

planning & design

Ask questions. People you should interview include executives, product managers, product marketing, salespeople who will sell this product, sales engineers with experience installing similar products, technical writers, customer support staff and quality testers. You will need to develop good relationships with marketing and distribution staff to bring them into your area of influence. Work with sales or product marketing people to get access to users and customers. Teamwork is essential to creating a great software experience.

Take a brand inventory. Does a strong corporate brand exist? Are there product line brand standards already in place? If so, the product being designed needs to complement and extend the existing brand. If there aren't brand standards already defined for products, create them. Collaborate with the marketing communications department to keep everyone on the same page.

Study the competition. Aggressively analyze the top 3-5 strongest competitors and document their brand, strategies, strengths and weaknesses in a **Competitive Analysis**. Use their products if you can, and pin their screenshots to your wall... this is who you will "design to outshine."

Communicate your intentions. Document the various activities that will happen throughout the software process in a **User Experience Plan** and include it with the design deliverables (along with requirements and functional specs) to help make user experience part of the process and development timeline.

Explain who users are. You may have a varied user audience, or customers who aren't users (enterprise software is often purchased by the highest level of executives, but used by people who did not choose the software.) **Personas** of fictional users and customers can help people across the company understand the vision and design decisions.

Write your own requirements. Writing your own user experience-related and product branding **Requirements** will elevate them to a level of importance and help ensure they are done before software can be released, and remind marketing and developers of the additional work that must be accounted for in the delivery schedule.

Use scenarios to depict different vantage points. Consider the product from the viewpoint of a first-time user, a frequent user, an inexperienced computer user, an advanced user, one who will use many features, one who uses only a few features, etc. Use **Scenarios** to define the ideal experience you want each of these users to have and then design the software to accommodate the needs of varied users.

Work to make user experience part of the process. Unless you are part of an established UE department, chances are user experience is a nice-to-have aspect of the software development process. The user experience specialist has to work to get **UE Processes** designed and gain sign-off approval along with others required to pass from phase to phase. Propose optional and non-optional activities, so that you can add more influence over time, as people see how much value the user experience activities can add.

Create a Product Design Guide for developers. Guides should include all visual standards, such as primary colors, secondary colors, fonts, styles, chart colors, and interface elements such as buttons and dropdowns. Our **PDG** also includes layout prototypes for all screen types, product taxonomy & definitions, navigation design, the icon library, documentation styles and standards, and packaging design samples, so developers see the **big product vision** beyond the scope of their own screens. PDG's can be distributed as PDF documents, but setting them up as an intranet site in HTML allows for links to simulations and resources, easy changes to standards and notices from the user experience group, serving as a constant instructional tool.

Detail interaction and interface design. Use **Wireframes, Visual Prototypes, Sketches & HTML Product Simulations** to communicate your intended design to developers.

Define the product taxonomy. Consistent word usage is key to a great user experience. Define labels, terms and phrases in a **Product Taxonomy** document and train salespeople, marketing and customer support staff to use them when writing about or interacting with people to avoid user confusion.

An awesome user experience includes visual design and usability. No aspect of the interface should be overlooked to take advantage of building brand equity: login screens, about screens, the installation experience, reports, charts, help systems and desktop icons used to access the software are all opportunities to build the brand, polish the overall impression and delight the user.

Usability testing validates visual theories. **User Testing** in the planning & design phases can be conducted using sketches, wireframes, visual prototypes or a rough working site to identify issues and find problems with the design.

Task analysis aids product acceptance. A product has to meet users needs in order to sell in a competitive market. Using **Contextual Observation, Field Research** and conducting **Interviews** will take the guesswork out of the design and save time and money in development.

coding & testing

Brainstorm complex functions with developers. Time and again, I have seen product marketing and user experience folks try to "dictate" how a feature should work, rather than collaborate with the developers who will code the feature. Save time and eliminate guesswork by working as a team with your developers, especially when designing difficult or integrated features and functionality.

It's software, but create a sitemap. Developers use data flows to show each other how to code the experience. We start with a **Sitemap**, just like we would with any complex site, ecommerce site, or web-based application. A sitemap is easy to review with managers and developers to discuss flow, navigation and key features.

Icons will take a lot of time, thought and management. Regardless of who designs your icons, you will spend far more time thinking of concepts, approving icons, getting them into an easy-to-reference icon library, naming them and communicating with developers to get them used correctly than you anticipated. Train QA testers to look for icon issues to make sure nothing falls through the cracks.

Usability testing eliminates surprises. **User Testing** should be conducted using sketches, wireframes, visual prototypes or partially completed software to identify usability and learning issues. Be sure to prioritize your findings.

Review software development often. Make frequent checks a part of your own internal process, whether you get regular builds or have to look over developers shoulders, to prevent experience design factors from being overlooked or ignored. This will also help keep you in the loop so that you aren't surprised by inevitable feature changes or additions that weren't part of the original design.

Be aware of the bugs. Get access to the bug-tracking system, or request regular reports, for several reasons. You can see areas of difficulty, track UE-specific issues, and add your own bugs to the system when you see something not designed to standard, or an incorrect icon used, etc. Keep cosmetic, brand standards and usability issues important factors for release, or they may only get resolved if there is extra time.

Conduct a formal design review. A formal **UE Design Review** should be led by the user experience manager and include stakeholders from management, marketing, user experience, development and others. It must be done when the software is complete enough to get an accurate depiction of the user experience. Issues discovered should be prioritized for correction before launching.

marketing & distribution

Packaging is more than a box and CD. You need to think about and design a plan for the OOBE (out of the box experience.) This might take some negotiation, because marcom generally has the responsibility for creating packaging, but this is an area where the user experience specialist has to influence them to benefit the user. Visual design of packaging should be an extension of the product brand (not just the corporate brand) and inserts such as getting started and help manuals, installation posters and various errata need to be professionally written and designed.

Someone has to write helpful documentation. This may not seem like it is the user experience specialist's responsibility, but it is critical to the overall experience, and must be in your area of influence. If the company does not employ tech writers, and developers are writing the documentation, the UE specialist should edit and improve the instructions.

The software experience involves more than just using it. Users have to "learn" to use software, and in that regard, we must also become teaching specialists. In addition to a traditional online help system or printed help manual, users need: getting started help (a manual or on-screen dialog), tutorials, context-sensitive help and on-screen guiding text. Graphic and style standards should be developed for all areas of "help" that coordinate with the product brand.

Get involved with beta testing. Your marketing relationships will be important if you want influence on beta testing, which is often (unintentionally) marketing-focused and not truly user-focused. Help product marketing plan the beta program, write test questions, and follow up with users afterward to ensure that you get valuable user feedback that will help you make improvements to the experience. Ideally, contextual observation on-site during beta testing can yield amazing results and insights to improve either the immediate or future versions.

Review packaging and marketing collateral before printing to help catch errors. In the ultimate user experience, the message is aligned from the beginning (pre-sale) to the end (customer support, maintenance, upgrades, additional sales.) This will not happen unless people constantly keep the big picture in mind, and then work through the details across various departments to make it happen.

Plan for brand alteration. If a deal is struck with a partner or channel, your beautifully designed user experience can be altered. Work with business development to define standards of acceptance (in your brand's favor) upfront. They often don't realize the cost to your own brand by changing things, so think about how the design could be modified for partnerships or customer personalization and plan for it in advance.

sales & support

Ecommerce usability is as important as the software itself. If your software is available in an online store, you need to work with marketing or whoever is responsible to make sure the purchasing and upgrade experience is as nice and seamless as the software experience is... they should be viewed as part of the same system.

Customer support is part of the experience. User experience duties don't end when design documents are delivered to developers or the software is finally released. Now that the software is on the market, customers will inevitably run into issues and a great source of information for improving the next release can come from the customer support staff. Ask them to inform you of any usability-related issues, and request tracking information about call frequency broken down by top issues. Fix the biggest complaints to keep the user base happy - for every complaint you get, there are probably 9 others that did not call in and complain or ask for help.

View the support sites as an extension of the product. Users will want to download products and patches, search through knowledge bases & faq's, submit help tickets and help each other in community forums. This is part of their user experience and web sites should coordinate with the product brand for the most immersive experience.

Salespeople need you to consider their issues. A salesperson interacts with the software a little differently than the user does. What does the software look like in a demonstration or sales presentation? Have you made sure data exists in the product, and default settings are chosen, so that your sales staff can effectively demo and sell the product? This is yet another facet to consider when designing the software, and you should interview sales staff to find out their biggest needs and obstacles and then make sure they can sell your software with pride!

Salespeople can provide valuable feedback. Make it a point to solicit feedback from sales staff after they have sold the software for a few months. Their impressions are usually quite different from the concerns and thoughts of people developing in the trenches, and the additional perspective will help you.

Ask for feedback from customers and users. Work with product marketing on **Feedback Surveys & Interviews** to be done a few months after the software is released. Document this information as requirements for the next version and use it to prioritize new features and content your users need.

There's a better way to do it. Find it.

Thomas Edison