Human-Computer Interaction: User Support System

Mohamed Ibrahim[1]

[1]*Student, School of Computer Science and Engineering, Vellore Institute of Technology,Chennai, India*

1.1 Interaction

The goal of HCI is to make an interface as straightforward as possible so that the user may concentrate on the tasks at hand and not spend too much time thinking about the interface. However, rather than wanting the user to worry about the interface, the designer wants the user to focus on the work for as long as feasible. Consider an example, when a user is using a pre-defined software and the user is always thinking about how to use the software. It frustrates the task because the user is hoping the program would help him achieve his goals. So, a designer aims to make the user feel as though the user is actively participating in the work. When this interaction comes into the picture, the UI sort of disappears. The interaction between a person and a computer and the subsequent interaction between the computer and the human seems to be the most straightforward. The computer is only acting as a mediator between the user and the task.

1.2 Reflections: Interacting & Interfaces

Video games provide excellent examples of interfaces disappearing. The Interfaces are created, prototyped, and then evaluated. Instead of managing the game world by pushing buttons on a controller, a great video game makes the user feel like they are inside it. The user can do this by using a controller with an intuitive design, where pushing forward advances forward and pressing backward moves backward. But frequently, the designer relies on the player to gradually pick up game control. However, as users gain knowledge, it fades into the background of their communication. The concept of using several remote controllers to operate what seems to be the same device is a famous illustration of an environment where interaction is more obvious. Therefore, the TV and cable box are both controlled by two remotes. And watching TV seems to be the only task. Therefore, depending on what the user is attempting to achieve at a particular time, the designer must decide whether to use this controller's number pad or this controller [2].

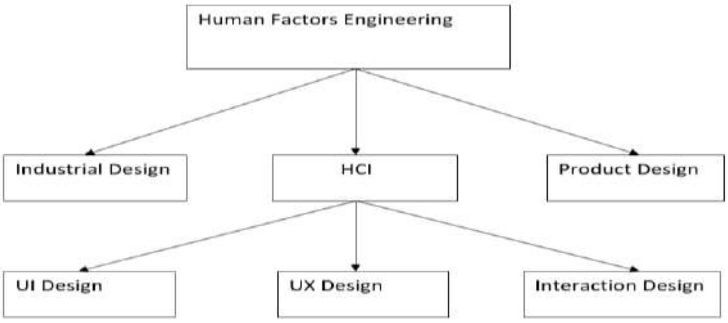

Figure 1: The HCI Space

1.3 HCI in the Big Picture

The HCI's remarkable ubiquity is one of its most interesting features. The users can engage with computers since they are all around. Although it is fascinating to consider creating the tools and interfaces so frequently, there is also risk involved. Because the users use computers to engage with other humans or other users. But cannot consider these users as experts. That is not the case, though.

“The users can be experts using computers, but that does not mean they are experts at developing relationships between other users or humans and computers.” For example, A person watches lots of movies. But that does not mean he can start directing movies. For directing movies, a person must first work on short films and get trained as an assistant director. Only then he can understand the mindset of the director.

1.4 HCI vs Human Factors

Let us compare HCI to the larger area of human factors. Human factors are interested in planning how users will interact with various items, systems, and gadgets. That must be a common phrase. Although computers are themselves systems or products, the focus is on developing the interactions between humans and computers. If a person buys a brand-new electric vehicle, it will likely be covered with numerous computers. Human factors are interested in how to engage with the car's non-computerized components as well as its computerized components. It cares about details like the chair's location, mirror size, steering wheel height, etc. Not just the electronic components, but the overall environment, is what it is interested in developing. But it also implies that technology is interested in many human traits, such as how individuals view the world and their expectations about it.

An automobile might not have been a vehicle for HCI twenty years ago. Ten years ago, the industrial design of a wristwatch would have taken precedence over HCI. Things like refrigerators and shower heads have begun to transform into fully digital gadgets just in the last few years. The distinction between human-computer interaction and human factors engineering is blurring as computers become more and more integrated into gadgets. It will not be long until almost everything in an automobile is controlled by a computer as they grow more and more commonplace. View the Model S's interior. Nearly every single component of a Tesla's console is a massive computer. Car keys have evolved into computers on key chains, watches have evolved into computers on wristbands, and cars themselves have evolved into computers. HCI is widely prevalent because computers are.

1.5 HCI vs User interface design

For a long time, user interface design dominated the field of human-computer interaction. The development of tools similar to the light pen and the first computer mouse, that allow adaptable interaction with objects on the screen, was one of the initial developments in human-computer interaction (HCI).

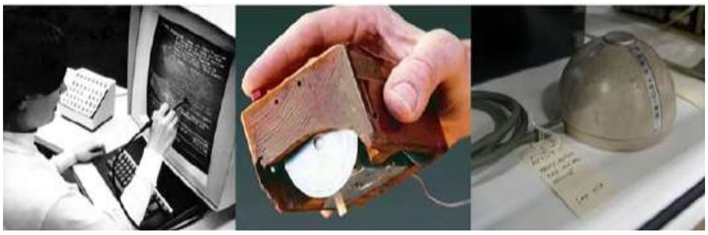

Fig 2: (a) Original Light Pen (b) First Computer Mouse (c) First Rollerball Mouse

Due to its lengthy history, UI design has developed into a distinct subject. UI design is where many of the concepts in HCI initially emerged. However, HCI is more concerned with issues that span beyond a user's interaction with a single screen.

1.6 HCI vs User Experience Design

The connection between user experience design and HCI is getting closer. Understanding how people interact with computers is a major focus of HCI. The goal of user experience design is to control how people interact with technology. Understanding the user is essential for excellent user experience design. The user must comprehend how they engage with interfaces. Because of this, users can observe user experience design as a branch of the larger discipline of HCI. The users can utilize HCI approaches and ideas to guide how to build user experiences. But it is crucial to recognize how synergistic this relationship is.

The designers may use the understanding to guide user experience design, but once tested the designs, the designers can utilize the success or failure of the evaluations to guide the growing understanding of human-computer interaction. The ability to produce high-quality designs as a result of future expertise is proof that the users have the right information. The understanding of human-computer interaction will advance if the users produce a design with some understanding and it fails. If this happens, the initial assumptions may be incorrect. Just as a user interacts with an interface to complete tasks before assessing the outcome of their participation, interface designers should assess the effectiveness of their work.

1.7 HCI vs Psychology

The connection between HCI and psychology is made via the research component of HCI. Furthermore, if one takes a step back and looks at the hierarchy of disciplines from a wider perspective, one might conclude human factor engineering is essentially the fusion of engineering, psychology and cognitive science domains. However, this relationship to psychology still exists and is mutually beneficial in HCI, where the engineering component takes the shape of software engineering.

In 1992, Apple psychologists were interested in learning how individuals managed the constant stream of information in their offices. The psychologists observed that individuals often gathered similar items into heaps, like a looser file system, so they created a computer interface to mirror that behavior. The outcomes of that research were then utilized to consider how individuals were first organizing their workplaces. As a result, they were able to better comprehend their consumers' mental processes and create an interface that was beneficial to them. The experience of building something can be taken from their workplaces as they give a deeper knowledge of how people think about their workspaces.

1.8 HCI: Research and Design

Research is crucial in HCI, researching the user, comprehending their needs, and assessing their interactions with ideas. HCI, however, is also about design. Even though the researcher is prototyping with research in mind, design is the prototype step. Human-computer interaction (HCI) is the study of designing user interfaces that facilitate human-computer interaction. This involves frequently applying established principles for good interaction, such as distributed cognition, ensuring that users form accurate mental models of how the interface functions, and universal design.

It is essential to understand that these are not two opposing viewpoints currently. The results of user research inform design work, and the effectiveness of those designs informs future research. Again, one may have noticed how similar this is to the feedback loops that are designed for clients. By using the existing design, the researchers engage in the activity and use feedback to refine and update the model. The researchers develop effective interfaces with the knowledge of psychology, cognition, and human perception in the mind, and the outcomes of those interfaces are then used to guide an ongoing study. This is the essence of what HCI is: utilizing research to guide designs and using the outcomes of those designs to guide ongoing research. That may entail utilizing a screen while seated at a desk. That can include wearing a wristwatch or a smartphone. It may include interacting with touch - or gesture-based devices, or it may involve cutting-edge innovations like augmented and virtual reality.

1.9 Conclusion

This chapter covered the difference between HCI and similar fields like UI design, UX design, Human Factors Engineering, etc. Also covered the broad topic of human-computer interaction, including how to define people, computers, and the interactions between them. Next, spoke about how HCI fits within a larger hierarchy of fields, including psychology, human factors, and UI design. Finally concluded by discussing HCI in research and design.

CHAPTER - 2: Exploring HCI

As more and more gadgets incorporate computers, HCI spreads over a wider range of applications. In the past, creating a car or a refrigerator would not require much consideration of HCI, but as computers become more and more ubiquitous, this is no longer the case. New technical advancements are also creating new research opportunities. HCI is witnessing a lot of intriguing advancements in fields like wearable technology, augmented reality, and virtual reality. Technologies, domains, and ideas can be categorized into three types.

Emerging technology capabilities called technologies enable researchers to design fresh and engaging user experiences. Domains are pre-existing industries, including healthcare and education, that might be dramatically affected by computer interfaces. They are hypotheses on how individuals engage with interfaces and their environment. The current categorization of this is quite arbitrary. Numerous areas overlap. Emerging concepts like contact-sensitive computing are made possible by modern technology like augmented reality.

2.1 Technology: Virtual Reality

The researchers have just started to understand what they can accomplish with these new technologies, and virtual reality represents an altogether new category of interaction and visualization. Virtual reality has received a lot of attention in the media recently, but there are other applications as well. Applications for tourism, business, art, education, and virtual reality may be found everywhere.

2.2 Technology: Augmented Reality

Virtual reality often functions by substituting its input for the actual world's visual, aural, and occasionally even all tactile or kinaesthetic sensations. On the other side, augmented reality enhances what the users hear and see in the actual world. Contemplate a device like Google Glass, for instance, that automatically overlays directions in users' range of vision. Instead of merely putting up some visual reminder, it would emphasize the path to take if the user were driving. Instead, of merely substituting for external stimuli, the input it offers complements them. And that brings about both tremendous obstacles and extraordinary opportunities. Consider the gadgets that will be able to improve by seamlessly integrating into daily life. Consider systems that could, for instance, translate voice or text into a foreign language automatically or that might display restaurant ratings to anyone passing by. Imagine a system that students might use to automatically point out fascinating facts when visiting national parks or museums, according to the individual student's interests. A fascinating set of societal issues are raised using cameras in augmented reality, notwithstanding the potential for genuinely amazing applications.

2.3 Technology: Ubicomp and Wearables

The trend toward incorporating computing power into more and more commonplace things is known as ubiquitous computing. It has a close connection to the newly emerging concept of the Internet of Things. A few years ago, computers, refrigerators, or watches were not so common, but as microprocessor prices dropped and global connectivity grew, computers became gradually commonplace. The modern HCI involves considering whether someone might use a computer while operating a vehicle or exercising. It entails figuring out how to create intelligent machines that also need some cognitive effort from the user, such as refrigerators that keep track of their contents and provide people with advice when it is needed. The desire for greater ubiquity has also fuelled the growth of wearable technology. The most typical instances of this are undoubtedly exercise trackers, but other examples of wearable technology include smartwatches, Google Glass, augmented reality headsets, and even items like cutting-edge hearing aids and robotic prosthetic limbs.

2.4 Technology: Robotics

The physical makeup and capabilities of robots as well as the artificial intelligence that powers them are now receiving a lot of attention. But as robots gain popularity, a brand-new area of human-computer interaction called human-robot interaction will start to take shape. The area is already developed. The first conference on human-robot interaction was held in Salt Lake City in 2006, and since then, numerous others have been established. How, for instance, can designers guarantee that robots will not hurt people due to poor reasoning? How do we deal with unemployment as robots get more and more capable? How can the researchers ensure that humans engaging with robots are being heard and understood by giving them subtle, implicit feedback? How can the researchers help people teach robots things rather than merely programming them, or is it possible to build robots that can instruct people? Robotic innovations are already being used in fields like health care and services for people with disabilities.

2.5 Technology: Mobile

The phenomenal expansion of mobile as a computing platform has been one of the largest advances in computing over the past several years. The environment is truly mobile-first, which poses some important design difficulties. There are also significant prospects for HCI in mobile computing. Most individuals constantly have a computer with them because of mobile phones. That may be used to help with activities like stargazing and navigation. As it has the technology required to support initiatives, context-aware computing, ubiquitous computing, and augmented reality are all closely tied to mobile computing. Mobile computing, however, already poses a few intriguing problems that need to be solved. However, mankind is still a long way from being able to perform all the jobs that perform on computers on cell phones, such as writing essays or basic programming. Smartphones are fantastic for social networking, self-management, gaming, and many other things.

2.6 Idea: Context-Sensitive Computing

The goal of context-sensitive computing is to provide computer interfaces with the same contextual awareness that people have in their daily lives. For instance, a user uses a smartphone differently depending on where the user is—while traveling, operating in a car, or seated on the sofa at home. But what if the user did not have to consciously tell his phone what mode he/she was in? Imagine if it just recognized that the user was driving and opened Google Maps and audible for the user. Although some services have begun to offer this, much more study must be done on context-sensitive computing, particularly as it pertains to technologies like wearables, augmented reality, and ubiquitous computing.

2.7 Idea: Gesture-Based Interaction

Gestures are used to communicate. Gestures make up the entirety of a language. Therefore, would not it be fantastic if computers could also understand the gestures of users? Gesture-based interaction is a newly developing topic in technology. Devices like Microsoft Kinect, which has a wide range of uses from gaming to healthcare. There are now various mobile gesture-based interaction programs as well as wristbands that respond to certain hand gestures. The potential for gesture-based engagement is huge. Given that the muscles in the fingers have some of the most refined motions, a system based on finger movements might enable an astounding number of interactions. Based on the system's identification of the movement of the muscles in the wrist, could one day be able to type invisibly in the air in front of individuals. This might finally lead to the complete demise of conventional PCs in favor of mobile devices.

2.8 Idea: Pen-and-Touch-based Interaction

The way some technologies appear to come full circle intrigues manhood every time. Before the invention of computers, had only ever engaged directly with the objects created. The demand for interfaces between users and work arose unexpectedly. Computers are now deliberately attempting to mimic the organic ways in which were traditionally interacted. Nowadays, touch screens are on almost all computers. Because it puts the user closer to the tasks they are seeking to do, it is an effective method for designing straightforward user interfaces. Imagine someone who has never used a mouse before. He might need to alternate between looking at the mouse and the screen to observe how engaging down here can alter what he/she sees up here. The user can interact with a touch-based interface, in the same manner, he does with objects in his immediate environment. A user can use a pen on a touch screen just like they can on paper. They capture a person typing on a screen. This offers the accuracy required to engage with work extremely precisely and carefully. Tablet-based interaction techniques are now being applied in industries including music and painting. Most comics discovered online are created just like this, fusing the dexterity of human hands with the might of computing.

2.9 Idea: Information Visualization

The enormous accessibility of data is one of the greatest developments of the digital era. Data science and machine learning are tools that scientists and researchers use to analyze large amounts of data and make judgments. These insights, however, are frequently only applicable if they can be conveyed to regular people. Information visualization can help with this. Now, data visualization may not immediately spring to mind as an illustration of HCI. After all, there is no computer involved in the process at all; could create data visualization on a napkin and print it in a newspaper. However, the power of computers allows users to re-present data in intricate, animated, and interactive ways.

2.10 Idea: CSCW

Computer-Supported Cooperative Work is known as CSCW. The field is exactly what it says it is. There are instances of CSCW in use besides distributed teams. The community frequently divides the world into two dimensions time and location.

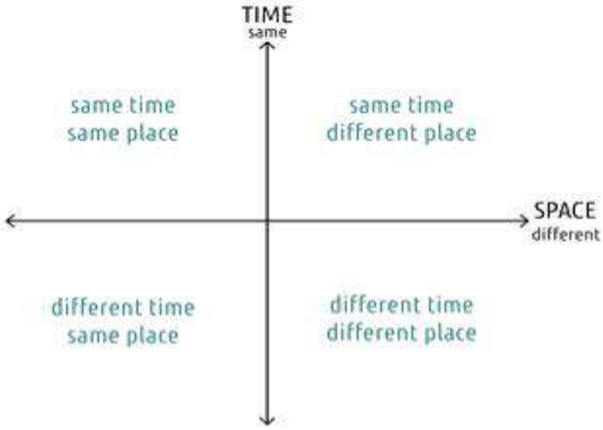

Figure -3: Two dimensions of space and time with respect to CSCW

How might computers support collaborative work among people? Whether designing for users who will be present at the same time and location or for users who will be present at various times and locations might be how to define the design. Facebook and Instagram are two instances of workplace chat tools that are used at the same time and in various locations. They simulate the real-time office environment by enabling instantaneous communication beyond physical distance. Imagine a museum kiosk that requests visitors' whereabouts to generate a map showing everyone's origins. It would be a different period and location then. At the same time and location, everyone utilizes the interface. Computers can still facilitate collaboration even when the users are in the same location and time. Therefore, CSCW can provide simultaneous co-locational cooperation as well as cooperation beyond established geographic or temporal boundaries.

2.11 Idea: Social Computing

The area of HCI known as social computing is concerned with how computers influence how humans communicate and engage with one another. Recreating social norms within computer systems is one item that fits under this general category. So, one could frequently utilize emojis or emoticons when communicating online, for instance. Those are virtual recreations of a few days, of unspoken encounters with one another. So, for instance, depending on the emotion expressed, each of them has a distinct meaning. Of course, social computing is interested in much more than simply emojis, including Wikipedia, online gaming, social networking, and dating services. Every place where computing and social life overlap is where social computing is fascinating.

2.12 Domain: Special Needs

Helping those with special needs is one of the most intriguing HCI application areas. Users can use computing to make up for age, injury, and incapacity. Consider a robotic prosthesis as an illustration. Of course, part of that is engineering, and part of it is neuroscience. But it is also important to understand how the person intends to use such a limb and the tasks they need to perform. That's HCI intersecting with robotics.

Figure - 4: One-Armed Robotic hand

Many of the new HCI technology areas have the potential to be of enormous use to those with special needs. Imagine developing an autonomous wheelchair with artificial intelligence and context-aware computing, or envision providing virtual reality to persons who are paralyzed in some way. These initiatives would only focus on a limited segment of the populace. However, the effect of that part would be incomparable.

2.13 Domain: Healthcare

Processing the vast amounts of data that are being captured every day is a major focus of modern healthcare initiatives. However, a connection to actual persons must be made at some point for that data to be valuable. Perhaps it is giving clinicians tools to evaluate and contrast various diagnoses visually more readily. Perhaps it is providing people with the resources they need to keep tabs on their health and available treatments. Perhaps the purpose of information visualization is to help patients comprehend how different choices influence their overall health. The ability to recognize when patients are going to do something they should not is possible because of context-aware computing. Numerous HCI apps are also available for maintaining personal health, such as Fitbit for tracking physical activity and MyFitnessPal for eating. If those interfaces are simple for users to use, they will be successful. They should ideally be virtually undetectable. Virtual reality, however, is the emerging HCI and healthcare confluence that is most intriguing. In a secure, yet incredibly lifelike setting, virtual reality can assist people in facing their fears and worries. Human health is at the center of healthcare generally. And in contemporary healthcare, computers are frequently employed. Therefore, there are many healthcare applications for human-computer interaction.

2.14 Domain: Security

Most researchers in network security focus on the algorithms and encryption methods that must be secured to sustain secure connections. But if individuals just choose not to utilize them, even the most secure communication techniques lose their effectiveness. And historically, seen that individuals have very limited patience for situations in which security precautions interfere with their ability to perform their duties. Security must be useable to be effective. People just will not utilize it if it is not usable. There are several ways that XEI might improve security usability. One benefit is that it could just make such acts simpler to carry out. Forms using CAPTCHAs are used to verify that users are real people. And whereas they once required the recognition of letters in intricate graphics, they are now frequently as straightforward as a checkbox. The computer utilizes mouse motions that resemble human movements to determine whether the user is a person. As a result, taking part in that security action is far less irritating. HCI, however, may also improve usability by articulating and visualizing the requirement for security. It appears arbitrary, which is why many individuals become irritated when systems demand complicated passwords that match requirements. The requirement may be considerably less unpleasant if the system instead communicates to the user the reasoning behind it.

2.15 Domain: Games

One of the most basic forms of HCI is seen in video games. They are a fantastic environment for studying HCI since so many of the subjects we cover are so relevant now. For instance, we spoke about how actions and consequences must be logically mapped. A great game is an example of that. The user should have the impression that they are engaging with the game environment when using the controller. As the player acts, judges the outcomes, and adjusts as necessary, video games are essentially perpetual feedback cycles. A lot of the critiques of video games are criticisms of poor HCI if one goes through the reviews. It is difficult to find out what happened because the controls are difficult to utilize. Failure has a too-low or high cost. They are all illustrations of subpar interface design. Although the objective and the interface are so closely related in gaming. Their displeasure with a task might help us locate interface issues fast.

2.16 Conclusion

This chapter covered a variety of generic applications for HCI, including various content domains, technologies, and procedures. Discussed the various fields, such as education, healthcare, and virtual reality, where HCI may be used. Giving an overview of HCI research and development has been the aim.

CHAPTER – 3: HCI and Agile Development

3.1 Introduction to Agile Methods

The content we have covered so far was developed over the course of several decades of research in HCI in human factors, and it is all still applicable today as well. New technologies and new areas call for new principals and new workflows, and specifically, the advent of the Internet ensured in new methods for HCI. Many software developers now adopt an agile workflow that emphasizes earlier delivery, more continuous improvement, and rapid feedback cycles.

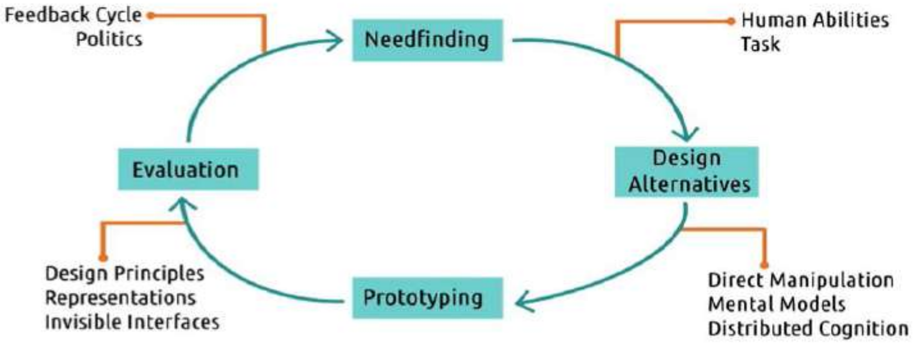

Figure 10: Agile prototyping methods

Long prototyping processes that move from paper to wireframes to live demos involve lots of users in slow qualitative methodologies and are still very valuable. But, nowadays, one wants to build something fast and get it in front of real users. So, in this chapter, we will talk about how one might use agile development methods to engage in quicker feedback cycles [2].

3.2 The Demand for Rapid HCI

Where did these changes come from? One can think of them in terms of some of the costs associated with elements of the design lifecycle. Think back to before the age of the Internet, developing software was very expensive. It required a very specialized skill set. Software distribution was done the same way as buying coffee mugs or something else, one would go to the store and would physically buy the software. The distribution method was expensive as well. If one shipped software was hard to use, the cost of fixing it was enormous. One had to mail each individual person an update disk, and really the only way to get user feedback or even to find out if it was usable, was the same way one would do it before distribution, by having users come in for testing. All of this meant, there was an enormous need to get it right the first time. If not, it would be difficult to fix the actual software, difficult to get the fix to users, and difficult to find out that a fix was even needed.

Fast-forward to now though, is that still true? Development is not cheap now but it is cheaper than it used to be. A single person can develop in a day what would have taken a team of people months to do, 20 years ago. Thanks to advances in hardware, programming languages, and the available libraries. One can look at all the imitators of popular games on either the Android or the iPhone app store, to quickly see how much development costs have come down. It is certainly feasible to churn out a quick imitator when something becomes popular.

But more importantly, distribution for software is now essentially free and updating software is essentially free as well. Every day one can download new apps and have them update automatically in the background. If release something that has a bug in it, can fix it and roll out the fix immediately. Tesla for example regularly pushes software updates to its cars via the Internet. Perhaps most importantly, one can gather usage data from live users automatically and essentially for free as well. It is not just usage data, it is product reviews, error reports, and buzz on the Internet. Lots of feedback about user applications now comes naturally, without having to spend any money to gather it. What all this means is, there is now more incentive to build something fast and get it to users to start getting real feedback, as early as possible. Now make no mistake, this is not a justification to just throw out the entire design lifecycle. The majority of HCI design and research still go through a longer process.

Need several iterations through the full design lifecycle for big websites, complex apps, anything involving designing hardware, anything involving a high-profile first impression, and really anything involving anything, even somewhat high in stakes. A person may come up with an idea for a simple Android game, the person could probably implement the game, get it in front of a real user, and get a lot more feedback [3].

3.3 When to Go Agile

When can one consider using Agile methodologies? Many software engineering theorists have studied this area. First, they say, it must be in a non-critical environment. Agile development by its very nature means that the user does some testing. As such, do not want to use it in an environment where bugs and poor usability would have a major impact. Investing in health or finance is not the best place for agile development. Although there have been efforts to create standards that allow the methodology to be used without compromising security.

Table 1: Categorized between the traditional method and Agile method

	Traditional	Agile
Criticality is…	High	Low
Requirements change…	Rarely	Frequently
Team size is…	Large	Small
Team embraces…	Order	Change

But for things like smartphone games and social media apps, the importance is low enough. Second, it should be a place where requirements change frequently. One of the benefits of Agile processes is the ability for teams to quickly adapt to changing expectations and needs. However, sites like Udacity are constantly adapting to meet new student interests and student needs. If one is working on an interface that fits a more agile process, one also needs to set up a team to work well within the agile process. In other words, a small team is accustomed to change. Unlike large teams that value order. Agile processes in general can be good or bad with the right people [4].

3.4 Paper Spotlight: Towards a Framework for Integrating Agile Development and User-Centered Design

In 2006, Stephanie Chamberlain, Helen Sharp, and Neil Maiden investigated the conflicts and

opportunities of applying agile development to user-centered design. They found interestingly that the two had significant overlap. Both agile development and user-centered design emphasized

iterative development processes building on feedback from previous rounds. That is the entire design life cycle that we have talked about. That is at the core of both agile development and user-centered design. Both methodologies also place a heavy emphasis on the user's role in the development process. And both also emphasize the importance of team coherence. Agile methods and user-centered design agree on the most fundamental element, the importance of the user. User-centered design disagrees with agile development on the importance of documentation and the importance of doing research prior to the design work beginning. But clearly, the methodologies have the same objectives. They just disagree on how to best achieve them. As a result, the authors advocate five principles for integrating user-centered design and agile development. Two of these were shared between the methodologies in the first place, high user involvement, and close team collaboration. User-centered designs' emphasis on prototyping and the design life cycle shows that by proposing that design is run a sprint ahead of developers to perform the research necessary for user-centered design. To facilitate this, strong project management is necessary [5].

3.5 Live Prototyping

One application of agile development in HCI is the new idea of live prototyping. Live prototyping is an oxymoron, and the fact that it is an oxymoron speaks to how far prototyping tools have come. In some areas of development, building a working interface is as easy as building a prototype. This is an example. A tool called Optimizely that in Udacity. Drag and drop to create a working website, the interface is very similar to many wireframing tools, but this site is live. Just click a button to publish this page. Building the final interface is as easy as building a prototype, so why bother building a prototype before building the final interface? Just one of the reasons. One does not usually build it just because it is easy, build it to get feedback before showing everyone a bad design.

But if needed to make small tweaks or minor revisions, this might not be a bad place to start. Especially true when it is low and the potential profit of success is particularly high in any type of e-commerce site. The cost of failure could be losing some of the sales, but the potential benefit is getting more sales over a much longer period. One application of agile development in HCI is the new idea of live prototyping. Live prototyping is an oxymoron, and the fact that it is an oxymoron speaks to how far prototyping tools have come. In some areas of development, where building a working interface is as easy as building a prototype [6].

3.6 A/B Testing

Therefore, in some situations, building the actual interface is no more difficult than building the prototype, so one may skip the prototyping stage entirely. However, the prototype also allowed users to collect user feedback. Now that one can build an interface easily, do not want to expose a fully untested interface to everyone who visits the site right away. One may be able to fix it quickly, but it still undermines users' trust and wastes their time. This is where the second aspect, A/B testing, comes into play.

A/B testing is the term for rapid software testing, typically between A and B’s two choices. Statistically, it is no different from the T-test. What is unique about A/B testing? It usually tests small changes quickly with real users. Usually do this by only deploying the new version, the B version, to a small number of users to make sure there are no major issues. That way one can make sure changes are positive or at least neutral before rolling them out to everyone, but check where the testing feedback is coming from. It automatically accompanies the real user during normal use. There are no additional costs to recruit participants and feedback is immediate [7].

3.7 Mitigating Risk in HCI and Agile Development

Here are quick tips for using HCI and Agile development together. Specifically, to mitigate the risks to the user experience that this more agile development process introduces. First, let us start with something more traditional. Start with a more traditional needs discovery and prototyping process, and then move to a more agile development once you have something up and running. Getting something working gives the user the opportunity to further explore the user experience, but first, one needs something solid, and that comes from more traditional processes. Second, focus on small changes.

Table 2: The sigma rules for using HCI and Agile development

1.	Start with more traditional.
2.	Focus on small changes.
3.	Adopt a parallel track method.
4.	One must be careful with consistency.
5.	Nest the design cycle into agile development

The third is to use the parallel track method. Agile development often uses sprints and development as short as two weeks. This setup allows HCI research to get sprints ahead of implementation. The HCI team can conduct its two-week sprint for prototype needs assessment and low-fidelity assessment. Then hand off the results to the development team for the next sprint. Fourth, pay attention to consistency. One of our design principles was to be consistent both within the interface and across the interface design. If your UI is intended for frequent visitors or users, you need to be careful about how often you respond to user expectations. However, if you are designing something like a museum kiosk, you can be more forgiving with frequent changes. Fifth, nest the design cycle into agile development. You can quickly run many small design cycles, each with little new information. Take all this new information you have gathered and use it in the context of a broader, more traditional design cycle aimed at big long-term improvements rather than small tweaks [8].

3.8 Conclusion

This chapter covered how HCI can work in more agile development. HCI and Agile development emphasize feedback cycles, getting user feedback, and rapid changes. HCI traditionally has done reaching real users and agile emphasizes completing in real-time. Then discussed how HCI aligns with the theory and the goals of Agile development. Agile, however, involves a more intricate set of stakeholders and operations.

References

1. Hiroyoki Nishiyama, Fumio Mizoguchi, “Design and implementation of a user-support system for finding the other person talking on smartphones”, Springer Journal, 2013.
2. Mizuda Takeda, Kaiji Sato, Yasuhisa Hirata, Takahiro Katayama, Yasuhide Mizuta, and Atsushi Koujina, “Standing, walking and sitting support robot based on user state estimation using a small number of sensors”, IEEE Access, 2021.
3. Motoko Iwashita and Makiko Ishikawa, “Buddy System: An Adaptive Mental State Support System Based on Active Inference and Free-Energy Principles”, IEEE Transactions, 2022.
4. Fumio Mizoguchi and Hiroyuki Nishiyama, “Design of a user-support system for vision information using a smartphone”, Springer Journal, 2012.
5. Andrius Lauraitis, Rytis Maskeliunas, Robertas Damasevicius, Dawid Polap, Marcin Wozniak, “A smartphone application for automated decision support in the cognitive task-based evaluation of central nervous system motor disorders” IEEE Journal, 2019.
6. Amparo Guemes, Giacomo Cappon, Bernard Hernandez, Monika Reddy, Nick Oliver, Pantelis Georgiou, “Predicting Quality of Overnight Glycaemic Control in Type 1 Diabetes using Binary Classifiers”, IEEE, 2019.
7. Hiroyuki Nishiyama and Hiroki Imamura, “Design and Implementation of a user-support system for scheduling using a smartphone”, Springer Journal, 2013.
8. MacKenzie, I.S, “Human-Computer Interaction: An Empirical Research Perspective”, Elsevier., 2013.
9. Norman D, “The Psychopathology of Everyday Things” , book Don Norman, 1988.