Principles of UX/UI Design

Notes by José A. Espiño P. [1]

Spring Semester 2023–2024

Contents

1 Introduction

In the design spectrum, the widest area is known as **Service Design**. It is all–encompassing and includes all the interactions the user has, analog or digital. Inside Service Design, we find **Customer Experience Design** (CX), which refers to all the interactions the customer has with the service provider. As opposed to CX, **User Experience Design** (UX) is focused on all the interactions between the user and a *specific* touchpoint. UX usually is associated with digital products, but it can also be applied to physical ones. It is important to note that UX is not about visuals, but rather about the whole experience. Lastly, we have **User Interface Design** (UI), which is a subset of UX and focuses on the visual aspects of the product. We are going to mainly focus on UX and UI in this document.

UX, or user experience, includes everything related to **all** the interactions a user has with a company, its services, and its products. Successful UX design is an iterative process, in which you observe users, identify issues that may arise, brainstorm solutions, prototype these solutions, and test them with users; you repeat these stages until you have a product that meets the needs of your users.

The main goal of UX design is **usability**, how intuitive the product is to use. Usability can be broken down into five sub–components:

- Learnability: How easy is it for users to accomplish basic tasks **the first time** they encounter the design?

- Efficiency: Once users have learned the design, how quickly can they perform tasks?

- Memorability: When users return to the design after a period of not using it, how easily can they re–establish proficiency?

- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

- Satisfaction: How pleasant is it to use the design?

Throughout the UX process, documents named **artifacts** are used to help teams organise and present information. These artifacts include:

- User Personas: A character or model that portrays a possible user of the product. It normally comes from user research and includes information such as the user's range, age group, sociometric status, and goals. Some common user research methods are customer interviews, making assumptions, using web analytics, and conducting surveys. Once the user research is done, the information must be divided into customer persona groups which will then be consolidated into single–user personas, concentrating on the primary needs of the most important user group and one significant pain point. Personas have a header that includes the persona's name, a photo, and a quote that summarises the persona's needs. The body of the persona includes the persona's background, demographics, goals, needs, and frustrations.

- User Stories: A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. User stories are often written in the form of: "As a [type of user], I want [some goal] so that [some reason]."

- Scenarios: A scenario is a narrative that describes a user's interaction with a product. Scenarios are used to help designers understand how users will interact with a product and to identify potential issues that may arise.

- Storyboards: A storyboard is a series of sketches that show how a user will interact with a product. Storyboards are used to help designers visualise how a user will interact with a product and to identify potential issues that may arise.

The UX process normally follows some key steps:

1. Empathise: Understand the user's needs and motivations. The key is to understand your users' needs, motivations, and frustrations. This might lead you to generate user personas, map scenarios, and create empathy maps.

2. Define: You collate and distill all the information collected from your users and identify the key problems and needs that you should address. Furthermore, you need to prioritise these problems and needs to ensure that you are focusing on the most important issues.

3. Ideate: It is about generating ideas, whichever way you find the most useful (e.g. sketching, brainstorming...) Once you have sketched your ideas, you can develop them into **wire frames**, a two–dimensional illustration of the user interface that specifically focuses on space allocation and prioritisation of content, functionalities available, and intended behaviours.

4. Prototype: Create a simulation of the final product. Furthermore, you should simulate the interactions that users will have with the product. Prototypes can be low–fidelity (paper prototypes, wireframes) or high–fidelity (interactive prototypes). At this point, you should test your design, receive feedback, and go back to address any issues before moving on to the next stage.

5. Test: Test your prototype with real users. This is the most important step in the UX process. You should test your prototype with real users to see how they interact with it and to identify any issues that may arise. You should also gather feedback from users to help you improve your design.

UI, on the other hand, is about the look and feel of the product, especially when it comes to visual design through components like buttons, text, images, and other elements. UI design is about creating a visual hierarchy that guides the user through the product and helps them understand the product's structure. A good UI can make or break a product—the intuitiveness of the design can be what makes your app stand out from that of the competition.

One of the most commonly–used industry tools for UI design is **Figma**. Figma is a cloud–based design tool that allows you to create, prototype, and collaborate on designs in real time. In Figma, the **canvas** is where you create your designs, and the **frames** are the containers that hold your designs. You can create multiple frames within a single canvas to create different versions of your design. You can also create **components** in Figma, which are reusable elements that you can use throughout your design. Components are a great way to maintain consistency in your design and to save time when creating new designs. There is also a section of the Figma interface called the **layers panel**, which shows you all the layers in your design and allows you to organise them. Lastly, **groups** are a way to organise layers in Figma. You can group layers together to keep your design organised and to make it easier to work with. After finishing your design, you may want to use a feature known as **prototyping**, which allows you to create interactive prototypes of your designs in order to test them with users.

1.1 User–centered Design

User interviews are an extremely effective way to learn about your users and their needs—as well as your interface's strengths and weaknesses. If you design without conducting such interviews, you will have to make assumptions about your product that may lead it to fail. The experiences of your users can be illustrated through a **customer journey map**, which is a visual representation of the steps the user takes to complete a task, as well as their emotions and thoughts throughout.
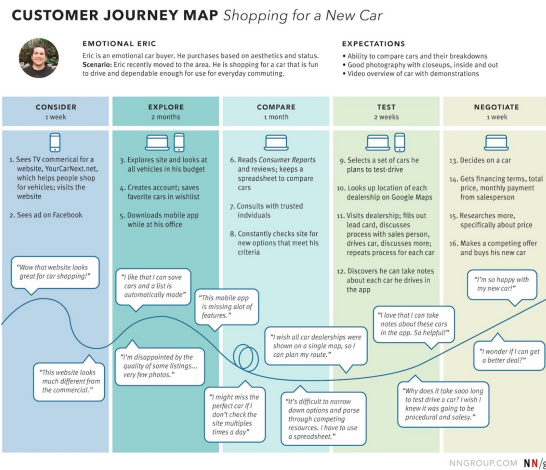
Figure 1: Customer Journey Map, sourced from here

Once you have the findings from user research, you can start with the UX process directly. Some common tools used in the early stages of the process, the empathising, are:

- User Persona: A realistic model of your main target audience. It prevents designers from making assumptions by trying to continuously empathise with this persona. Although the persona is fictional, it should be based off traits from user research.
  Customer personas usually include items like the persona's name, a photo, a quote, a short bio, demographics, goals, needs, and frustrations.

- Empathy Map: A tool used to help designers understand the user's needs and motivations by putting them in the user's shoes. It is divided into four quadrants: **Says**, **Thinks**, **Does**, and **Feels**. Often, these are used to analyse particular personas.

- Scenarios: A narrative that describes a user's interaction with a product.

2 Evaluating Interactive Design

Design is inherently subjective, but there are guidelines to ensure the experience of the overall userbase is as smooth as possible. These are but mere guidelines—apply them broadly during the design and ideation stages as aids to spot bad habits before they happen. The most common guidelines are:

- Rams' Ten Principles of Good Design: Postulated by Dieter Rams, a famous industrial designer, these principles are:
  1. Good design is innovative.
  2. Good design makes a product useful.
  3. Good design is aesthetic.
  4. Good design makes a product understandable.

5. Good design is unobtrusive.

6. Good design is honest.

7. Good design is long–lasting.

8. Good design is thorough down to the last detail.

9. Good design is environmentally friendly.

10. Good design is as little design as possible.

- Nielsen's Heuristics: Postulated by Jakob Nielsen, a web usability consultant, these heuristics are:

   1. Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

   2. Match between system and the real world: The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system–oriented terms. Follow real–world conventions, making information appear in a natural and logical order.

   3. User control and freedom: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

   4. Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

   5. Error prevention: Even better than good error messages is a careful design that prevents a problem from occurring in the first place. Either eliminate error–prone conditions or check for them and present users with a confirmation option before they commit to the action.

   6. Recognition rather than recall: Minimise the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

   7. Flexibility and efficiency of use: Accelerators–unseen by the novice user–may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

   8. Aesthetic and minimalist design: Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

   9. Help users recognise, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

   10. Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any

such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

They are all related to human–computer interaction.

- Schneiderman's Eight Golden Rules: Postulated by Ben Schneiderman, a computer scientist, these rules are:

  1. Strive for consistency: Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

  2. Enable frequent users to use shortcuts: As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

  3. Offer informative feedback: For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

  4. Design dialogues to yield closure: Sequences of actions should be organised into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

  5. Offer simple error handling: As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

  6. Permit easy reversal of actions: This feature relieves anxiety since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options.

  7. Support internal locus of control: Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

  8. Reduce short–term memory load: The limitation of human information processing in short–term memory requires that displays be kept simple, multiple page displays be consolidated, window–motion be used, and that the user not be burdened with remembering information from one part of the dialogue to another.

Accessibility is also a key concept in design. It is about making your product usable by those users with disabilities or impairments. The most important principle of accessibility is to make sure you design for the wide range of users who might use your product, including those with who are blind, deaf, those who have cognitive disabilities, or any other accessibility needs.

Other important elements are not using colour as the only way to convey information, making sure there is enough contrast between foreground and background, ensuring that interactive elements are easily identifiable and accessible with only a keyboard (some users may be unable to use a mouse), ensure that form elements include clear labels, provide easily identifiable feedback for errors, designing for different viewport sizes, providing controls for contents that start automatically, and using headings and spacing to group related contents. In UI design, a **component** is any part of your design that can be logically grouped together and can be reused. You might start small with simple components such as a button, combine them into more complex components like a form, and end up building full interfaces using individual and combined components. Components are a great way to maintain consistency in your design and to save time when creating new designs; for example, you can create a header component for every page in your website instead of designing it from scratch each time. Consistent components not only make your product aesthetically pleasing and consistent, but also aid the user by making the product more predictable.

A **design system** is a collection of reusable components, guided by clear standards, that can be used to design products at scale similar to brand style guide. They contain rules and best practise behaviours for each design components. The most successful design systems tend to have design guidelines, a design pattern library, a UI kit or component library (usually buttons and widgets), and process design libraries.

Navigation Icons are another essential part of UI design, since they help users find information within your product. Some of the most common ones are:

- Hamburger Menu: A common icon used to represent a menu. It is usually represented by three horizontal lines stacked on top of each other. It is best to avoid this icon in the desktop view of your product, as it is not as intuitive as other icons.

- Kebab Menu: A vertical list of three dots that is used to represent a menu that opens a menu with more options. It is usually located in the upper corner of a screen.

- Meatball Menu: A horizontal three–dot menu that opens another menu with more options. It is usually used to display actions for a specific item or open a menu. The meatball menu is easier to repeat and use on the web for elements such as tables or other horizontally oriented elements.

- Bento Menu: A grid of nine squares that is used to represent a menu that opens a menu with different applications, solutions, or functions within the same product, allowing you to switch between them quickly

- Döner Menu: The most widely used filter symbol, it is often used together with the "sort by" option. It is a vertical stack of three lines of varying sizes.

2.1 Form Design

Forms are a key part of many digital products, and they are often the primary way that users interact with a product. When designing forms, it is important to consider the following:

- Keep forms simple and straightforward!

- Use one column when possible—this is a lot more straightforward for users.

- Sort the form fields from simplest to most difficult

- Include smart defaults, like using the user's current location to instantly fill the location field

- Use reCAPTCHAs instead of CAPTCHAs

- Space input fields consistently

- Create logical grouping, using whitespace or borders to separate groups of related fields

- Implementing a clear path of completion for users to know where they are in the form and how much they have left to complete. Progress indicators are useful for this

- A clear validation error message, ideally at the point where the user makes the mistake (i.e. upon input)

- Clear and specific labels for each field

- Provide exact requirements for each field, as well as a strength indicator for passwords

- Align every text field and title to the left

Some of these guidelines can be generalised to other components of design.

3 Applied Design Fundamentals

Let us recap some important concepts in Figma:

- Frame: This is where your design lives. You may think of them as screens in an app.

- Layer: Object within a frame. As you add objects to a frame, new layers are created in the layers panel.

- Group: In a group, multiple objects can be combined and they will appear as a single layer in the layers panel. In Figma, you can modify elements within a group by double–clicking on the group.

Responsive Design is an approach to web design that uses **flexible layouts** that adjust to the size of the screen the user is viewing the content on—thus removing te need for designing multiple versions of the same site. According to this paradigm, web pages/apps are built of squares and rectangles contained within a grid with invisible lines. These lines are the ones that align and organise the content of your page.
In Figma, grids are created by clicking on the *layout grid* section. Here, you may choose between a column grid (ideal for ordering content horizontally), a row grid (ideal for ordering

content vertically), or a grid grid (ideal for ordering content in a grid). You may also choose the number of columns, the gutter width, and the margin width. The most common number of columns are 12 for desktop, 8 for tablet, and 4 for mobile. Most grids have 60-80px column widths. Gutters are the negative space between columns and usually are of 20px. The intersection of rows and columns creates units of space called modules, where you can place design elements. Once you are using the grid, you may fix content to parts of it by using the *constraint* option. Once an element is constraint, it will dynamically change size and position if the frame size is also altered.

There are several types of grids according to your needs:

- Block grids: Simplest grid structure, a large rectangular area taking most of the space. It only has a single column whose elements are arranged vertically, surrounded by margins. They are ideal for continuous content like large blocks of text or full–width images.

- Column grids: The most common grid, they are composed of multiple columns.

- Modular grids: Presents several items simultaneously, arranging them at an equal distance based on their position as modules in the grid. They are ideal for content that needs to be displayed in a grid, like a gallery or a list of products.

- Hierarchical/free style grids: They are used when the content is not linear and needs to be displayed in a more complex way, like in dashboards. Any irregular grid that accomodates specific content needs can be called Hierarchical.

- Baseline grids: They are used to set the leading from one line of text to the next for consistency.

Wireframes are a visual guide that represents the skeletal framework of a website. Normally, they do not include things like colours, branding, or images, since they are mainly used as a reference for the page. The focus of a wireframe is to focus on the user experience and what they need to do to accomplish a task. Usually companies start designing mobile wireframes and then adapt to desktop, since nowadays most internet users access websites via their mobile devices. Another important tool when designing a website is a **prototype**. Prototypes are meant to be a cost–effective representation of a website that can be shown to potential users so that design and user experience flaws can be spotted and fixed at an early stage. They could be extremely simple, like a series of drawings on paper (paper prototypes), or even semi–functional programs (interactive prototypes).

At this point, you usually perform what is known as **usability testing**. This involves giving tasks to real people, monitor their behaviour in using the app, and ask for feedback that can allow you to improve the app in the future. The most effective method to obtain insights from usability testing is through qualitative testing, which implies observing the user's behaviour and asking them questions. When resources or time are a constraint, you may use quantitative testing, which involves collecting data (such as screen time on each page) from a large number of users and analysing it to identify patterns.

4 Designing your UI

There are a few key design elements you should understand properly:

- Line: They assist in directing the eye to a certain area. A line's thickness can convey information, like importance or hierarchy, where the thicker the line the more important the information it conveys. Lines are very useful in separating content and creating a hierarchy.

- Typography: To make your app readable in multiple devices, you need to balance factors like typeface, type size, colour, line height, and letter shapes.

- Colour: It sets the mood and can help evoke emotions and create a sense of cohesiveness. Qualities you should pay attention to in the colour are hue, the variety of a colour; tint, the addition of white to a colour; shade, the addition of black to a colour; tone, the addition of grey to a colour; and saturation, the intensity of a colour.

- Shapes: They draw the user's attention to a layout. Geometric shapes are accurate and can be used to create a sense of order, while organic shapes are more fluid and can be used to create a sense of movement.

- Space: The area occupied by shapes and other elements. The area that is left empty is called negative space, and it is used to create a sense of balance and order in a design. Elements in the design can help you connect or separate different parts of the design.

Going from a wireframe to a high–fidelity design, a design that closely ressembles the end product, is a process that involves adding the elements in the brand style guide you may be provided or have created. Sometimes, a **UI kit** is also provided. This kit contains a series of files that are necessary for a good style.