the rule: services that offer their product on many platforms try to minimize their differences in the user's experience. Whether a product-consistent cross-platform UX or a platform-adaptable UX is better for you depends on your user's expectations. If your product is available on many different platforms and your users will be actively using more than one, a product-consistent approach may be best. If your product is such that the user will only use one platform at a time, adapting your UX to the platform may be better.

Within your application, you should use similar elements, typography, and icons. If your application is reliant on a visual hierarchy, that hierarchy should remain the same regardless of where you are within the program's menu tree or application state. Branding should not change within your app, even if you're using someone else's sub components or services.

Consistency applies to the overall UX too. The interactions with your company should be consistent throughout the user's interaction with your product. Your product, your sales team, and your support staff should all work synchronously to reinforce the brand promise and bolster the user experience.

UX and KISS

Much of the user experience is about the KISS principle (keep it simple, stupid). Determine what in your product is essential versus what's nice to have. This means that – no matter how much the engineers want it, sales begs for it, or management requests it – the UX team and product management need to work together to put the brakes on feature creep. One way is to focus on items that differentiate your product. If you understand your competition and your target audience, you should be able to narrow in on the feature set that will define your product clearly and cleanly. Simplicity of purpose will significantly help your users to have a streamlined experience when using your product.

Building a Working UX Process

At this point you may be wondering how you can incorporate a good user experience into your company's workflow. You'll have to infuse an awareness of UX principles throughout your entire organization since your product's UX lives and dies with internal collaboration, with the UX designer as the hub.

Your UX lives and dies with internal collaboration.

Your UX designer acts as the voice of the user and helps other departments understand the user's needs and how the user wants to approach things. The UX designer helps the entire organization by:

Providing user stories and user journeys to the product manager to help shape product requirements in a way that makes for a positive UX

Collaborating with graphical designers to help polish the UI look and ensure there's no confusing elements

Becoming an integral part of the R&D team, working with developers to understand technology limitations and opportunities for improvement, helping polish the feel of the UI, and ensuring that the intent of the design remains intact when it is translated to implementation

In order for the UX designer to have the right insights about their users, they'll need to spend a lot of time familiarizing themselves with the people behind the personas. This requires a good amount of industry and domain knowledge. While some of that can come from analyzing the competition and the market, it takes time to really understand your users and the challenges they face. If you're a UX designer just coming into a new market, don't expect that you'll innately understand your users, their needs, and their motivations.

Iterate, Iterate, Iterate

Design is incremental, and nothing is perfect, including a UX experience. That's why iteration is such an important part of the design process. By figuring out what works and what doesn't, you can determine what needs refinement and what can stand as is. Your design will be built on top of previous designs and decisions in a continual learning process.

To kick-off the UX design process, an old-fashioned pen and paper approach is very valuable during early phase ideation. Unlike specialty UX design tools, paper is inexpensive and anyone can use it. Paper also supports fast iterations for new concepts, and it's disposable if you need to start over. Using simple sketches on paper can be the fastest way to move from a basic idea to user input.

Quite quickly the paper designs will have served their purpose and you'll need something more concrete to test out your UX ideas. Creating an interactive prototype is the best way to validate a concept with real end-users and other stakeholders before it moves into the development phase. A prototype should feel like a real product to prospective users as much as possible. You don't want to waste effort in actually developing the real thing, but you need a prototype with enough fidelity that you can draw accurate conclusions from it. The prototype's adherence to the product has at least three components:

Visual: Whether it looks like a sketch or is styled similarly to the real thing

Content: Whether the data within the prototype is realistic

Functional: Whether workflows mimic the actual product interactions

> For the functional part of the prototypes, you'll want to build screens that illustrate the main screen flows as well as include more detailed interactions to test some of the key features. A fleshed-out workflow in the prototype will be essential to getting the feedback that is necessary for refinement.

There are a number of prototyping tools on the market that can make building a UX easy, and many of these are easily adapted for actual product use. Your interactive prototype will help you validate the design against key user journeys, find user pain points (as well as moments of delight), and understand user navigation flow. Your prototype will also be a useful tool for sharing your ideas about the product UI/UX. With an interactive prototype you can communicate your design intent to developers, managers, and customers.

Design iteration and development can occur in parallel to speed up the process – something that's well suited to an agile development model. A dual-track approach starts iterating on the design before the implementation phase begins and it typically focuses on one user story at a time.

After you've got an initial UX prototype working, start development while you test and refine your prototype. Any alterations that become evident during the UX design/prototype/test loop can be fed back into development scrums. These things help keep the design effort reasonable and minimize wasted development effort, while the overlapped tracks minimize the time necessary for development to begin. It also makes sure that design is tightly woven into the development loop so it minimizes the chance of designs being created only to be thrown away later.

Evolution Over Time

Your prototype may also be able to help you highlight how the use of an application will change over time. The way a user uses a product will evolve from first use, to casual use, to regular use. As the user's needs change, so should your product. Making a product that still pleases the user after a year requires planning for the usage patterns of both new and experienced users. As an example, a first-time user of an email program needs to configure accounts, and manage a small number of emails without much scrolling. After a year, that same user will need to search through thousands of emails, mark emails as read and unread as they're processed, and mark some as priority to find them easily. Testing for this program will need to validate a positive experience with 10 emails as well as 10,000.

UX Testing

How do you test a user experience? You need three essential factors: product, users, and metrics. The product may be a prototype or real hardware, depending on where you are in the development cycle. Whatever it is, you'll want your prototype to be as close to the UX design as possible with the same colours, fonts, responsiveness, tactile inputs, etc. Your user testing will help validate the good parts and isolate the bad, letting you determine if your product will be usable as well as desirable. And finally, you'll need performance metrics that you can measure against, starting with a baseline to gauge if any of your UX improvements are successful or not. Example metrics include:

Functional qualities of the user's tasks such as total task time, number of clicks required, task completion rate, or number of errors encountered

Emotional aspects such as joy, frustration, or boredom that users experience while using your product

Page views, downloads, and installs, which help track the early part of the user experience and whether the user is interested enough early in the funnel to install your application

Rating stars, reviews, or customer feedback for an unsolicited glimpse into the user's high or low points

Your UX testing can be either qualitative or quantitative. In qualitative testing, your goal is to gather insights about the needs and motivations of your users through anything from a small focus group to an online survey with a global audience. In qualitative testing, you are validating usability, getting user research, and acquiring feedback directly from users. The resulting user surveys, use case tests, video recordings, and observational reports help you identify problems in the design and get insights into the user experience. Quantitative testing is the numerical side of the UX testing equation. It provides an indirect method of determining the usability of a design through the user's behaviour. The logs, events, timings, and counts from a quantitative test may not let you understand a user's motivations, but they do help you understand how users are using the product. Because measurements can be collected in a way that doesn't intrude on the user's thought processes, collecting quantitative data is often far easier than collecting qualitative data.

However, because quantitative data cannot tell you the user's "why", it is best used to measure UX performance against a baseline, to track usability over time, or to derive statistics on user behavior. If your quantitative data shows there is a persistent UX problem, you can always supplement it with qualitative data from user research to help identify the root cause of the problem.

> *"The problem with the designs of most engineers is that they are too logical. We have to accept human behavior the way it is, not the way we would wish it to be."*
>
> *- Don Norman*

Summary

Hopefully after reading this paper, it will come as no surprise that the companies that invest heavily in UX are market leaders – Amazon, Apple, Costco, Disney, and so on. These companies invest in UX because it helps them be more competitive through better products, reduced development costs, happier customers, and above-average revenue growth.

Research by Forrester (and commissioned by Adobe) shows that experience-led businesses have 1.6x higher brand awareness, 1.9x higher average order value, 1.7x higher customer retention, 1.9x return on spend, and 1.6x higher customer satisfaction rates.

If you're not paying attention to the needs of your users but your competitors are, you'll soon start to see an impact on your business.

While investing in your product's UX might take some time there's certainly no time like the present to get started. If we can help you create an experience-led product, contact us for friendly advice.

The Qt Company develops and delivers the Qt development framework under commercial and open source licenses. We enable the reuse of software code across all operating systems, platforms and screen types, from desktops and embedded systems to wearables and mobile devices. Qt is used by approximately one million developers worldwide and is the platform of choice for in-vehicle digital cockpits, automation systems, medical devices, Digital TV/STB and other business critical applications in 70+ industries. With more than 250 employees worldwide, the company is headquartered in Espoo, Finland and is listed on Nasdaq Helsinki Stock Exchange. To learn more visit http://qt.io

About the Author

Dr. Antti Aaltonen

Dr. Antti Aaltonen is the Head of User Experience at the Qt Company and leads the design of tools for UI/UX designers. He had always liked working at the intersection of design, technology and product development and he has 20 years of experience creating usable, delightful and polished experiences with products. Before joining the Qt Company, he was leading UX in Rovio, the creator of Angry Birds games, for seven years. Prior to that Antti spent over a decade with Nokia and was one of the founding members of the Nokia N9 UX team. He received his Ph.D. in Computer Science (University of Groningen, The Netherlands) in 2007.