Entry 1: Usability vs UX

Usability, UX and Human-Computer Interaction may sound similar for most people, mostly because they have the “core” objective of making a product usable, or at least, not annoying and difficult to use, however, their meanings are slightly different. Usability is defined by ISO 9241-11 as “effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments”, UX, on the other hand, is defined by the same standard (ISO 9241-210) as “all aspects of the user’s experience when interacting with the product, service, environment or facility”, In addition, HCI can be defined as a compound of various knowledge areas in order to improve the information sending between humans and computers/systems and make the work on them more efficient. The interaction with the user is what connects these 3 terms, it is that interaction where each term highlights their its unique aspect., I remember a past class where a very similar question was made: “What’s the relation between UX, Usability and HCI?”, I answered that both UX with its focus on experience and HCI with its focus in interaction between user-computer and efficiency, were what defined a “usable” system, something that was “fit for use”, but it seems that was quite the opposite, UX was the “umbrella term” with usability and part of HCI inside of it, meaning that usability with its focus on system’s goals (considering the user) and HCI collaborate to create an “User experience”.

We can say that usability focuses on the goal when using a product or system (defined by users), while UX focuses on human factors that make the product pleasant to use, for example, a website can have all the expected functionalities, be learnable, sturdy against errors, effective, efficient and overall, usable, but if colors are unpleasant, there’s not visual “feedback” or consistency in the UI, the website won’t be enjoyable for the user. On the other hand, HCI borrows from those terms but also focuses on efficiency, effectiveness, enjoyment, psychology and ergonomics to make and a system efficient to use, smooth and pleasant for use.

A UX feature for the HCI project I’m working on would be confirmation messages when creating/editing/deleting sessions to prevent confusion or even the repetition of the action if the user doesn’t notice the change made. This feature would allow feedback for the users (that is an attribute by itself), allowing them to know if changes/actions were made, and it could be verified with observation of the system usage to know if the user feels comfortable, doesn’t confuse and a questionnaire evaluating the average feeling and impact of those messages on the user.

All these terms tend to be confusing for me because they can be stretched at a point the meaning might dilute, for example, learnability and efficiency are attributes of usability, but isn’t interoperability, performance and resource usage also related to it?, because a system without those attributes leads to errors and defects noticeable by the user and therefore, affecting the goals of the system, and usability.

Entry 2: Elicitacion methods

In the user research of my team (Team 3) we use an unstructured interview for our user research, this technique consists on an interview without a plan, structure or objectives defined before, the purpose of this technique is allow the interviewer to lead it on different topics that often spontaneously appear when making questions, allowing adaption and flexibility but with the risks of diverting from objectives and totally relying on the interviewer. Our team implemented almost like it should but with the difference that there were multiple interviewers.

At the end, only 1-2 of us asked the questions and the other registering the findings but having at least 3 of us doing the same task ended up with redundant information and some of the “scribes” not doing their task (I believe that happened because some of us believed that the others and the recording had it covered), what ended up on the team relying on review extensive audio recordings to get the missing information.

When doing the interview, I tried to ask some questions while registering findings but some of these felt off topic, unlike the questions of the main interviewers, that left me thinking how they came up with the right questions, maybe is about experience (like Diaz having a job and having done this interview more than once) or just sheer initiative, confidence and intuition like Deco? I ended up believing that it´s a mixture of all that and abstract “plan”, not a complete structure for the interview, but key points and ideas annotated or registered to get back on the interview.

The way we could improve the technique is going over the documentation and other artifacts before the interview to know what do we know as a team, and what do we need, preparing some questions and key points to have a guideline for the interview, defining roles for each member of the team and encourage initiative within the members to make questions and express their doubts about the things told in the interview in order to avoid incorrect or missing information.

Even with this said, I find hard to know exactly what the point to “attack” are on a interview, because the main stuff (processes, mock-up, documents used, questionnaires) where there, we just had to ask the specific stuff and how do you know something that specific and detailed is important or not? I have the idea that it is about getting as much information as possible and then filtering, but even the filter has that problem of being selective and having to dissect a 2-hour long audio doesn´t sound fun either.

Entry 3: AI Tools

I remember when ChatGPT first came out to the public, I saw other classmates using it intensively, I first tried to use it as less as possible to “truly learn” how to program, document and design, but I ended up using it frequently to make mind-numbing tasks faster, precise research and having references to know how to code for example. I initially feared that this would make me overdependent on AI’s but at the end, it’s up to individual if use Generative AIs for support on your learning journey or just make it do everything for you, besides, some people though similar things about Wikipedia, in terms of research. Despite approving the use of Gen AI’s, I usually don’t let them make the UI part, I usually send prompts about that if there´s an error I can’t fix or something that could be improved, so when I sent a prompt of a calculator in HTML, CSS and Javascript I expected something inconsistent, but it surprised me.

The calculator worked but what matters is the UI, that was simple, grey with a white box with gray buttons and bars and some of the buttons were misplaced (or placed strangely). It was usable but it didn’t fulfill at all the UX part, then I sent another prompt to improve the UI; The result was a more compact calculator, still with the buttons problem but with stronger visual feedback (contrast of numbers and the operation box, colors on the ‘C’ and ‘=’), still has some details that may decrease the usability, but re-arranging the buttons and the operation box, enabling typing and adjusting the calculator size, may improve the usability of the system widely.

It was surprisingly easy, even with the mistakes the AI made, it provided a “skeleton” of the system and allowed me to focus almost exclusively on the UI, the corrections on the UI weren’t perfect but some of them were good implementations like the increase of contrast.

AI tools might do things fast, but they still have a long way in terms of aesthetics and visual and aren’t 100% precise. These tools may allow developers to focus entirely on UI with solid “bases” for their applications. These still can´t develop a detailed UI without lots of prompts and corrections, but they can do simple yet tiring work for you to focus on other things, because that’s one of technology’s purposes, making difficult things easier, right?

Entry 4: Lean UX

Ux and agile development are terms that are getting quite popular these days, no wonder why they usually get up being relevant in lots of projects and end up colliding due to their opposite nature, one based on deliverables in a stair-like progress and the other with sprints that make hard to work UX properly.

The lean UX is the proposal for an “Agile UX”, by leaving the deliverables and focusing on ideas, evaluating the plausibility and developing a basic version of the concept. One major advantage is that lots of ideas get considered, evaluated, incorporated if there’s a valuable result, otherwise, it’s discarded.

From personal experience with the HCI project, is that our team didn’t apply a lean UX at all, the focus was more traditional and centered on deliverables like use-cases, prototypes, etc. We already had some deliverables on Maliachi’s repository, so we gathered information and made the others, we still followed that staircase-like sequence. Even though Lean UX should be the “fastest” way to develop UX, does it have the same result if we are not applying Agile methodologies? From my point of view, if we sticked to classic development methodologies, mixing them with Lean UX would cause pacing problems, and an absence of deliverables, that are important on traditional methodologies and UX. To be honest, most of us felt comfortable with the current development pacing and didn’t know about Lean UX, but the process wasn´t perfect.

Even with a traditional development we followed a “do it last night” process, that could resemble the “sprint” of Agile development (without the irresponsibility of course). Maybe it’s just a matter discussing and formally applying the change of the development.

But when we tested mock-up creation in HTML/CSS/Javascript using ChatGPT, we realized that could be our “opportunity” to adapt us to Lean UX, due to the low time it took to make an interactive mock-up (2 hours approximately and if given more time, a basic version of the system. Even considering this, a change to Lean UX would require clear implementation of an Agile methodology (like SCRUM), assignation of product owner, SCRUM master, dev team, creating a SCRUM backlog etc. The weekly deliveries might be another attempt to transition from the classic development, but with the problem of having to transition in first place